

# Outil pour le Web

## 7 - PHP

M.E. Bellemare, **E. Tranvouez**

[erwan.tranvouez@polytech.univ-mrs.fr](mailto:erwan.tranvouez@polytech.univ-mrs.fr)

<http://erwan.tranvouez.free.fr>

***Polytech'Marseille***

**Département Génie Industriel et Informatique**

- Problématique
  - Contexte, Objectifs, Principe général
- Langage PHP
  - Syntaxe, intégration avec HTML
- Lien avec une base de donnée
  - Connexion et requête, formulaires...

**Nota:** le code a été testé. Il fonctionne. Avant de signaler une erreur de code (possible entre 2 versions de test) vérifier que l'exportation des sources Support -> source Php n'ait pas entraîné des erreurs de formats des " ou « » liées à l'auto-correction de PowerPoint.

# Bibliographie/Webliographie

- Pratique de MySQL et PHP, Philippe Rigaux, O'Reilly, 2<sup>ème</sup> éditions.
- <http://php.developpez.com> : documents, tutoriaux et revues d'ouvrages sur php.
- <http://www.php.net> : site officiel de php, contient notamment une documentation très claire sur le langage, dont certains exemples ont été repris dans ce cours.
- <http://www.easyphp.org> : environnement de test/prototypage php. Comprends un serveur apache avec php et un serveur mySQL. Pour tester le code de ce support, créer un alias sur un répertoire créé dans [chemin\_easy\_php]/www et y copier les fichiers php.

---

# 1. Problématique

---

Motivations et principes

# Rappel « historique »

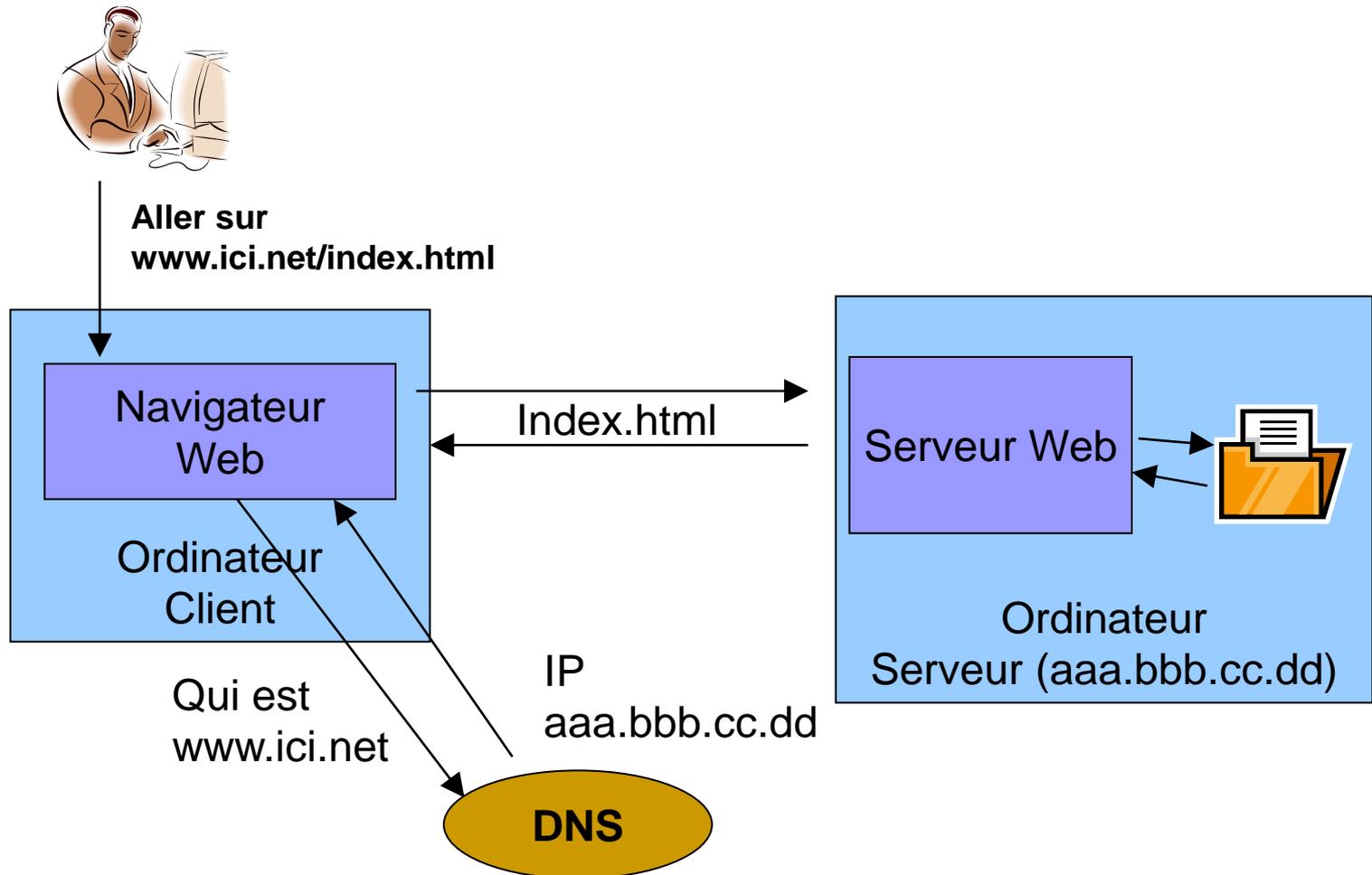
- Évolution du web :
    - Plus de contenu
    - Plus de richesse dans le contenu : passage d'un mode statique (contenu défini une fois pour toute) à un mode dynamique (contenu dépendant des besoins d'informations des utilisateurs web)
    - Programmation Web devient une programmation comme une autre : avec par exemple interfaçage avec une Base de Données
    - => On retrouve les mêmes objectifs que pour tout développement d'application
      - Séparation données traitement (web dynamique)
      - Séparation application / présentation
-

# Rappel du fonctionnement d'un serveur web

- Le client (un navigateur internet) fait des requêtes de fichiers HML sur un serveur internet.
    - ⇒ Demander une information = demander 1 page dans laquelle l'information est déjà encodée
    - ⇒ Faire un site « dynamique » revient à prévoir 'n' version d'une page...
    - ⇒ Évidemment solution lourde et impraticable
  - HTML = langage de structuration de contenu
    - type information, style par défaut, ...
    - Mais pas de programmation (pas de fonction de calcul et transformation hormis la capacité à produire des requêtes à un serveur Web via des formulaires )
-

# Rappel du fonctionnement d'un serveur web

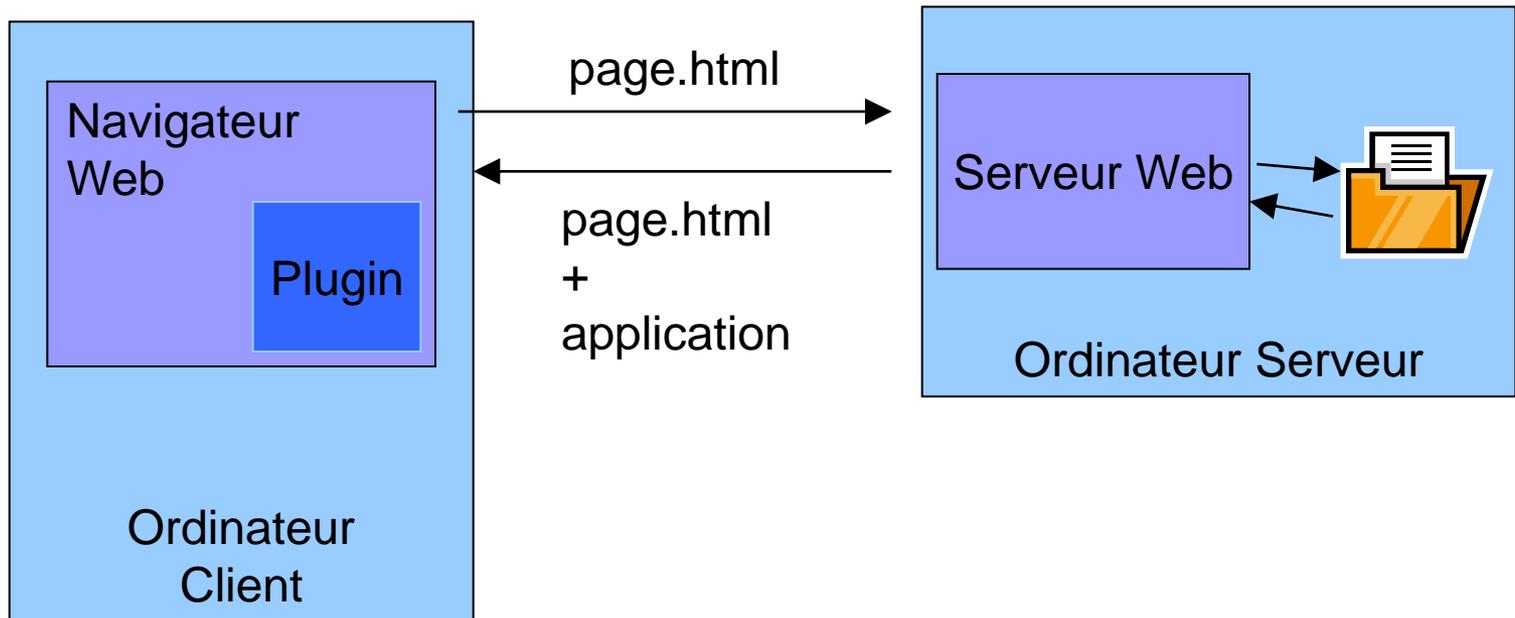
7  
71



# Solutions pour des pages paramétrables

- Elle peut se trouver du **côté client**

- ⇒ le navigateur web client dispose de toutes les informations pour répondre au besoin de l'utilisateur
- ⇒ suppose le téléchargements d'informations non HTML

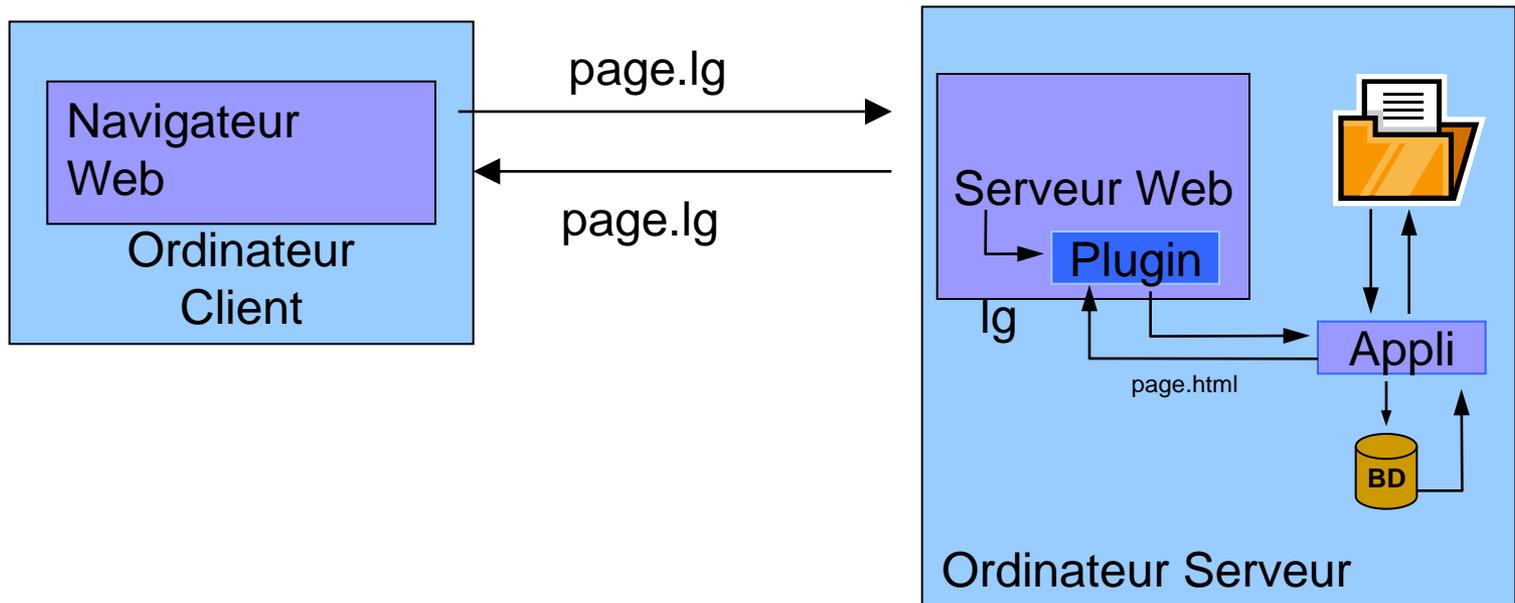


- **JavaScript :**
    - Code téléchargé avec la page web et exécuté localement lors de la lecture de la page. L'information transformable doit donc déjà être disponible (ex. avec CSS modification de la visibilité d'un composant).
  - **Java :**
    - Code intégré dans la page (aspect non HTML)
    - Possibilité d'exécuter des requêtes BD
    - Suppose l'existence d'une Machine Virtuelle Java (acquis)
    - Code compilé peut être lourd à télécharger
  - **Flash, etc...**
    - Richesse de mise en forme (par rapport au débuts des applets)
    - Même inconvénient que Java
-

