(suite de la page précédente)

```
else
        echo "Mauvaise réponse : les couleurs de Fleury Hand Ball␣
↪sont: Rose Noir et Blanc !";
}
```

3.17 Connexion aux bases de données depuis PHP avec PDO

3.17.1 Une table simple en SQL :

```
        CREATE TABLE `CARNET` (
          `ID` int(11) NOT NULL AUTO_INCREMENT,
          `NOM` varchar(30) DEFAULT NULL,
          `PRENOM` varchar(30) DEFAULT NULL,
          `NAISSANCE` date DEFAULT NULL,
          `VILLE` varchar(30) DEFAULT NULL,
          PRIMARY KEY (`ID`)
        ) ENGINE=InnoDB AUTO_INCREMENT=13 DEFAULT CHARSET=utf8;

        INSERT INTO `CARNET` VALUES
        (1,'SMITH','JOHN','1980-12-17','ORLEANS'),
        (2,'DURAND','JEAN','1983-01-13','ORLEANS'),
        (3,'GUDULE','JEANNE','1967-11-06','TOURS'),
        (4,'ZAPATA','EMILIO','1956-12-01','ORLEANS'),
        (5,'JOURDAIN','NICOLAS','2000-09-10','TOURS'),
        (6,'DUPUY','MARIE','1986-01-11','BLOIS'),
        (7,'ANDREAS','LOU','1861-02-12','ST Petersbourg'),
        (9,'Kafka','Franz','1883-07-03','Prague'),
        (11,'Dalton','Joe','2003-12-06','Dallas');
```

On insère cette table dans MySQL en ligne de commande ou à l'aide de PHPMyAdmin. Puis, pour consulter cette table depuis PHP, on utilise le connecteur PDO qui offre une interface de connexion utilisable pour tous les SGBD (Systemes de Gestion de Bases de Donnees) habituels comme MySQL, PostgreSQL, Oracle ou Microsoft SQL Server.

3.17.2 Connexion Simple en PHP avec PDO

```
<!doctype html>
<html>
<head>
<title>
Connexion à MySQL avec PDO
</title>
```

(suite sur la page suivante)

(suite de la page précédente)

```
<meta charset="utf-8">
</head>
<body>
<h1>
Interrogation de la table CARNET avec PDO
</h1>

<?php
require("connect.php");
// pour oracle: $dsn="oci:dbname=//serveur:1521/base
// pour sqlite: $dsn="sqlite:/tmp/base.sqlite"
$dsn="mysql:dbname=".BASE.";host=".SERVER;
    try{
      $connexion=new PDO($dsn,USER,PASSWD);
    }
    catch(PDOException $e){
      printf("Échec de la connexion : %s\n", $e->getMessage());
      exit();
    }
$sql="SELECT * from CARNET";
if(!$connexion->query($sql)) echo "Pb d'accès au CARNET";
else{
     foreach ($connexion->query($sql) as $row)
     echo $row['PRENOM']." ".
          $row['NOM']."né(e) le ".
          $row['NAISSANCE']."<br/>\n";
}
?>
</body>
</html>
```

Avec un fichier connect.php contenant les informations de connexion au serveur MySQL :

```
<?php
define('USER',"scott");
define('PASSWD',"tiger");
define('SERVER',"localhost");
define('BASE',"dbscott");
?>
```

Resultat brut html :

```
<!doctype html>
<html>
<head>
<title>
Connexion à MySQL avec PDO
```

(suite sur la page suivante)

(suite de la page précédente)

```
</title>
<meta charset="utf-8">
</head>
<body>
<h1>
Interrogation de la table CARNET avec PDO
</h1>


JOHN SMITHné(e) le 1980-12-17<br/>
JEAN DURANDné(e) le 1983-01-13<br/>
JEANNE GUDULEné(e) le 1967-11-06<br/>
EMILIO ZAPATAné(e) le 1956-12-01<br/>
NICOLAS JOURDAINné(e) le 2000-09-10<br/>
MARIE DUPUYné(e) le 1986-01-11<br/>
LOU ANDREAné(e) le 1900-01-01<br/>
bbb aaané(e) le 2020-04-22<br/>
</body>
</html>
```