# Solutions pour des pages paramétrables

10  
71

- Elle peut se trouver du **côté serveur** :
  - ⇒ le serveur calcule et renvoie l'information spécifique demandée par le client (sous forme HTML)



- C'est le serveur qui calcule la page HTML :
    - Le client ne fait que recevoir un fichier HTML => pas de contraintes sur le client
  - CGI (Common Gateway Interface) : le serveur transmet la requête HTML vers un programme dédié qui calcule et renvoie du code HTML au serveur
  - Le programme peut être en
    - Code binaire (ie compilé) : ex. C, C++, Pascal
    - Langage de script : interprété par un programme dédié : ex. Bash, Perl, PHP, JSP, ASP...
  - Suppose
    - que le serveur web est capable d'interagir avec ces programmes
    - une ressource avec capacité de calcul rapide (connexion avec plusieurs clients)
    - Donc application serveur optimisée
-

# Illustration : évolution commerce en ligne

- Capacité à voir un catalogue et à créer une commande
    - Solution HTML: page complète avec lien mail (au client d'indiquer ce qu'il veut)
    - Solution HTML+JavaScript : idem avec formulaire
    - Solution CGI : Programme C + HTML + JavaScript
    - Solution Scripts : Script + HTML + JavaScript
-

# Principe de fonctionnement

13  
71

## ■ Illustration avec le langage C

- Si on voulait faire un script CGI en C, il faudrait créer un programme qui :
  - Respecte la spécification du standard CGI (Common Gateway Interface) qui assure que le programme côté serveur saura utiliser les informations que lui transmettra le serveur web
  - Doit générer du code HTML : donc produira des chaînes de caractères commençant par "<HTML>" et finira par "</HTML>"
  - Et au milieu au moins "<BODY>... "</BODY>"
  - Et bien sur compiler le programme et l'uploader sur le serveur
- L'idée est que le programme pourra personnaliser, en fonction des paramètres, ce qui se trouve entre les 2 balises BODY

## ■ Envoi de paramètres :

- peut se fait en mode GET, avec par exemple une URL du type :
- [http://www.unsite.com/une\\_page.cgi?param=etudiant](http://www.unsite.com/une_page.cgi?param=etudiant)
- Le programme enverra au programme CGI la variable 'param' et sa valeur 'etudiant'.
- Exemple de script CGI en :  
<http://www.ftls.org/fr/initiation/cgi/index5.shtml#s5.3.3>

# Exemple simple en C

14  
71

## ■ Exemple simplifié en C

- On veut que le programme produise du HTML
  - Vous connaissez le fameux printf :

```
/* code source de program.c */  
main(int argc, char *argv[])  
{ printf("vous avez exécuté <%s>,\n avec  
comme paramètre <%s>.", argv[0], argv[1] ); }
```



```
C:>program test  
vous avez exécuté <program>,  
avec comme paramètre <test>.
```

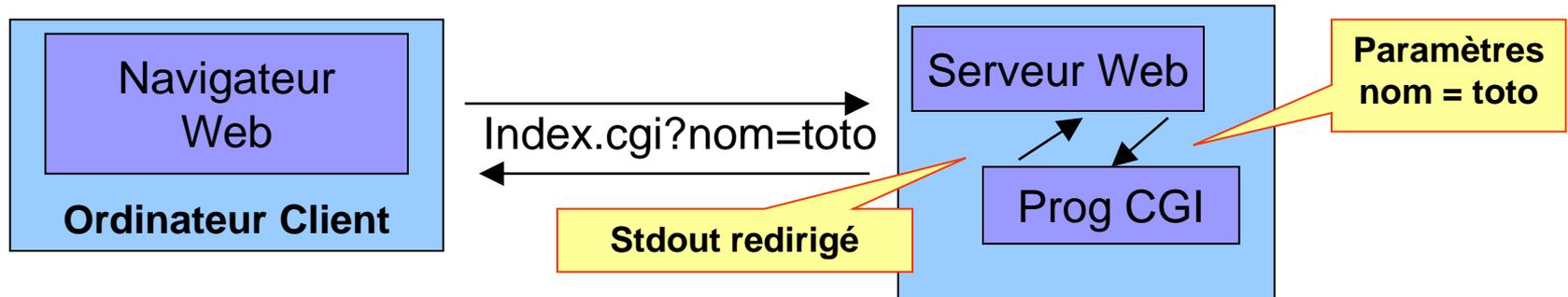
- printf envoie la chaîne de caractère vers la sortie standard (stdout) c'est-à-dire la Console (exemple ci-dessus)
- Pour un programme CGI, il suffit de rediriger stdout vers la sortie CGI. A l'image de ce qu'on fait avec fprintf() qui permet d'écrire dans un FILE \* (donc un fichier), cette sortie peut écrire dans un flux réseau (imaginer un tube entre le navigateur et le serveur web dans lequel circulera des chaînes de caractères).
- On aura donc plein de printf qui contiendront du code HTML ...

# Exemple simple en C

15  
71

## ■ Suite exemple avec C

- Le client (navigateur web) envoie une demande de page avec un paramètre (par ex. via un formulaire web)



- La librairie CGI du C (cgi-util) contient des fonctions permettant de récupérer des informations depuis le serveur web (ex. getenv()...) Considérons que le programme a réussi à récupérer la valeur du paramètre 'nom' (ex. getentry(name, "Name") )

```
// Extrait de code C
char s[100] ;
// ... code spécifique librairie CGI
getentry(s, "nom"); /* met la valeur
'toto' dans la variable s */
printf("<HTML><BODY>\n");
printf("<p> Hello %s! </p>\n"; s );
printf("</BODY></HTML>\n");
```

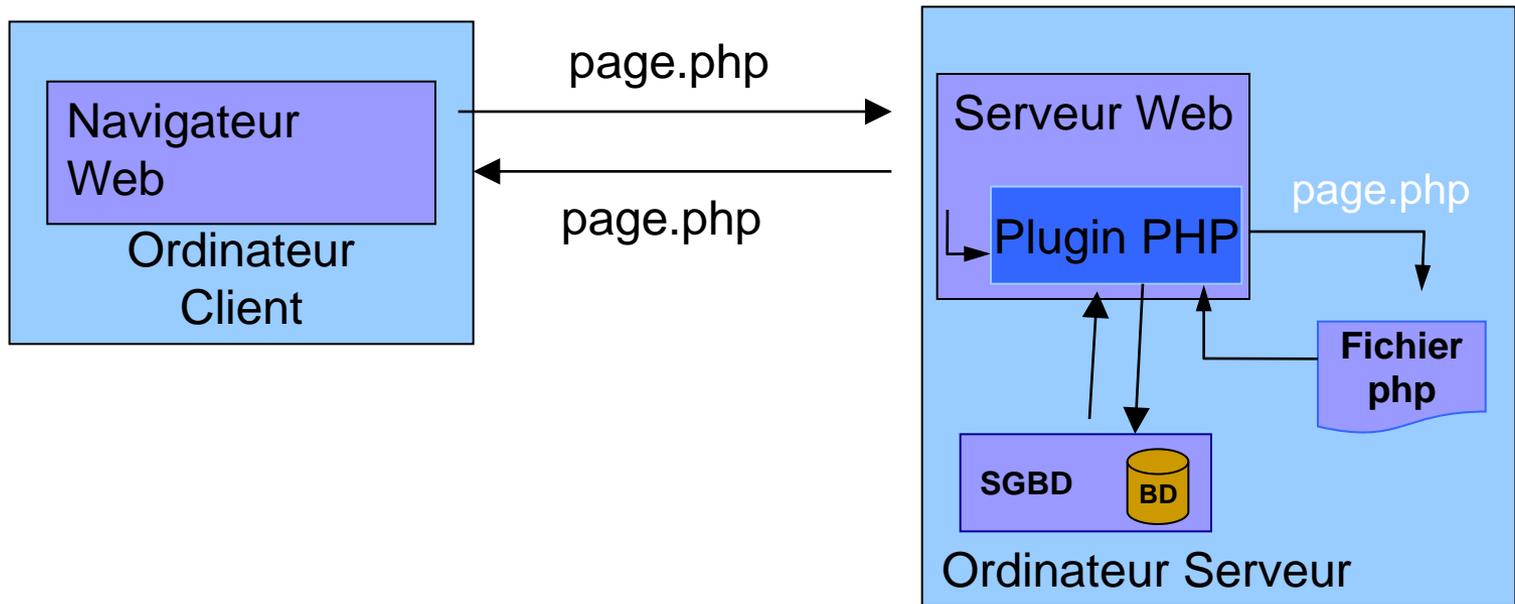
Envoie au client

```
<HTML><BODY>
<p>Hello toto!</p>
</HTML></BODY>
```

```
// equivalent php
<HTML><BODY>
<?php
echo '<p> Hello ' . $_GET['nom'];
echo '! </p>\n' ;
?>
</BODY></HTML>");
```

# Architecture PHP/MySQL

16  
71



## ■ Combinaison classique :

- ❑ Serveur Web : Apache
- ❑ Plugin PHP
- ❑ SGBD : MySQL (mais aussi PostgreSQL...)

# Le pourquoi du succès de PHP

17  
71

- Solution basée sur des logiciels libres en opposition avec le couple Microsoft IIS/ASP
  - Solution plus souple que l'équivalent Java/JSP
-

---

## 2. Programmer en PHP

---

# Propriétés du langage PHP

- PHP est un langage de script
    - Pas de déclaration de type des variables
    - Manipulation et conception d'objets
  - PHP est associé avec un grand nombre de bibliothèques augmentant la capacité de calcul ou de transformation d'information et simplifiant d'autant le travail du programmeur
  - Solution basée sur des logiciels libres en opposition avec le couple Microsoft IIS/ASP mais reste ouvert à ces environnements
-

# Exemple : code php

20  
71

```
<HTML>                                <!-- index.php ->
<HEAD>
  <LINK rel="stylesheet" type="text/css" href="tableau.css" >
<TITLE>Mixité PHP/HTML</TITLE>
</HEAD>
<BODY>
  <H1>Manipulation de tableau</H1>
  <?php
    echo "<BR> ";
    echo "<B>Hello IP :". $_SERVER['REMOTE_ADDR'] . "</B><BR>";
    echo "<B>You're using :". $_SERVER['HTTP_USER_AGENT'] . "</B>";
    echo "<BR>";
  ?>
  <P>__ Fin Code PHP __</P>
</BODY>
</HTML>
```

Marqueur php indiquant le début de la zone de code php (code propre a php)

Seule cette partie du code sera évaluée par l'interpréteur php. Le reste sera directement répété à l'identique.

Idem pour la fin de zone

# Exemple : zoom sur le code php

21  
71

**\$\_SERVER** Variable globale contenant dans un tableau les variables d'environnement du serveur

Élément du tableau contenant l'adresse IP du client web

```
<?php
echo "<BR> ";
echo "<B>Hello IP : " . $_SERVER['REMOTE_ADDR'] . "</B><BR>";
echo "<B>You're using : " . $_SERVER['HTTP_USER_AGENT'] . "</B>";
echo "<BR>";
?>
```

**echo** : affiche (ie écrit dans le flux standard donc au final dans le fichier HTML renvoyé au client) la chaîne de caractère qui suit. Le caractère '.' permet de concaténer 2 chaînes

Élément du tableau contenant un descriptif du navigateur se connectant sur cette page

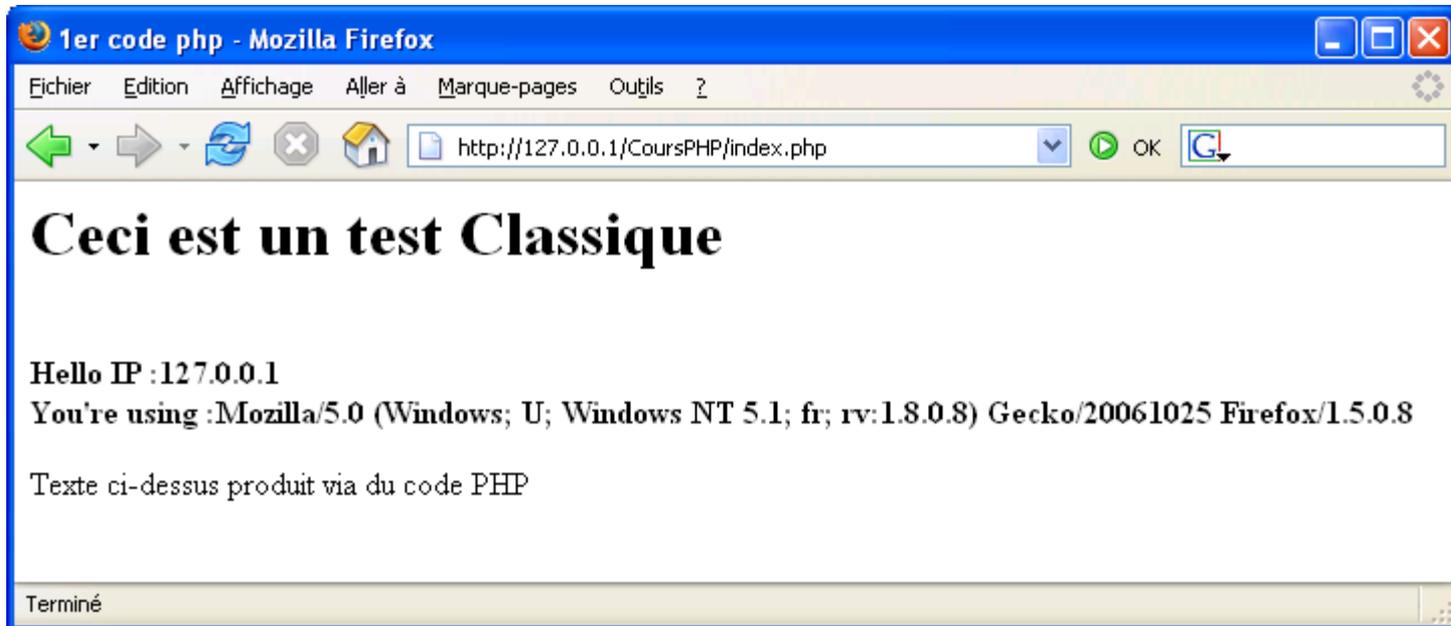
traduction  
php

```
<BR>
<B>Hello IP : 127.0.0.1</B><BR>
<B>You're using : Mozilla/5.0 (Windows; U; Windows NT 5.1; fr; rv:1.8.0.8) Gecko/20061025
Firefox/1.5.0.8</B>
<BR>
```

# Résultat à l'affichage

22  
71

- Au final : le client web ne voit que du code HTML ... il n'a donc aucun moyen de savoir comment ce code à été produit (hormis via l'extension du fichier) ....



# Résultat à l'affichage

23  
71

- ... comme le prouve le code source de la page web :



```
<HTML>
  <HEAD>
    <TITLE>1er code php</TITLE>
  </HEAD>
  <BODY>
    <H1>Ceci est un test Classique</H1>
    <BR> <B>Hello IP :127.0.0.1</B><BR><B>You're using :Mozilla/5.0 (Windows; U;
Windows NT 5.1; fr; rv:1.8.0.8) Gecko/20061025 Firefox/1.5.0.8</B><BR>
    <P>Texte ci-dessus produit via du code PHP</P>
  </BODY>
</HTML>
```

- Pas de déclaration de type !
- Le nom d'une variable est préfixée par le caractère \$
- Les variables globales (notamment celles provenant de l'environnement) sont notées en majuscules.
- Chaînes de caractères :
  - Opérateur '.' surchargé (comme + avec JavaScript)
    - `$ch2 = $ch1 . " est la"` : concatène les 2 chaînes dans \$ch2
  - Délimitées par "chaine" ou 'autrechaîne'
    - " est préféré car au contraire de "" le contenu de la chaîne ne sera pas interprétée par php.
      - Ex. `"Bonjour $nom"` est équivalent à `'Bonjour ' . $nom`

- Un tableau php peut fonctionner :
    - comme un tableau classique (indexation des éléments via un indice entier `$tab[i]` ) ( $i \in [0, \text{taille\_tableau}-1]$ )
    - Comme une table de hashage ou l'index peut être une chaîne `$tab['albert']`.
  - La déclaration d'un tableau se fait
    - à son initialisation :
      - `$tab = array();` // équivalent d'un malloc en C
      - `$tab = array('tableau', 'de', 'chaîne');`
      - `$tab = array('albert' => 20, 'geraldine' => 25, 'alex ' => 33);`
  - Remplir un tableau
    - À la position `i` : `$tab[i]=$val`
    - À la fin : `$tab[] = $val;`
  - Utilisation de fonction dédiées :
    - `array_push()`, `array_slice()`, `array_merge()`, `array_fill()` ...
    - Lister les « clés » d'indexation : **`array_keys($tab)`** renvoie un tableau contenant les indexs alphanumériques de `$tab`
-

# Variables d'environnement

- **PHP s'exécutant sur le serveur, accède aux informations suivantes**

Tableau associatif	Contenu
<code>\$_SERVER</code>	Variables CGI : <code>SERVER_NAME</code> , <code>REQUEST_METHOD</code> , <code>QUERY_STRING</code> , <code>HTTP_USER_AGENT</code> , <code>REMOTE_ADDR</code> , <code>REMOTE_USER</code> , <code>REMOTE_PASSWORD</code> ...
<code>\$_ENV</code>	Variables d'environnement système
<code>\$_GET</code> (resp. <code>\$_POST</code> )	Paramètres HTTP passé en mode GET (resp. POST) ie valeur d'un formulaire
<code>\$_SESSION</code>	Variable de session php
...	

# Exemple : \$\_SERVER

COMSPEC	C:\WINDOWS\system32\cmd.exe
DOCUMENT_ROOT	c:/program files/easyphp1-8/www
HTTP_ACCEPT	text/xml,application/xml,application/xhtml+xml;text/html;q=0.9;text/plain;q=0.8,image/png,*/*;q=0.5
HTTP_ACCEPT_CHARSET	ISO-8859-1,utf-8;q=0.7,*;q=0.7
HTTP_ACCEPT_ENCODING	gzip,deflate
HTTP_ACCEPT_LANGUAGE	fr-fr;q=0.8,en-us;q=0.5,en;q=0.3
HTTP_CACHE_CONTROL	max-age=0
HTTP_CONNECTION	keep-alive
HTTP_HOST	127.0.0.1
HTTP_KEEP_ALIVE	300
HTTP_USER_AGENT	Mozilla/5.0 (Windows; U; Windows NT 5.1; fr; rv:1.8.0.8) Gecko/20061025 Firefox/1.5.0.8
PATH	C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\Program Files\Fichiers communs\GTK2.0\bin;C:\Program Files\QuickTime\QTSystem\;C:\Program Files\ES Tsoft\ALZip\;C:\Program Files\VDMSound\Cdev\Sun\App Server\jdk\bin;C:\dev\Sun\App Server\bin;C:\Program Files\ES Tsoft\ALZip\;C:\PROGRA~1\EASYPH~1\Apache;C:\PROGRA~1\EASYPH~1\PHP
REMOTE_ADDR	127.0.0.1
REMOTE_PORT	1377
SCRIPT_FILENAME	c:/program files/easyphp1-8/www/coursphp/var_environnement.php
SERVER_ADDR	127.0.0.1
SERVER_ADMIN	admin@localhost
SERVER_NAME	127.0.0.1
SERVER_PORT	80
SERVER_SIGNATURE	Apache/1.3.33 Server at 127.0.0.1 Port 80
SERVER_SOFTWARE	Apache/1.3.33 (Win32) PHP/4.3.10
SystemRoot	C:\WINDOWS
WINDIR	C:\WINDOWS
GATEWAY_INTERFACE	CGI/1.1
SERVER_PROTOCOL	HTTP/1.1
REQUEST_METHOD	GET
QUERY_STRING	
REQUEST_URI	/CoursPHP/Var_environnement.php
SCRIPT_NAME	/CoursPHP/Var_environnement.php
PATH_TRANSLATED	c:/program files/easyphp1-8/www/coursphp/var_environnement.php
PHP_SELF	/CoursPHP/Var_environnement.php

```
<?php // code php correspondant
$keys=array_keys($_ENV);
echo "<table style='font-size:x-small;'>";
for( $i=0; $i<count($keys); $i++){
    echo "<TR><TD>$keys[$i]</TD>";
    echo "<TD>" . $_ENV[$keys[$i]] .
"</TD></TR>";
}
echo "</table>";
?>
```

# Structures de contrôle : for, while ...

28  
71

- Syntaxe similaire à celle du C
- Condition :
  - `if($a>$max)`
- Boucle for :
  - `for ($i = 0; $i <= 10; $i+=$pas) {...}`
  - `for ($i = 0; $i <= 10; $i++) {...}`
- while & compagnie
  - `do {...} while ($toto != 'ok')`
  - `while( !$arreter ) { ... }`
- switch - case :
  - même syntaxe que C avec en plus la capacité d'utiliser des chaînes.

```
switch( $val) {
    case "albert":
        echo "Cher ami ";
    case "tous" : // tous les cas
        echo "Bonjour ";
        break;
    default:
        echo "connais pas";
}
```

- PHP propose une librairie riche offrant une large palette de fonctions pour développer des pages dynamiques :
    - Gestion/envoi de mail (remplace l'infâme mailto:)
    - Interaction avec systèmes de fichiers du serveur (ouverture, ftp, ...)
    - Fonctions de connexion/requêtes à des SGBD
      - MySQL
      - PostgreSQL
      - (MS) SQL Server ...
    - Fonction de manipulation de fichiers XML
    - Fonctions COM / .Net
    - Fonctions de génération de fichier pdf **ClibPDF**
    - ... avec évidemment des fonctions standards de calculs, manipulation de chaînes et de tableau ...
-