Execution

carnet php

Fabrication d'une table HTML avec les résultats :

```
<html>
<head>
<title>
Connexion à MySQL avec PDO
</title>
<meta charset="utf-8">
<body>
<h1>
Interrogation de la table CARNET avec PDO
</h1>

<?php
require("connect.php");
$dsn="mysql:dbname=".BASE.";host=".SERVER;
    try{
      $connexion=new PDO($dsn,USER,PASSWD);
    }
    catch(PDOException $e){
      printf("Échec de la connexion : %s\n", $e->getMessage());
      exit();
```

(suite sur la page suivante)

(suite de la page précédente)

```
    }
$sql="SELECT * from CARNET";
if(!$connexion->query($sql)) echo "Pb d'accès au CARNET";
else{
?>
<table>
<tr> <td> Nom </td> <td> Prénom </td></tr>
  <?php
  foreach ($connexion->query($sql) as $row)
echo "<tr><td>".$row['NOM']."</td><td>".
  $row['PRENOM']."</td></tr>\n";
  ?>
</table>
<?php } ?>
</body>
</html>
```

Résultats bruts :

```
<html>
<head>
<title>
Connexion à MySQL avec PDO
</title>
<meta charset="utf-8">
<body>
<h1>
Interrogation de la table CARNET avec PDO
</h1>


<table>
<tr> <td> Nom </td> <td> Prénom </td></tr>
  <tr><td>SMITH</td><td>JOHN</td></tr>
<tr><td>DURAND</td><td>JEAN</td></tr>
<tr><td>GUDULE</td><td>JEANNE</td></tr>
<tr><td>ZAPATA</td><td>EMILIO</td></tr>
<tr><td>JOURDAIN</td><td>NICOLAS</td></tr>
<tr><td>DUPUY</td><td>MARIE</td></tr>
<tr><td>ANDREA</td><td>LOU</td></tr>
<tr><td>aaa</td><td>bbb</td></tr>
</table>
</body>
</html>
```

Execution

Carnet Table

On peut faire un petit refactoring avec une fonction qui établit la connexion à la base :

```
<?php

require("connect.php");
function connect_bd(){
        $dsn="mysql:dbname=".BASE.";host=".SERVER;
                try{
                $connexion=new PDO($dsn,USER,PASSWD);
                }
                catch(PDOException $e){
                printf("Échec de la connexion : %s\n", $e->
 ↪getMessage());
                exit();
                }
        return $connexion;
}

?>
```

et améliorer l'affichage des résultats :

```
<html>
<head>
<title>
Connexion à MySQL avec PDO
</title>
<meta charset="utf-8">
<link rel="stylesheet" href="tabstyle.css" />
</head>
<body>
<h1>
Interrogation de la table CARNET avec PDO
</h1>

<?php
require_once('connexion.php');
$connexion=connect_bd();
$sql="SELECT * from CARNET";
if(!$connexion->query($sql)) echo "Pb d'accès au CARNET";
else{
?>
<table class="centre" id="jolie">
<tr> <td> ID </td> <td> Prénom </td> <td> Nom </td><td> Naissance </
 ↪td> </tr>
```

(suite sur la page suivante)

(suite de la page précédente)

```
  <?php
  foreach ($connexion->query($sql) as $row)
echo "<tr><td>".$row['ID']."</td>
          <td>".$row['PRENOM']."</td>
          <td>".$row['NOM']."</td>
          <td>".$row['NAISSANCE']."</td></tr><br/>\n";
  ?>
</table>
<?php } ?>
</body>
</html>
```

Avec le fichier CSS :

```
/* Bordure simple autour des tableaux */
table,th, td { border: 1px solid grey;}
table{border-collapse:collapse;}
/* Centrage tableau */
table.centre{ margin:auto;}
/* centrage du texte dans les cellules du tableau */
table.centre td{text-align:center;}

table#jolie tr:first-child{
        background:LightPink;
}
table#jolie tr:nth-child(2n){
        background:#EFD3C9;
}
table#jolie tr:nth-child(2n+3){
        background:#BCBCD0;
}
/* si un tableau a une seule ligne on l'affiche en rouge */
table tr:only-child{
        background:red;
}
```

Résultats bruts :

```
<html>
<head>
<title>
Connexion à MySQL avec PDO
</title>
<meta charset="utf-8">
<link rel="stylesheet" href="tabstyle.css" />
</head>
```

(suite sur la page suivante)

(suite de la page précédente)

```
<body>
<h1>
Interrogation de la table CARNET avec PDO
</h1>


<table class="centre" id="jolie">
<tr> <td> ID </td> <td> Prénom </td> <td> Nom </td><td> Naissance </
↪td> </tr>
  PHP Notice:  Undefined index: ID in /Users/roza/work/iut/prog/php/
↪source/exemples/pdo/cartable2.php on line 25
PHP Stack trace:
PHP   1. {main}() /Users/roza/work/iut/prog/php/source/exemples/pdo/
↪cartable2.php:0

Notice: Undefined index: ID in /Users/roza/work/iut/prog/php/source/
↪exemples/pdo/cartable2.php on line 25

Call Stack:
    0.0002     398656   1. {main}() /Users/roza/work/iut/prog/php/
↪source/exemples/pdo/cartable2.php:0

<tr><td></td>
          <td>JOHN</td>
          <td>SMITH</td>
          <td>1980-12-17</td></tr><br/>
PHP Notice:  Undefined index: ID in /Users/roza/work/iut/prog/php/
↪source/exemples/pdo/cartable2.php on line 25
PHP Stack trace:
PHP   1. {main}() /Users/roza/work/iut/prog/php/source/exemples/pdo/
↪cartable2.php:0

Notice: Undefined index: ID in /Users/roza/work/iut/prog/php/source/
↪exemples/pdo/cartable2.php on line 25

Call Stack:
    0.0002     398656   1. {main}() /Users/roza/work/iut/prog/php/
↪source/exemples/pdo/cartable2.php:0

<tr><td></td>
          <td>JEAN</td>
          <td>DURAND</td>
          <td>1983-01-13</td></tr><br/>
PHP Notice:  Undefined index: ID in /Users/roza/work/iut/prog/php/
↪source/exemples/pdo/cartable2.php on line 25
PHP Stack trace:
PHP   1. {main}() /Users/roza/work/iut/prog/php/source/exemples/pdo/
↪cartable2.php:0
```

(suite sur la page suivante)

(suite de la page précédente)

```
Notice: Undefined index: ID in /Users/roza/work/iut/prog/php/source/
 ↪exemples/pdo/cartable2.php on line 25

Call Stack:
    0.0002     398656   1. {main}() /Users/roza/work/iut/prog/php/
 ↪source/exemples/pdo/cartable2.php:0

<tr><td></td>
          <td>JEANNE</td>
          <td>GUDULE</td>
          <td>1967-11-06</td></tr><br/>
PHP Notice:  Undefined index: ID in /Users/roza/work/iut/prog/php/
 ↪source/exemples/pdo/cartable2.php on line 25
PHP Stack trace:
PHP   1. {main}() /Users/roza/work/iut/prog/php/source/exemples/pdo/
 ↪cartable2.php:0

Notice: Undefined index: ID in /Users/roza/work/iut/prog/php/source/
 ↪exemples/pdo/cartable2.php on line 25

Call Stack:
    0.0002     398656   1. {main}() /Users/roza/work/iut/prog/php/
 ↪source/exemples/pdo/cartable2.php:0

<tr><td></td>
          <td>EMILIO</td>
          <td>ZAPATA</td>
          <td>1956-12-01</td></tr><br/>
PHP Notice:  Undefined index: ID in /Users/roza/work/iut/prog/php/
 ↪source/exemples/pdo/cartable2.php on line 25
PHP Stack trace:
PHP   1. {main}() /Users/roza/work/iut/prog/php/source/exemples/pdo/
 ↪cartable2.php:0

Notice: Undefined index: ID in /Users/roza/work/iut/prog/php/source/
 ↪exemples/pdo/cartable2.php on line 25

Call Stack:
    0.0002     398656   1. {main}() /Users/roza/work/iut/prog/php/
 ↪source/exemples/pdo/cartable2.php:0

<tr><td></td>
          <td>NICOLAS</td>
          <td>JOURDAIN</td>
          <td>2000-09-10</td></tr><br/>
PHP Notice:  Undefined index: ID in /Users/roza/work/iut/prog/php/
 ↪source/exemples/pdo/cartable2.php on line 25
PHP Stack trace:
PHP   1. {main}() /Users/roza/work/iut/prog/php/source/exemples/pdo/
 ↪cartable2.php:0
```