# Exemple de manipulation de code *php*

30  
71

```
<?php
```

```
function compte_a_dans_b($a='a', $b) { /* code de la page précédente */ }
```

```
$stab = array('java' => 'cours/coursJava.pdf', 'web' => 'cours/coursWeb.pdf');
```

```
$stab['agl']='cours/coursAGL.pdf';
```

```
echo "<BR>affichage du contenu de $stab de type " . gettype($stab) . " <BR>";
```

```
echo " contenant " . sizeof($stab) . " elements." ;
```

```
print_r($stab);
```

```
echo "<BR>";
```

```
$test='pipo';
```

```
if(!array_key_exists($test, $stab)) { echo "Le cours " . $test . " n'existe pas!<BR>"; }
```

```
echo "Liste des clefs";
```

```
print_r( array_keys($stab) );
```

```
echo "<BR>";
```

```
$stab_string = array( 'albert', 'geraldine', 'chris', 'albert' );
```

```
$str = 'albert';
```

```
echo 'Contenu de $stab_string : \n' ;
```

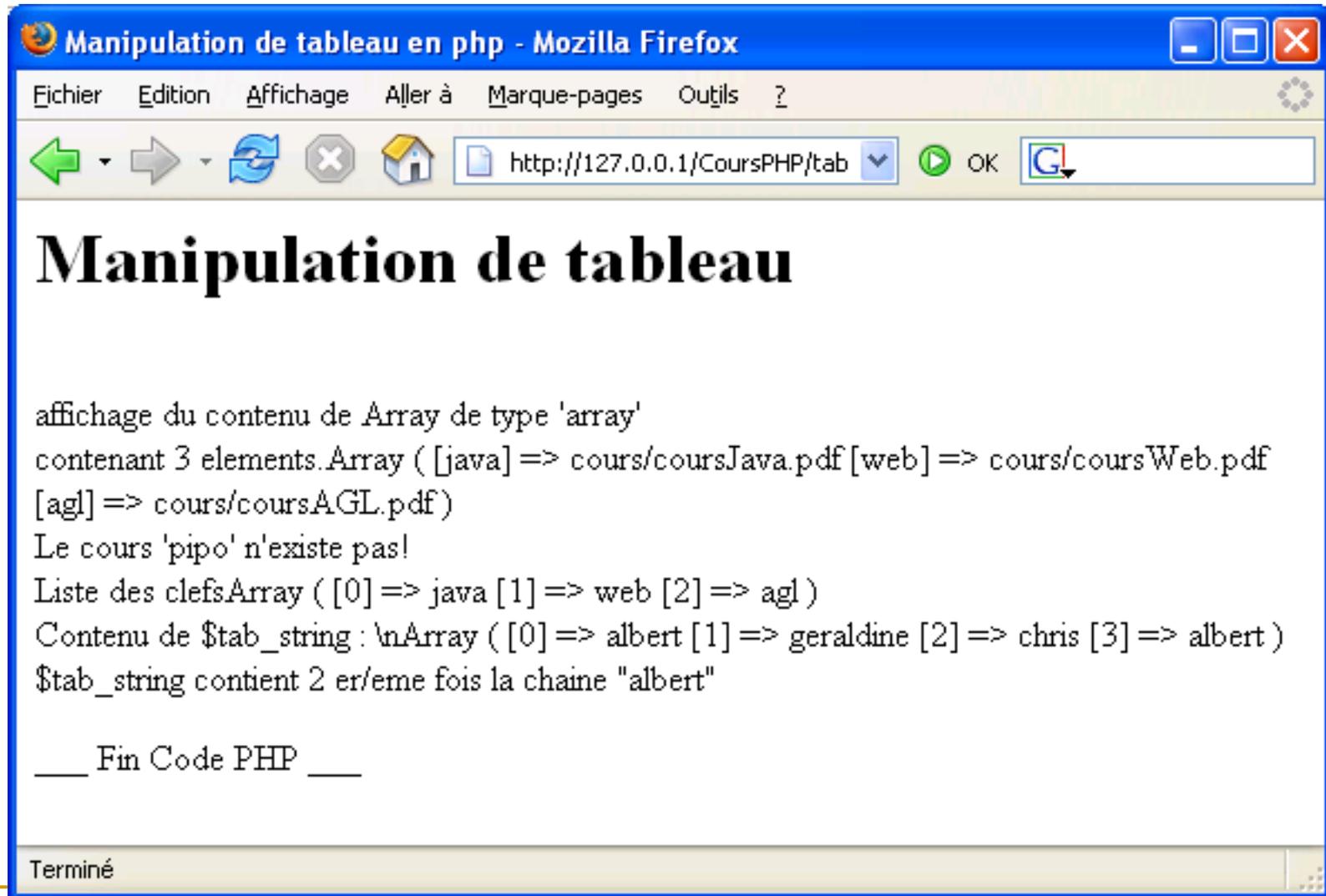
```
print_r($stab_string);
```

```
echo '$stab_string contient ' . compte_a_dans_b( $str, $stab_string). ' er/eme fois la chaine "' . $str . "'";
```

```
?>
```

# Résultat à l'affichage

31  
71



Manipulation de tableau en php - Mozilla Firefox

Fichier Edition Affichage Aller à Marque-pages Outils ?

http://127.0.0.1/CoursPHP/tab

## Manipulation de tableau

affichage du contenu de Array de type 'array'  
contenant 3 elements.Array ( [java] => cours/coursJava.pdf [web] => cours/coursWeb.pdf  
[agl] => cours/cours.AGL.pdf )  
Le cours 'pipo' n'existe pas!  
Liste des clefsArray ( [0] => java [1] => web [2] => agl )  
Contenu de \$tab\_string : \nArray ( [0] => albert [1] => geraldine [2] => chris [3] => albert )  
\$tab\_string contient 2 er/eme fois la chaine "albert"

\_\_\_ Fin Code PHP \_\_\_

Terminé

# Lien entre tableau PHP & tableau HTML

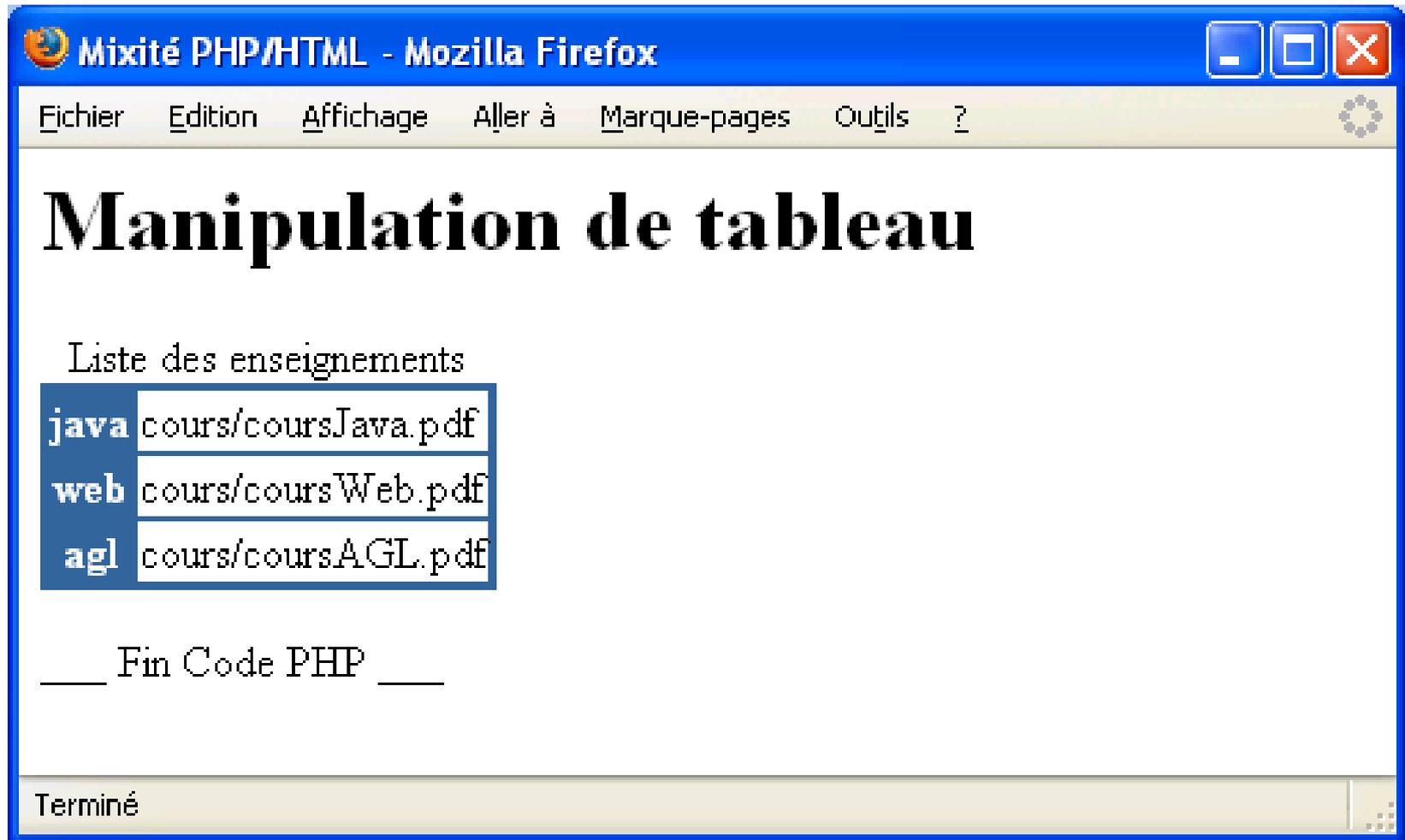
- Revient à séparer Données / Présentations
- Facilité de production du code HTML via des boucles
- Ex

```
function formate_tableau($tab,$legende=""){
    if( !is_array($tab) )    return;
    $titre = array_keys($tab);

    $tableau_html='<table><caption>' . $legende . '</caption>';
    for($i=0; $i<count($tab); $i++){
        $tableau_html.='<TR><td class="titre">' . $titre[ $i ] . '</TD>';
        $tableau_html.='<TD>' . $tab [ $titre[ $i ] ] . '</TD></TR>';
    }
    $tableau_html.='</table>';
    return $tableau_html;
}
```

# Résultat à l'affichage

33  
71



The screenshot shows a Mozilla Firefox browser window with the title "Mixité PHP/HTML - Mozilla Firefox". The menu bar includes "Fichier", "Edition", "Affichage", "Aller à", "Marque-pages", "Outils", and "?". The main content area displays the output of a PHP script:

## Manipulation de tableau

Liste des enseignements

java	cours/coursJava.pdf
web	cours/coursWeb.pdf
agl	cours/coursAGL.pdf

\_\_\_ Fin Code PHP \_\_\_

Terminé

# Code HTML produit

34  
71

```
<HTML>
  <HEAD>
    <LINK rel="stylesheet" type="text/css" href="tableau.css">
    <TITLE>Mixité PHP/HTML</TITLE>
  </HEAD>
  <BODY>
    <H1>Manipulation de tableau</H1>
    <table><caption>Liste des enseignements</caption>
      <TR>
        <td class="titre">java</TD>
        <TD>cours/coursJava.pdf</TD>
      </TR>
      <TR>
        <td class="titre">web</TD>
        <TD>cours/coursWeb.pdf</TD>
      </TR>
      <TR>
        <td class="titre">agl</TD>
        <TD>cours/coursAGL.pdf</TD>
      </TR>
    </table>
    <P>___ Fin Code PHP ___</P>
  </BODY>
</HTML>
```