(suite sur la page suivante)

(suite de la page précédente)

```
Notice: Undefined index: ID in /Users/roza/work/iut/prog/php/source/
 ↪exemples/pdo/cartable2.php on line 25

Call Stack:
    0.0002     398656   1. {main}() /Users/roza/work/iut/prog/php/
 ↪source/exemples/pdo/cartable2.php:0

<tr><td></td>
          <td>MARIE</td>
          <td>DUPUY</td>
          <td>1986-01-11</td></tr><br/>
PHP Notice:  Undefined index: ID in /Users/roza/work/iut/prog/php/
 ↪source/exemples/pdo/cartable2.php on line 25
PHP Stack trace:
PHP   1. {main}() /Users/roza/work/iut/prog/php/source/exemples/pdo/
 ↪cartable2.php:0

Notice: Undefined index: ID in /Users/roza/work/iut/prog/php/source/
 ↪exemples/pdo/cartable2.php on line 25

Call Stack:
    0.0002     398656   1. {main}() /Users/roza/work/iut/prog/php/
 ↪source/exemples/pdo/cartable2.php:0

<tr><td></td>
          <td>LOU</td>
          <td>ANDREA</td>
          <td>1900-01-01</td></tr><br/>
PHP Notice:  Undefined index: ID in /Users/roza/work/iut/prog/php/
 ↪source/exemples/pdo/cartable2.php on line 25
PHP Stack trace:
PHP   1. {main}() /Users/roza/work/iut/prog/php/source/exemples/pdo/
 ↪cartable2.php:0

Notice: Undefined index: ID in /Users/roza/work/iut/prog/php/source/
 ↪exemples/pdo/cartable2.php on line 25

Call Stack:
    0.0002     398656   1. {main}() /Users/roza/work/iut/prog/php/
 ↪source/exemples/pdo/cartable2.php:0

<tr><td></td>
          <td>bbb</td>
          <td>aaa</td>
          <td>2020-04-22</td></tr><br/>
</table>
</body>
</html>
```

Execution

Carnet Table Version2

On peut générer des pages différentes avec des listes déroulantes ou des listes de liens, listes à puces etc.

Création d'une liste déroulante :

```
<!doctype html>
<html>
<head>
<title>
Connexion à MySQL avec PDO
</title>
<meta charset="utf-8">
</head>
<body>
<h1>
Interrogation de la table CARNET avec PDO
</h1>
<?php
require_once('connexion.php');
$connexion=connect_bd();
$sql="SELECT * from CARNET";
if(!$connexion->query($sql)) echo "Pb d'accès au CARNET";
else {
      ?>
      <form action="recherche.php" method="GET">
             <select name="ID">
       <?php
       foreach ($connexion->query($sql) as $row)
       if(!empty($row['NOM']))
             echo "<option value='".$row['ID']."'>"
         .$row['PRENOM']." ".$row['NOM']."</option>\n";
       ?>
      </select>
      <input type="submit" value="Rechercher">
      </form>
      <?php
       }
      ?>
</body>
</html>
```

Remarquez l'usage de la clef primaire de la table comme value des options, ce qui assurent l'unicité des valeurs et évite toute amiguité.