# Profiter de la mixité PHP/HTML

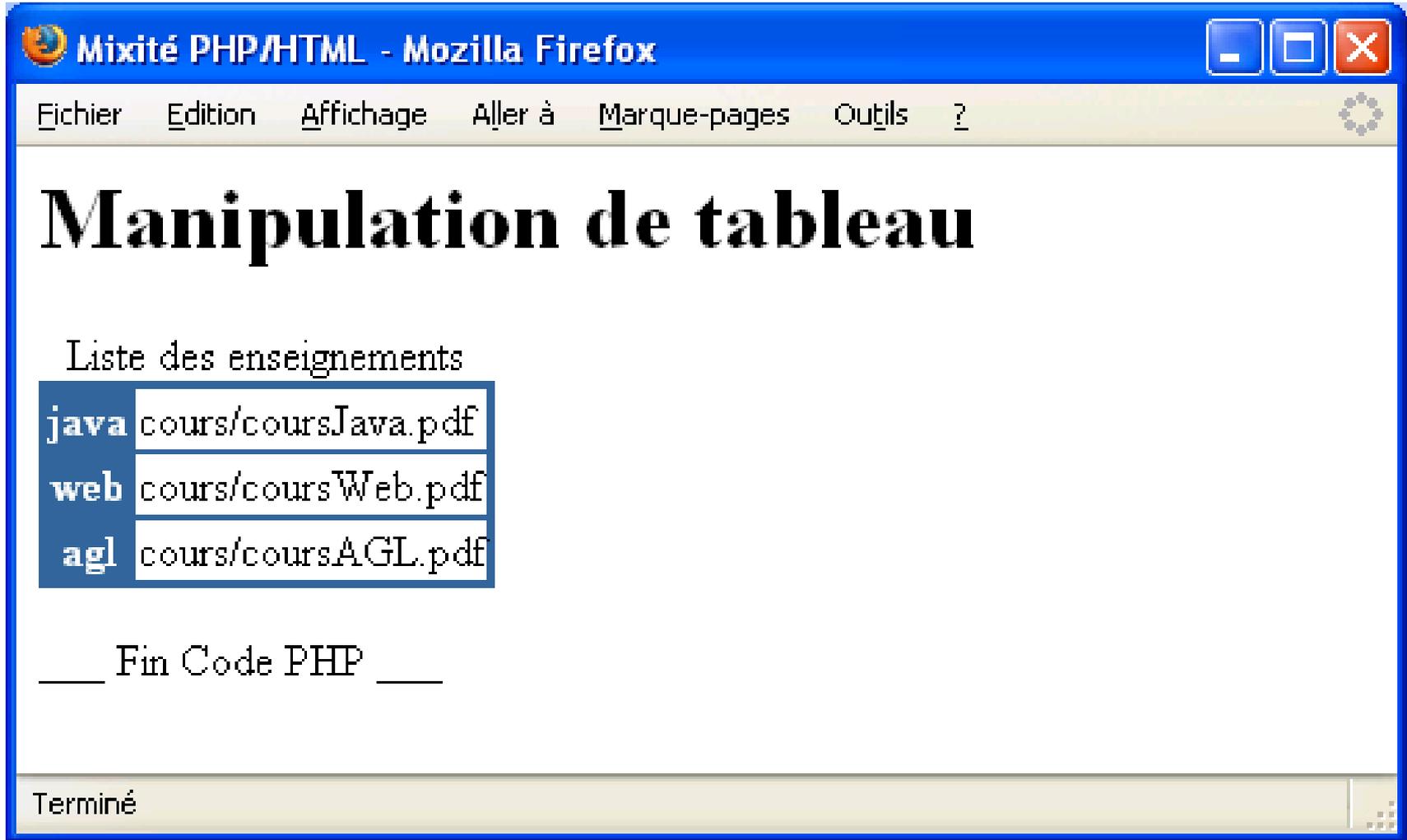
35  
71

```
<HTML>                                <!-- tableau.php -->
<HEAD>
  <LINK rel="stylesheet" type="text/css" href="tableau.css">
  <TITLE>Mixité PHP/HTML</TITLE>
</HEAD>
<BODY>
  <H1>Manipulation de tableau</H1>
  <?php
function formate_tableau($tab,$legende=""){ // cf code précédent  }
$stab = array('java'=>'cours/coursJava.pdf', 'web'=>'cours/coursWeb.pdf', 'agl'=>'cours/coursAGL.pdf');
echo 'affichage contenu de $tab dans un tableau';
echo
echo '<BR>' . formate_tableau($tab);
?>
<P>___ Fin Code PHP ___</P>
</BODY>
</HTML>
```

```
table {                                // tableau.css
border:2px solid blue;
border-spacing: 0px;
padding: 0px;
border-color:#336699; }
tr, td {      border:thin solid #336699;      }
.titre {
background-color: #336699;
color: #fff;
family-font: Verdana, Arial, Helvetica, sans-serif;
font-weight: bold;
text-align: center;      }
```

# Résultat à l'affichage

36  
71



Mixité PHP/HTML - Mozilla Firefox

Fichier Edition Affichage Aller à Marque-pages Outils ?

## Manipulation de tableau

Liste des enseignements

java	cours/coursJava.pdf
web	cours/coursWeb.pdf
agl	cours/coursAGL.pdf

\_\_\_ Fin Code PHP \_\_\_

Terminé

# Code source de la page web affichée

37  
71

```
<HTML>
<HEAD>
<LINK rel="stylesheet" type="text/css" href="tableau.css">
<TITLE>Mixité PHP/HTML</TITLE> </HEAD> <BODY>
<H1>Manipulation de tableau</H1>
affichage contenu de $tab dans un tableau <BR>
<table>
  <TR> <TD>java</TD> <TD>cours/coursJava.pdf</TD> </TR>
  <TR> <TD>web</TD> <TD>cours/coursWeb.pdf</TD> </TR>
  <TR> <TD>agl</TD> <TD>cours/coursAGL.pdf</TD> </TR>
</table>
<P>___ Fin Code PHP ___</P>
</BODY>
</HTML>
```

# Synthèse : séparation contenu/format

38  
71

- Prolongation du principe abordé tr. 27 avec en plus séparation au niveau des fichiers :
  - **1 fichier texte pour les données**
  - **1 fichier php pour l'affichage**
- Intérêt :
  - Évite d'avoir à modifier du code HTML pour ajouter des lignes et colonnes.
  - Modification limitée à celle un fichier texte.
  - **Profiter de PHP & CSS pour que la mise à jour ne concerne que la modification d'un fichier texte contenant toutes les données à afficher.**
- Conduit à produire des bibliothèques de fonctions assez générique pour être réutilisable... => comparer les codes des tr. 27 & 34
- Ex. de fichier texte:
  - 1<sup>ère</sup> ligne : titre des colonnes
  - Lignes suivantes : données

```
Code Cours|Intitulé|date version
C221|Programmation Orientée Objet|10/11/2006|coursJava.pdf
C121|Outils pour le web|10/10/2006|coursWeb.pdf
C221|Atelier de Génie Logiciel|10/09/2006|coursAGL.pdf
```

# Profiter de la mixité PHP/HTML

39  
71

```
<?php          // fonctions_table.php

function formate_titre($tab) {
    $ligne='<tr class="titre">';
    for( $i=0; $i<count($tab) ; $i++ )
        if( isset($tab[$i]) )
            $ligne.='<TD>' . $tab[$i] . '</TD>';
    $ligne.='</TR>';
    return $ligne;
}

function formate_ligne_cours($tab) {
    $ligne='<TR><TD>' . $tab[0] . '</TD>';
    $ligne .='<TD><a href="" . $tab[3] . "">' . $tab[1] . '</a></TD>';
    $ligne .='<TD>' . $tab[2] . '</TD>';
    return $ligne;
}

function formate_tableau($tab, $legende=""){
    if(!is_array($tab) ) return;

    $tableau_html='<table><caption>' . $legende . '</caption>';
    $tableau_html .=formate_titre( array_shift($tab) );
    for($i=0; $i<count($tab); $i++)
        $tableau_html .=formate_ligne_cours($tab[$i] );
    $tableau_html .='</table>';
    return $tableau_html;
}
?>
```

Vérifie que l'élément existe.  
isset renvoie faux si la variable n'est pas définie ie qu'il n'y a pas d' \$i<sup>ème</sup> élément

Extrait et supprime le premier élément du tableau. La fonction formate tableau s'attend à trouver le titre dans la 1<sup>ère</sup> ligne

Insérer chaque ligne du tableau dans une d'un tableau HTML

# Code php (suite)

40  
71

```
<HTML>
<HEAD>
  <LINK rel="stylesheet" type="text/css"
        href="tableau.css">
  <TITLE>Mixité PHP/HTML</TITLE>
</HEAD>
<BODY>
  <H1>Manipulation de tableau</H1>
  <?php
    require_once("fonctions_table.php");

    $tab=charge_fichier("BD.txt");
    echo formate_tableau2($tab);

  ?>
  <P>___ Fin Code PHP ___</P>
</BODY>
</HTML>
```

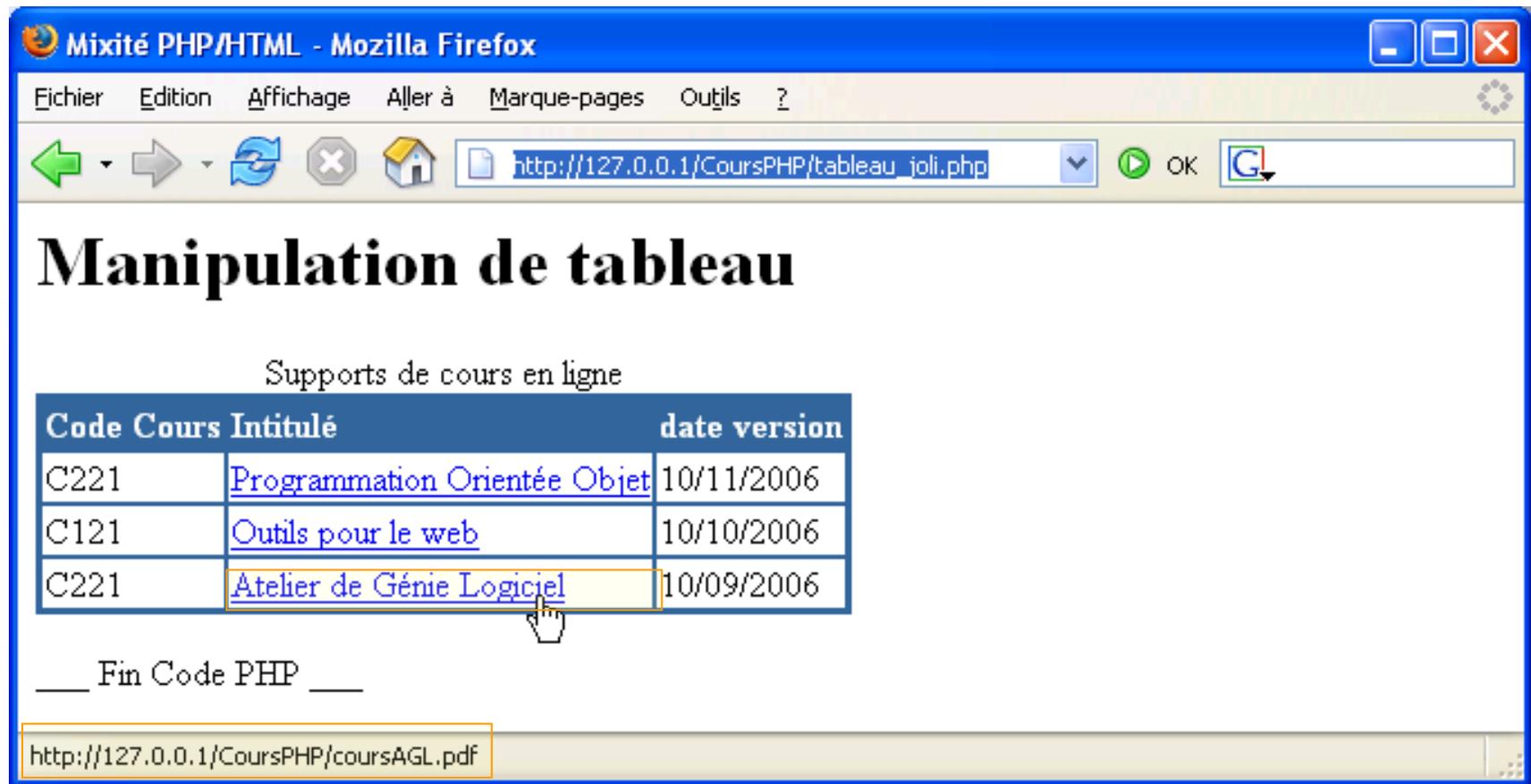
**fopen(...), fclose(), fgets(), strlen :**  
fonctions équivalentes à leur homonyme  
du langage C.

**explode :** transforme une chaîne de  
caractère en tableau de chaîne en  
utilisant le 1<sup>er</sup> paramètre comme  
délimitation de découpage (ici '|')

```
function charge_fichier($file) {
  $tab=array();
  $handle=fopen($file,"r");
  if($handle==False)
    return False;
  $str=""; $i=0;
  while ( $str=fgets($handle) ) {
    if (strlen($str)>0 )
      $tab[]=$explode('|',$str);
  }
  fclose($handle);
  return $tab;
}
```

# Résultat à l'affichage

41  
71



Mixité PHP/HTML - Mozilla Firefox

Fichier Edition Affichage Aller à Marque-pages Outils ?

http://127.0.0.1/CoursPHP/tableau\_joli.php

## Manipulation de tableau

Supports de cours en ligne

Code	Cours Intitulé	date version
C221	<a href="#">Programmation Orientée Objet</a>	10/11/2006
C121	<a href="#">Outils pour le web</a>	10/10/2006
C221	<a href="#">Atelier de Génie Logiciel</a>	10/09/2006

\_\_\_ Fin Code PHP \_\_\_

http://127.0.0.1/CoursPHP/coursAGL.pdf

---

# 3. Lien avec une base de donnée

---

- La séparation des rôles est claire :
    - HTML/CSS transmet et formate l'information à destination du client web
    - PHP : calcule, transforme l'information pour produire du code HTML
    - Le SGBD fournit les données nécessaires au code PHP pour répondre au besoin de l'utilisateur du client web (catalogue, modification base de donnée...).
    - Suivons la logique jusqu'au bout : il est alors possible d'administrer une BD depuis un code PHP !
      - phpMyAdmin
-

# phpMyAdmin

44  
71

127.0.0.1 >> localhost >> bdcours | phpMyAdmin 2.6.1 - Mozilla Firefox

Fichier Edition Affichage Aller à Marque-pages Outils ?

Serveur: localhost ▶ Base de données: bdcours

Structure SQL Exporter Rechercher Requête Opérations Supprimer

Table	Action	Enregistrements	Type	Interclassement	Taille	Perte
<input type="checkbox"/> cours		0	MyISAM	latin1_swedish_ci	1,0 Ko	-
<b>1 table(s)</b>	<b>Somme</b>	<b>0</b>	<b>--</b>	<b>latin1_swedish_ci</b>	<b>1,0 Ko</b>	<b>0 Octets</b>

Tout cocher / Tout décocher Pour la sélection :

Version imprimable Dictionnaire de données

Créer une nouvelle table sur la base bdcours:

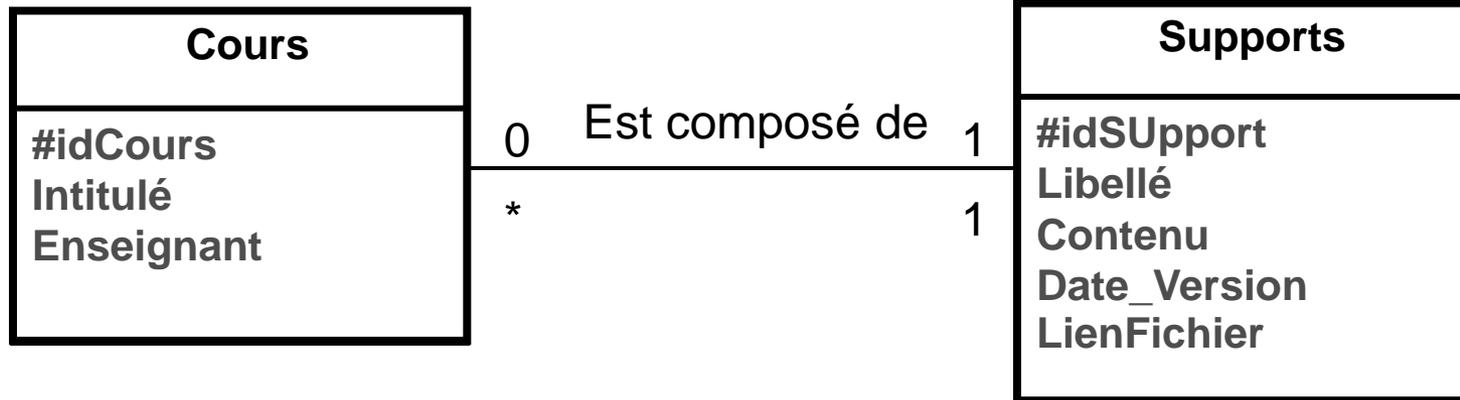
Nom:

Champs:  Exécuter

Terminé

# Définition des tables

45  
71



```
CREATE TABLE `cours` (  
  `pk_cours` tinyint(4) NOT NULL auto_increment,  
  `Intitulé` text NOT NULL,  
  `Enseignant` text NOT NULL,  
  PRIMARY KEY (`pk_cours`)  
)
```

```
CREATE TABLE `supports` (  
  `pk_support` tinyint(4) NOT NULL auto_increment,  
  `fk_cours` tinyint(4) NOT NULL,  
  `libellé` text NOT NULL,  
  `contenu` text NOT NULL,  
  `date_version` date NOT NULL  
  `lien` text NOT NULL,  
  PRIMARY KEY (`pk_support`),  
  FOREIGN KEY (`fk_cours`)  
    REFERENCES `cours`(`pk_cours`)  
)
```

# Remplissage de la table

46  
71

```
INSERT INTO `cours`
```

```
( `pk_cours` , `Intitulé` , `Enseignant` )
```

```
VALUES
```

```
(", 'Support de cours Java', 'P. Bellemare'),
```

```
(", 'Cours Web', 'P. B, V. P, E. T'),
```

```
(", 'Mémoire d'entreprise et Capitalisation des connaissances', 'E. Tranvouez')
```

```
);
```

```
INSERT INTO `supports`
```

```
VALUES
```

```
(1, 1, 'Support général', 'Cours de conception et de programmation orienté objet basé sur le langage Java', '2006-10-10', 'pdf/coursJava.pdf'),
```

```
(2, 2, 'Session CSS', 'Conception de feuille de style CSS permettant de séparer le fond de la forme : contenu / présentation', '2006-10-18', 'pdf/coursCSS.pdf'),
```

```
(3, 3, 'Cours général', 'Problématique de la gestion des connaissances en entreprise : enjeux, concepts, solutions méthodologiques et technologiques', '2006-11-05', 'pdf/coursME-KM.pdf'),
```

```
(4, 2, 'Session PHP', 'Conception de page web dynamique avec PHP.\r\nSyntaxe du langage et exemples pratiques.', '2006-11-10', 'pdf/coursPhp.pdf')
```

```
;
```

# PhPMyAdmin

47  
71

The screenshot shows the phpMyAdmin 2.6.1 interface in a Mozilla Firefox browser window. The address bar shows the URL `http://127.0.0.1/mysql/`. The page title is `phpMyAdmin 2.6.1 - Mozilla Firefox`. The interface includes a menu bar (Fichier, Edition, Affichage, Aller à, Marque-pages, Outils), a navigation bar with icons for Structure, Afficher, SQL, Rechercher, Insérer, Exporter, Opérations, and Vider. The main content area displays the following information:

Serveur: localhost | Base de données: bdcours | Table: cours

Affichage des enregistrements 0 - 2 (3 total, traitement: 0.0003 sec.)

requête SQL:  
`SELECT *`  
`FROM `cours``  
`LIMIT 0, 30`

[Modifier] [Expliquer SQL] [Créer source PHP] [Actualiser]

Afficher : 30 ligne(s) à partir de l'enregistrement n° 0  
en mode horizontal et répéter les en-têtes à chaque groupe de 100  
Trier sur l'index: aucune [Exécuter]

	pk_cours	Intitulé	Enseignant
<input type="checkbox"/>	1	Support de cours Java	P. Bellemare
<input type="checkbox"/>	2	Cours Web	P. Bellemare, V. Pujalte, E. Tranvouez
<input type="checkbox"/>	3	Mémoire d'entreprise et Capitalisation des connais...	E. Tranvouez

Tout cocher / Tout décocher Pour la sélection :

Terminé

- Présenter sous forme de tableau la liste des supports de cours :
    - => Gestion la BD
      - Définir les tables
      - Peupler les tables
      - Définir la requête permettant d'extraire les données désirées
    - => Développement php
      - Gérer la connexion SGBD
      - Ouvrir la BD
      - Exécuter une requête et récupération des résultats
      - Exploiter les résultats (à la manière de celle employée avec le fichier texte)
-

- Connexion:
    - **mysql\_connect**(\$ip\_serveur, \$nom, \$mot2passe)
      - Renvoie une valeur permettant de gérer la connexion avec le serveur SGBD MySQL
  - Sélection de la Base de Donnée (BD)
    - **mysql\_select\_db**(\$db)
      - Se positionne sur la base de donnée indiquée
  - Exécution de la requête
    - **mysql\_query**(\$str)
      - Ne renvoie pas le résultat de la requête mais le résultat de son exécution (vaut **false** si la requête n'a pas pu être traitée)
-

- Récupération des résultats (*liste non exhaustive!*)
  - Par lignes
    - `$tab = mysql_fetch_row($res)`
      - `$res != false => succès`
      - `$tab` est un tableau dans lequel les clés sont les noms des colonnes
      - Ex: `$tab['nom_auteur'] =>` renvoie la valeur pour la ligne courante de la colonne 'nom\_auteur'.
  - Par objets
    - `$obj = mysql_fetch_object($res)`
      - Idem pour `$res`
      - `$obj` est un objet (sorte de « structures évoluées »)
      - Il se manipule comme un pointeur sur une structure en C
      - `$obj -> nom_champ` : renvoie la valeur pour la ligne courante de la colonne 'nom\_champ'

# Exemple : code php de connexion & transformation HTML

51  
71

<?php

```
define('SERVEUR','localhost');   define('USER','root');
define('M2P','');                 define('BD','bdcours');
```

```
function requete_tableau($req) {
    $connexion=mysql_connect(SERVEUR,USER,M2P);
    if (!mysql_select_db(BD) ) { echo mysql_error() ; return false;
```

```
    $res = mysql_query($req);
    $stab=array();                $i=0;
    while ( $temp=mysql_fetch_array($res,MYSQL_ASSOC) )
        $stab[]=$temp;
    return $stab;
```

```
}
```

```
function formate_tableau_sql( $stab, $titre=""){
    if(!is_array($stab) && count($stab)<1 )    return;
    $keys=extrait_cles_asso(array_keys($stab[0]));
    $tableau_html="<table><caption>$titre</caption>";
    $tableau_html.=formate_titre_sql($keys);
    for($i=0; $i<count($stab) ; $i++)
        $tableau_html.= formate_ligne_sql($stab[$i],$keys);
    $tableau_html.="</table>";
    return $tableau_html;
```

```
}
```

**constantes globales** : identique à un #define SERVEUR localhost. Pour faciliter leur mise à jour ces variables peuvent être stockées à part. Déployer le code sur un serveur ne nécessitera la mise à jour que de ce fichier.