Résultats bruts :

```
<!doctype html>
<html>
<head>
<title>
Connexion à MySQL avec PDO
</title>
<meta charset="utf-8">
</head>
<body>
<h1>
Interrogation de la table CARNET avec PDO
</h1>

     <form action="recherche.php" method="GET">
            <select name="ID">
      PHP Notice:  Undefined index: ID in /Users/roza/work/iut/prog/
 ↪php/source/exemples/pdo/carselect.php on line 25
PHP Stack trace:
PHP   1. {main}() /Users/roza/work/iut/prog/php/source/exemples/pdo/
 ↪carselect.php:0

Notice: Undefined index: ID in /Users/roza/work/iut/prog/php/source/
 ↪exemples/pdo/carselect.php on line 25

Call Stack:
    0.0002     398576   1. {main}() /Users/roza/work/iut/prog/php/
 ↪source/exemples/pdo/carselect.php:0

<option value=''>JOHN SMITH</option>
PHP Notice:  Undefined index: ID in /Users/roza/work/iut/prog/php/
 ↪source/exemples/pdo/carselect.php on line 25
PHP Stack trace:
PHP   1. {main}() /Users/roza/work/iut/prog/php/source/exemples/pdo/
 ↪carselect.php:0

Notice: Undefined index: ID in /Users/roza/work/iut/prog/php/source/
 ↪exemples/pdo/carselect.php on line 25

Call Stack:
    0.0002     398576   1. {main}() /Users/roza/work/iut/prog/php/
 ↪source/exemples/pdo/carselect.php:0

<option value=''>JEAN DURAND</option>
PHP Notice:  Undefined index: ID in /Users/roza/work/iut/prog/php/
 ↪source/exemples/pdo/carselect.php on line 25
PHP Stack trace:
PHP   1. {main}() /Users/roza/work/iut/prog/php/source/exemples/pdo/
 ↪carselect.php:0
```

(suite sur la page suivante)

(suite de la page précédente)

```
Notice: Undefined index: ID in /Users/roza/work/iut/prog/php/source/
 ↪exemples/pdo/carselect.php on line 25

Call Stack:
    0.0002     398576   1. {main}() /Users/roza/work/iut/prog/php/
 ↪source/exemples/pdo/carselect.php:0

<option value=''>JEANNE GUDULE</option>
PHP Notice:  Undefined index: ID in /Users/roza/work/iut/prog/php/
 ↪source/exemples/pdo/carselect.php on line 25
PHP Stack trace:
PHP   1. {main}() /Users/roza/work/iut/prog/php/source/exemples/pdo/
 ↪carselect.php:0

Notice: Undefined index: ID in /Users/roza/work/iut/prog/php/source/
 ↪exemples/pdo/carselect.php on line 25

Call Stack:
    0.0002     398576   1. {main}() /Users/roza/work/iut/prog/php/
 ↪source/exemples/pdo/carselect.php:0

<option value=''>EMILIO ZAPATA</option>
PHP Notice:  Undefined index: ID in /Users/roza/work/iut/prog/php/
 ↪source/exemples/pdo/carselect.php on line 25
PHP Stack trace:
PHP   1. {main}() /Users/roza/work/iut/prog/php/source/exemples/pdo/
 ↪carselect.php:0

Notice: Undefined index: ID in /Users/roza/work/iut/prog/php/source/
 ↪exemples/pdo/carselect.php on line 25

Call Stack:
    0.0002     398576   1. {main}() /Users/roza/work/iut/prog/php/
 ↪source/exemples/pdo/carselect.php:0

<option value=''>NICOLAS JOURDAIN</option>
PHP Notice:  Undefined index: ID in /Users/roza/work/iut/prog/php/
 ↪source/exemples/pdo/carselect.php on line 25
PHP Stack trace:
PHP   1. {main}() /Users/roza/work/iut/prog/php/source/exemples/pdo/
 ↪carselect.php:0

Notice: Undefined index: ID in /Users/roza/work/iut/prog/php/source/
 ↪exemples/pdo/carselect.php on line 25

Call Stack:
    0.0002     398576   1. {main}() /Users/roza/work/iut/prog/php/
 ↪source/exemples/pdo/carselect.php:0
```

(suite sur la page suivante)