**A chaque cycle de la boucle** : récupération d'une ligne de résultat de la requête sous la forme d'un tableau associatif ayant pour clés les noms des champs des tables (ex. \$stab['pk\_cours']  
Puis ajout au tableau

Une fois les données extraites de la table, on retrouve un principe identique de boucle produisant des tableaux HTML...

# Exemple : code php de connexion & transformation HTML

52  
71

\$keys contient les noms des champs utilisés comme titre de tableau html.

```
// ... suite
```

```
function formate_titre_sql($keys) {  
  
    $ligne='<tr class="titre">';  
    for($i=0;$i<count($keys); $i++) {  
        $ligne.="<TD>$keys[$i]</TD>";  
    }  
    $ligne.='</TR>';  
    return $ligne;  
}  
  
function formate_ligne_sql( $tab, $keys) {  
    $ligne='<TR>';  
    for($i=0; $i<count($tab) ; $i++){  
        $ligne.="<TD>" . $tab[ $keys[$i] ] .  
"</TD>";  
    }  
    $ligne.='</TR>';  
    return $ligne;  
}
```

\$keys[ \$i ] contient le nom du i<sup>ème</sup> champ  
\$tab[ \$key[ \$i ] ] contient la valeur de ce champ

```
// ... suite
```

*/\* comme le résultat de la requête SQL est renvoyé dans un tableau associatif, les index alphabétiques du tableau contiennent les noms des colonnes de la table... \*/*

```
function extrait_cles_asso($tab) {  
    $asso = array();  
    for($i=0; $i < count($tab); $i++){  
        if( isset($tab[$i]) && is_string($tab[$i]) )  
            $asso[]=$tab[$i];  
    }  
    return $asso;  
}  
  
?>
```

Si \$tab[ \$i ] est défini et contient un string, l'ajouter au tableau.

# Exemple : code php de la page principale

53  
71

```
<HTML>
<HEAD>
  <LINK rel="stylesheet" type="text/css" href="tableau.css">
  <TITLE>Mixité MySQL/PHP/HTML</TITLE>
</HEAD>
<BODY>
  <H1>Affichage contenu table (catalogue)</H1>

  <?php
    require_once("connectionSQL.php");
    $tab=requete_tableau("select * from cours");
    echo formate_tableau_sql($tab);
  ?>
  <P>__ Fin Code PHP __</P>
</BODY>
</HTML>
```

**Requête SQL** : liste le contenu de la table Cours.

# Résultat à l'affichage

54  
71

- L'ajout d'élément peut se faire via phpMyAdmin ie un client mySQL permettant de faire des ajouts dans la table Cours.



The screenshot shows a Mozilla Firefox browser window with the title "Mixité MySQL/PHP/HTML - Mozilla Firefox". The address bar shows the URL "http://127.0.0.1/CoursPHP/tableau\_SQL.php". The main content area displays the heading "Affichage contenu table (catalogue)" and the subtitle "Supports de cours en ligne". Below this is a table with three columns: "pk\_cours", "Intitulé", and "Enseignant". The table contains three rows of data. At the bottom of the page, there is a line of code "\_\_\_ Fin Code PHP \_\_\_" and a status bar that says "Terminé".

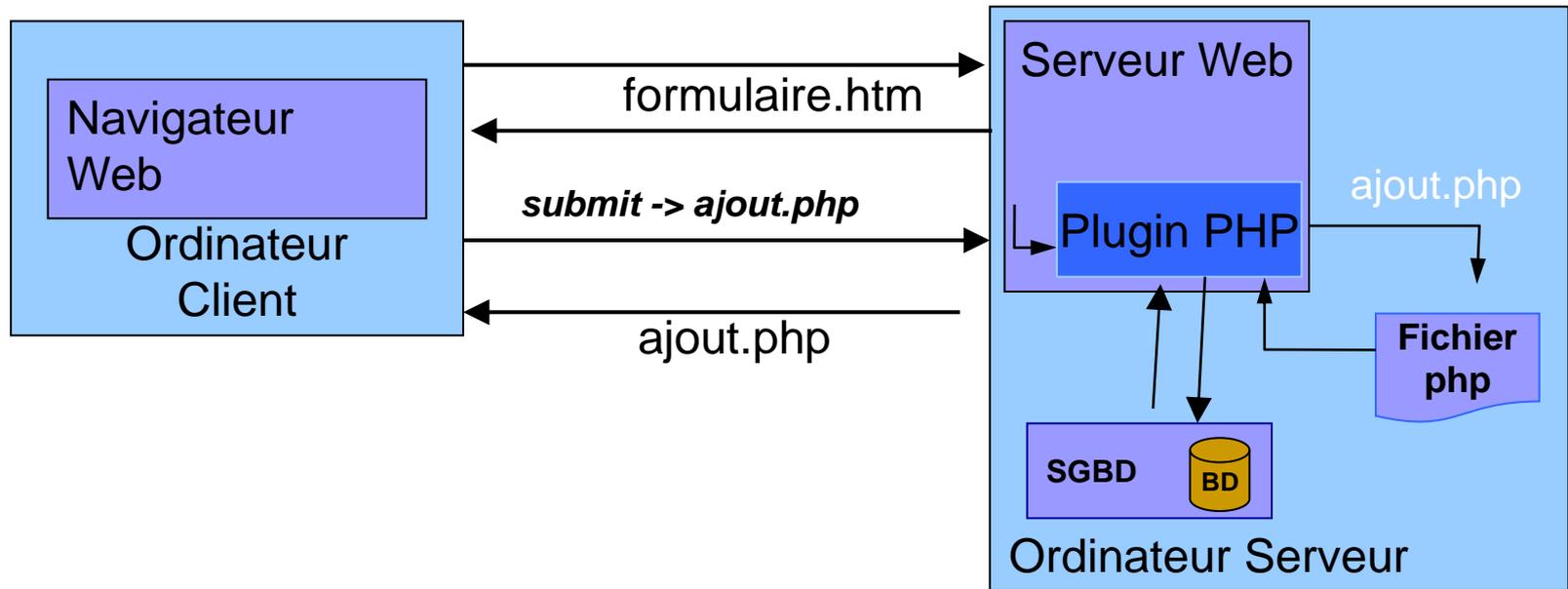
pk_cours	Intitulé	Enseignant
1	Support de cours Java	P. Bellemare
2	Cours Web	P. Bellemare, V. Pujalte, E. Tranvouez
3	Mémoire d'entreprise et Capitalisation des connaissances	E. Tranvouez

\_\_\_ Fin Code PHP \_\_\_

Terminé

# Formulaire php : principe

55  
71



- Le champs FORM de *formulaire.htm* indique que l'action doit être réalisée par *ajout.php*
- Il est aisément imaginable comment le formulaire pourrait lui-même être un script php...

# Rappels HTML sur les formulaires

56  
71

- Permet de transmettre une requête HTML d'un client à un serveur à l'aide des balises :
    - `<FORM ></FORM>` : balise encadrant le formulaire
    - `<INPUT TYPE=...>` TYPE pouvant valoir :
      - TEXT : saisie d'une zone de texte
      - RADIO : sélection d'un item dans une liste
      - CHECKBOX : case à cocher. Permet de sélectionner un ou plusieurs items.
      - BUTTON: bouton. Le code étant dans la page html côté client, y sera associé essentiellement du code javascript.
      - RESET: bouton mettant à zéro tous les champs.
      - SUBMIT: bouton validant l'envoi du formulaire au serveur web.
    - `<TEXTAREA>` : saisie d'un texte long (multi-lignes)
    - `<SELECT><OPTION>...` : menu déroulant
    - ...
-

# Exemple : code HTML d'un formulaire d'ajout

57  
71

```
<html>
<head>
  <title>Formulaire HTML</title>
</head>
<body>
  <h2>Premier Formulaire</h2>
  <form name="premierFormulaire" method="post" action="">
    <p>Intitulé du cours<input name="intitulé" type="text" value="" ></p>
    <p>Enseignant(s) responsable(s)<input name="ens" type="text" value="" ></p>
    <input name="ajoutBtn" type="submit" value="Ajouter">
    <input name="initBtn" type="reset" value="ReInitialiser">
  </form>
</html>
```

## Déclaration du formulaire :

-**name**: nom identifiant le formulaire

-**Method**: Sera envoyé au serveur l'intégralité de son contenu ie des couples de valeurs {nom\_objet\_form, valeur} par ex. {'ens', 'toto'}. La methode indique la manière dont les infos doivent être envoyées:

-**post**: message non visible par l'utilisateur

-**get** : ajouté à la suite de l'url (ex. requête google)

-**action** : nom de la commande à activer par le serveur web pour traiter ce formulaire: ex. un programme cgi, un script php...

# Exemple : code HTML d'un formulaire d'ajout

58  
71



# Exemple : code HTML du script d'ajout

59  
71

```
<HTML>
<HEAD>
<TITLE>Mixité MySQL/PHP/HTML</TITLE>
</HEAD>
<BODY>
<H1>Affichage contenu table (catalogue)</H1>
<?php
require_once("connectionSQL.php");
print_r($_POST);          echo '<pre><BR>';
$intitule= $_POST['intitule'];
$ens = $_POST['enseignant'];
$req = "INSERT INTO Cours VALUE ('', '$intitule . '", '$ens . ')";
echo $req;
$connexion=mysql_connect(SERVEUR,COMPTE,MdP);
if (!mysql_select_db(BD) )      echo mysql_error() ;
$res = mysql_query($req);
if ($res)
    echo "<p>Element ajouté<p>";
else
    echo "<p>Erreur ajout<p>";
?>
<P>__ Fin Code PHP __</P>
</BODY>
</HTML>
```

Récupération des valeurs du formulaire  
ie les valeurs qu'avaient les champs du  
formulaire lors de son envoi

Construction de la requête d'ajout  
(insert into) avec rien dans le 1<sup>er</sup> champ  
pk\_cours (") afin que la clé soit ajoutée  
automatiquement par MySQL