(suite de la page précédente)

```
<option value=''>MARIE DUPUY</option>
PHP Notice:  Undefined index: ID in /Users/roza/work/iut/prog/php/
 ↪source/exemples/pdo/carselect.php on line 25
PHP Stack trace:
PHP   1. {main}() /Users/roza/work/iut/prog/php/source/exemples/pdo/
 ↪carselect.php:0

Notice: Undefined index: ID in /Users/roza/work/iut/prog/php/source/
 ↪exemples/pdo/carselect.php on line 25

Call Stack:
    0.0002     398576   1. {main}() /Users/roza/work/iut/prog/php/
 ↪source/exemples/pdo/carselect.php:0

<option value=''>LOU ANDREA</option>
PHP Notice:  Undefined index: ID in /Users/roza/work/iut/prog/php/
 ↪source/exemples/pdo/carselect.php on line 25
PHP Stack trace:
PHP   1. {main}() /Users/roza/work/iut/prog/php/source/exemples/pdo/
 ↪carselect.php:0

Notice: Undefined index: ID in /Users/roza/work/iut/prog/php/source/
 ↪exemples/pdo/carselect.php on line 25

Call Stack:
    0.0002     398576   1. {main}() /Users/roza/work/iut/prog/php/
 ↪source/exemples/pdo/carselect.php:0

<option value=''>bbb aaa</option>
     </select>
     <input type="submit" value="Rechercher">
     </form>
     </body>
</html>
```

Execution

Carnet Select

3.18 Requêtes préparées en PHP

3.18.1 Recherche simple en PHP avec PDO

```
<!doctype html>
<html>
<head>
<title>
Recherche d'une personne par ID
</title>
<meta charset="utf-8">
</head>
<body>
<?php $wanted=$_GET['ID'];
if (!empty($wanted)){
        echo "<h1>Recherche de  $wanted </h1>";
        require_once('connexion.php');
        $connexion=connect_bd();
        $sql="SELECT * from CARNET where ID='".$wanted."'";
        if(!$connexion->query($sql))
                echo "Pb de requete";
        else{
        foreach ($connexion->query($sql) as $row)
                echo $row['NOM']." ".$row['PRENOM']."</br>\n";
  }
}
  ?>
</body>
</html>
```

Appel avec le paramètre ID passé sur l'URL : php exemples/pdo/recherche.php ?ID=3

Execution

recherche php

Mais lorsqu'il y a de nombreux paramètres, on se retrouve avec de nombreuses concaténations de chaines entourées de "cotes" ce qui est une grande source d'erreurs et de lenteurs d'écriture. Pour remédier à cela, on peut utiliser des requêtes préparées qui permettent de bien dissocier la requête des paramètres qui vont lui être fournis avant son exécution. Ces *PreparedStatement* seront également bien préférables en termes de sécurité et à utiliser **systématiquement**.

3.18.2 Recherche avec PreparedStatement

```
<!doctype html>
<html>
<head>
<title>
Recherche d'une personne par ID
</title>
<meta charset="utf-8">
</head>
<body>
<?php $wanted=$_GET['ID'];
if (!empty($wanted)){
  echo "<h1>Recherche de  $wanted </h1>";
  require_once('connexion.php');
  $connexion=connect_bd();
$sql="SELECT * from CARNET where ID=:id";
$stmt=$connexion->prepare($sql);
$stmt->bindParam(':id', $_GET['ID']);
$stmt->execute();
if (!$stmt) echo "Pb d'accès au CARNET";
else{
    if ($stmt->rowCount()==0) echo "Inconnu !<br/>";
    else
     foreach ($stmt as $row)
     echo $row['PRENOM']." ".$row['NOM'].
          "né(e) le ".$row['NAISSANCE']."<br/>";
}
}
?>
</body>
</html>
```

Les requêtes préparées limitent fortement la possibilité d'*injections SQL* comme nous le verront plus tard.

3.19 Compléments sur PDO - Sécurité

3.19.1 Filtrage d'entrées

On peut vouloir nourrir une requête directement avec des données provenant d'un formulaire :

```
<?php
  $sql = sprintf(
    'SELECT id FROM CARNET WHERE email = "%s"',$_GET['email'])
  );
```

On voit bien que la valeur de l'entrée email dans le tableau _GET n'est absolument pas vérfifiée

avant son utilisation !

On peut essayer dans ce cas d'utiliser un filtre PHP pour contrôler un peu les choses :

```
<?php
  $sql = sprintf(
      'SELECT id FROM CARNET WHERE email = "%s"',
      filter_input(INPUT_GET, 'email')
  );
```

Mais ceci constitue une manière peu sûre de procéder malgré l'utilisation du filtre PHP. Cela laisse en effet la possiblilité d'insertion de code malveillant non contrôlé.

L'exemple classique est la requête SQL construite dans la chaine suivante :

```
<?php
  $sql = "SELECT nom FROM USERS WHERE login='".
         $_REQUEST['login']."' AND PASSWD='".
         $_REQUEST['pass']."'";
```

Qui donne lors de son exécution avec $_REQUEST[“login”] = » toto” – » :

```
SELECT nom FROM USERS WHERE login='toto'  -- ' AND PASSWD= '".$_
↪REQUEST['pass']."' ";
```

Avertissement : Le - - constituant un début de commentaire SQL, ceci constitue une injection SQL qui est l'une des principales failles de sécurité exploitées par les Hackers. Pour s'en prémunir, il faut utiliser **à la fois** le filtrage des entrées et les requêtes préparées.

```
<?php
$sql = 'SELECT id FROM CARNET WHERE email = :email';
$statement = $pdo->prepare($sql);
$email = filter_input(INPUT_GET, 'email');
$statement->bindValue(':email', $email);
```

Il faut parfois préciser dans un troisième argument le type des paramètres attendus :

```
<?php
  $sql = 'SELECT email FROM CARNET WHERE id = :id';
  $statement = $pdo->prepare($sql);
  $userId = filter_input(INPUT_GET, 'id');
  $statement->bindValue(':id', $userId, PDO::PARAM_INT);
```

3.20 Filtrage en PHP

Les vérifications dans les formulaires HTML5 et en JavaScript sont valables uniquement côté client. Pour des raisons de sécurité, il faut réeffectuer complètement toutes les vérifications côté

serveur. PHP met pour cela à la disposition des programmeurs tout un ensemble de filtres. La première des vérifications consiste à bien prendre en compte les caractères spéciaux.

3.20.1 Gestion des caractères spéciaux :

Les « magic quotes » :

Avant PHP 5.4, une directive concernant ces magic quotes existait. Si dans le fichier de configuration de PHP, la directive magic_quotes_gpc etait activée, PHP modifiait automatiquement certaines entrées de formulaires en procédant à des protections de certains caractères spéciaux par des insertion de « backslashes ». Par exemple

- les caractères accentués
- les apostrophes
- les backslashes

Un peu à la manière de la fonction addslashes().

Cette protection était destinée à préparer les données avant des requêtes SQL pour empêcher une éventuelle injection SQL. Ce comportement automatique est toutefois parfois gênant si on veut simplement réafficher les chaines saisies et non pas les utiliser dans des requêtes SQL. En outre, on ne veut pas toujours protéger les chaines de la même façon selon l'usage qu'on veut en faire par la suite.

On peut vouloir dans certains cas, protéger des chaines par exemple :

- htmlspecialchars() pour éviter des injections de code HTML
- PDO : :quote() pour se protéger d'injections SQL

```
<?php
  $pdo = new PDO('sqlite:./tmp/mydb.sqlite');
  $string = 'Orléans';
  print "Chaîne sans quotes: $string\n";
  print "Chaîne avec quotes: " . $pdo->quote($string) . "\n";
```

Cela ne vous dispense pas d'utiliser les *PreparedStatement* vus précédemment.

Les filtres PHP :

Les plus directs à utiliser sur les formulaires sont basés sur la fonction filter_input() avec en paramètre INPUT_GET ou INPUT_POST Voici quelques exemples typiques :

```
<?php
    $entier = filter_input(INPUT_POST, 'var1', FILTER_SANITIZE_
↪NUMBER_INT);

    $url = filter_input(INPUT_POST, 'var2', FILTER_VALIDATE_URL);

    if (!filter_input(INPUT_GET, "email",  FILTER_VALIDATE_EMAIL))
        echo("Email non valide");
```

(suite sur la page suivante)