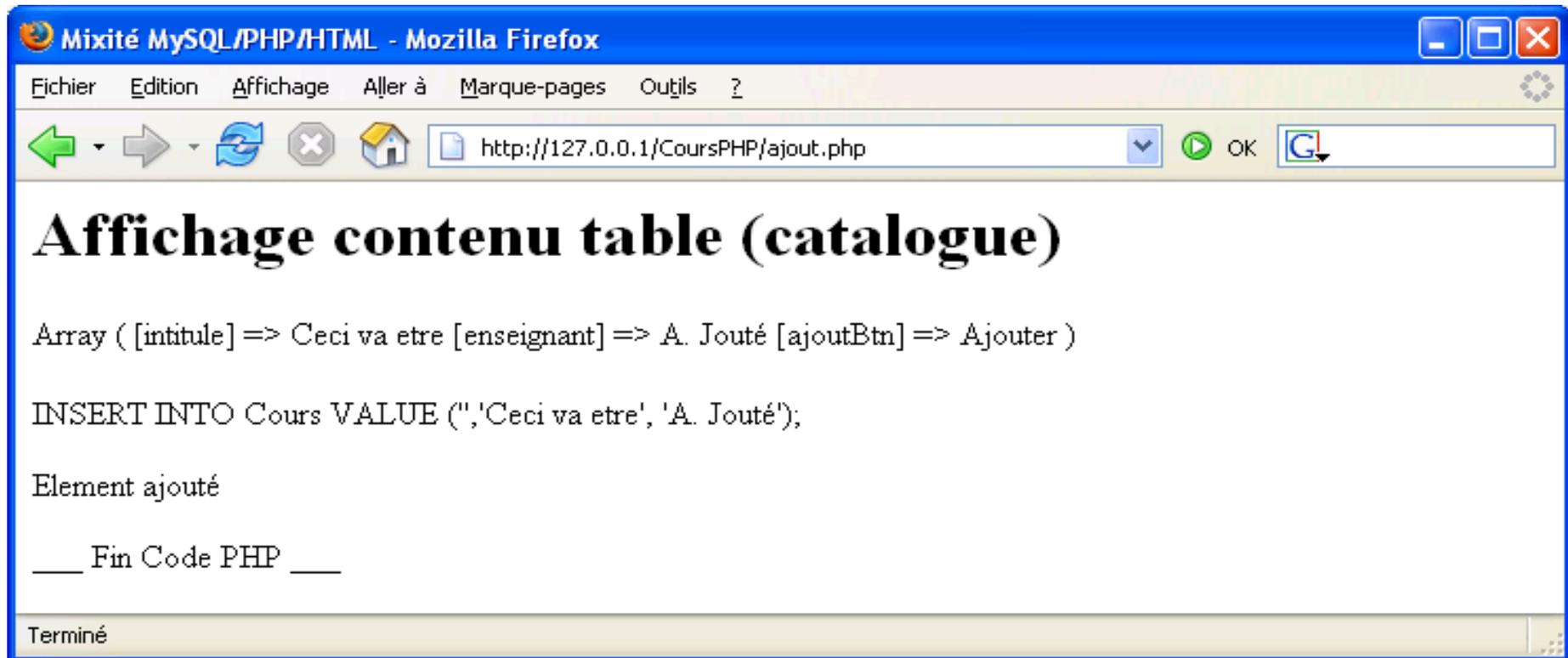
Connexion à la base et vérification

Exécution de la requête

Affichage destiné à l'utilisateur pour  
vérifier le succès ou non de l'ajout

# Exemple : résultat

60  
71



Mixité MySQL/PHP/HTML - Mozilla Firefox

Fichier Edition Affichage Aller à Marque-pages Outils ?

http://127.0.0.1/CoursPHP/ajout.php OK

## Affichage contenu table (catalogue)

Array ( [intitule] => Ceci va etre [enseignant] => A. Jouté [ajoutBtn] => Ajouter )

```
INSERT INTO Cours VALUE ('','Ceci va etre', 'A. Jouté');
```

Element ajouté

\_\_\_ Fin Code PHP \_\_\_

Terminé

# Aparté : Attaque par injection SQL

61  
71

- Principe : insérer du code SQL là où est attendue une valeur.
  - Ex : Zone de saisie d'un mot de passe
  - Danger => permet de redéfinir la requête et contourner les vérifications.

```
<?php
```

```
$query = "SELECT * FROM users WHERE user="" . $_POST['username'] . "" ";
```

```
$query .= "AND password="" . $_POST['password'] . "" ";
```

```
mysql_query($query);    ?>
```

Récupération des valeurs des 2 champs de formulaire 'user\_name' et 'password'

- Problème on ne vérifie pas le contenu de la chaîne renvoyée par `$_POST['password']`. Par exemple si l'utilisateur a saisi toto dans le champ username et `'OR '='` dans le champ password la requête envoyé au serveur devient :
  - `SELECT * FROM users WHERE user='toto' AND password='' OR '=';`
  - La condition "=" étant toujours vraie la condition du WHERE sera vraie quelque soit le nom d'utilisateur utilisé
  - => contournement de l'authentification...

# Aparté : Attaque par injection SQL

62  
71

- **Solution** : vérifier le contenu des champs avant de les envoyer au serveur...
  - Soit manuellement
    - ❑ Avec une fonction spécifique au SGBD qui sait quels caractères d'échappement utiliser pour ce SGBD (cf. ex. dans code source encadré. La fonction **sprintf** est similaire à son homonyme en C)
    - ❑ Avec des fonctions d'échappement *addslashes()* et sa fonction inverse *stripslashes()*
  - Soit automatiquement avec l'activation des apostrophes magiques (*magic quotes*) permettant automatiquement d'échapper les chaînes soumises via un formulaire

```
$query = sprintf("SELECT * FROM users WHERE user='%s' AND password='%s'",  
mysql_real_escape_string($user),  
mysql_real_escape_string($password));
```

# Exemple de synthèse

- Utilisation d'un script pour afficher les cours existants (= > requête SQL) ...
- Sélection du cours désiré
- Soumission du formulaire pour les supports de cours correspondants soient affichés...
- Donc :
  - 1 fichier formulaire php
  - 1 fichier résultat php
- Dans l'exemple suivant:
  - chaque résultat constitue 1 ligne du tableau permettant un affichage régulier des cours.
  - Donc chaque résultat est compris entre `<TR><TD>` et `</TD></TR>`.

# Formulaire : sélection du cours

64  
71

```
<html> .... <body>
<?php
require_once("connectionSQL.php");
$stab=requete_tableau("SELECT * FROM cours");
echo "<form action=\"affiche_supports.php\" METHOD=\"POST\" NAME=\"Affiche_support\" >";
echo "<table>\n";
$keys=array_keys($stab[0]);
for($i=0; $i < count ($stab) ; $i++){
    $value=str_replace("'", "\'", $stab[$i]["intitule"] . " " . $stab[$i]["enseignant"] );
    $strong_value= "<strong>" . $stab[$i]["intitule"] . "</strong> enseign&eacute; par <strong>";
    $strong_value .= $stab[$i]["enseignant"] . "</strong>";
    echo "<TR><TD><br>\n";
    echo "<input type=\"radio\" name=\"pk_cours\" onClick=\"alert('Ligne s&eacute;lectionn&eacute;e : ‘";
    echo "' . $value . " clef: " . $stab[$i]["pk_cours"] . ")\" value=\"" . $stab[$i]["pk_cours"] . "\"";
    echo ">" . $strong_value . "\n</TD></TR>\n";
}
echo "</table>";
echo "<input type=\"submit\" name=\"Valider\" title=\"Valider\">";
echo "</form>";
?>                ...                </body></html>
```

Réutilisation de la fonction définie précédemment. Elle retourne sous la forme d'un tableau le résultat de la requête

Remplace le ' par \' dans le noms du cours pour que le code javascript lié à l'événement onClick fonctionne

= 1  
ligne  
du  
tableau  
le =  
1 cours

# Aperçu du Formulaire

65  
71

The screenshot shows a Mozilla Firefox browser window with the title "Mixité MySQL/PHP/HTML - Mozilla Firefox". The address bar displays "http://127.0.0.1/CoursPHP/formulaire.php". The main content area is titled "Affichage des supports d'un cours" and contains a list of radio buttons for course selection:

- Support de cours Java enseigné par P. Bellemare
- Cours Web enseigné par P. Bellemare, V. Pujalte, E. Tranvouez
- Mémoire d'entreprise et Capitalisation des connaissances enseigné par E. Tranvouez
- Cours passionnant enseigné par Quelqu'un d'autre
- Ceci va etre enseigné par A. Jouté

Below the list is an "Envoyer" button and a text field containing "\_\_\_ Fin Code PHP \_\_\_". A status bar at the bottom left shows "Terminé".

A dialog box titled "http://127.0.0.1" is overlaid on the bottom right, displaying a warning icon and the message: "Ligne sélectionnée : Cours Web P. Bellemare, V. Pujalte, E. Tranvouez clef: 2". An "OK" button is visible at the bottom of the dialog box.

# Code HTML du Formulaire

66  
71

```
<HTML>
<HEAD>          <LINK rel="stylesheet" type="text/css" href="tableau.css">  </HEAD>
<BODY>
  <H1>Affichage des supports d'un cours</H1>
  <FORM ACTION="affiche_supports.php" METHOD="POST" NAME="Affiche_support" >
  <TABLE>
```

Correspond à 1 cours. L'attribue 'value' contient la valeur qui sera envoyée par le formulaire au serveur

Test de combinaison javascript et php. A chaque sélection une fenêtre popup s'affichera.

```
<TR><TD><BR><INPUT type="radio" name="pk_cours"
onClick="alert('Ligne sélectionnée : Support de cours Java P. Bellemare
clef: 1')" value="1"> <STRONG>Support de cours Java</STRONG>
enseigne; par <STRONG>P. Bellemare</STRONG></TD></TR>
```

```
<TR><TD><BR><INPUT type="radio" name="pk_cours" onClick="alert('Ligne
sélectionnée : Cours Web P. Bellemare, V. Pujalte, E. Tranvouez clef: 2')"
value="2"><STRONG>Cours Web</STRONG> enseigne; par <STRONG>P.
Bellemare, V. Pujalte, E. Tranvouez</STRONG></TD></TR>
```

```
<!-- ...etc -->
```

```
</TABLE>
```

```
<INPUT type="submit" onClick="alert('test')" name="Valider" title="Valider">
```

```
</FORM>
```

```
</BODY> </HTML>
```

# Fichier résultat du formulaire 1/2

67  
71

```
<HTML>
<HEAD>
  <LINK rel="stylesheet" type="text/css" href="tableau.css">
  <TITLE>Mixité MySQL/PHP/HTML</TITLE>
</HEAD>
<BODY>
  <?php
    require_once("connectionSQL.php");

    function nom_cours($pk_cours) {
      if ( ! connect_SQL() )
        return false;

      $req = "SELECT intitule FROM cours WHERE pk_cours=" . $pk_cours . ";";
      $res = mysql_query($req);
      if (!$res)
        $nom="COURS INCONNU";
      else{
        $tab=mysql_fetch_row($res);
        $nom=$tab[0];
      }
      return $nom;
    }
  }
}
```

Fonction qui retourne l'intitulé du cours correspondant à l'ID passé en paramètre

Teste que le cours a été trouvé...

mysql\_fetch\_row renvoie le résultat courant de la requête sous la forme d'un tableau avec index numérique...  
... comme 1 seul résultat est attendu il suffit de l'appeler 1 fois.

... la requête ne portait qu'un champ (intitule) aussi peut on directement prendre le 1<sup>er</sup> élément du tableau

# Fichier résultat du formulaire 2/2

68  
71

```
$pk_cours = $_POST['pk_cours'];  
$intitule_cours=nom_cours ( $_POST['pk_cours'] );
```

Valeur contenue dans le champ de formulaire pk\_cours

```
$req = 'SELECT * FROM supports WHERE (supports.fk_cours=' . $pk_cours . ')';  
$stab=requete_tableau($req);
```

Si n'est pas un tableau, c'est que le résultat est vide...

```
if( ! is_array($stab || count($stab) < 1 ) ) {  
    echo "<H1>ERREUR ... </H1>";  
    echo "<span style=\"color:red; font-weight:bold;\">Aucun support disponible";  
    echo "pour le cours -" . $intitule_cours . "-</span>";  
}  
else {  
    echo "<H1> Liste des supports disponible pour le cours " . $intitule_cours . "</H1>";  
    echo formate_tableau_sql($stab);  
}  
?>
```

Réutilisation de la fonction définie précédemment. Elle retourne sous la forme d'un tableau le résultat de la requête

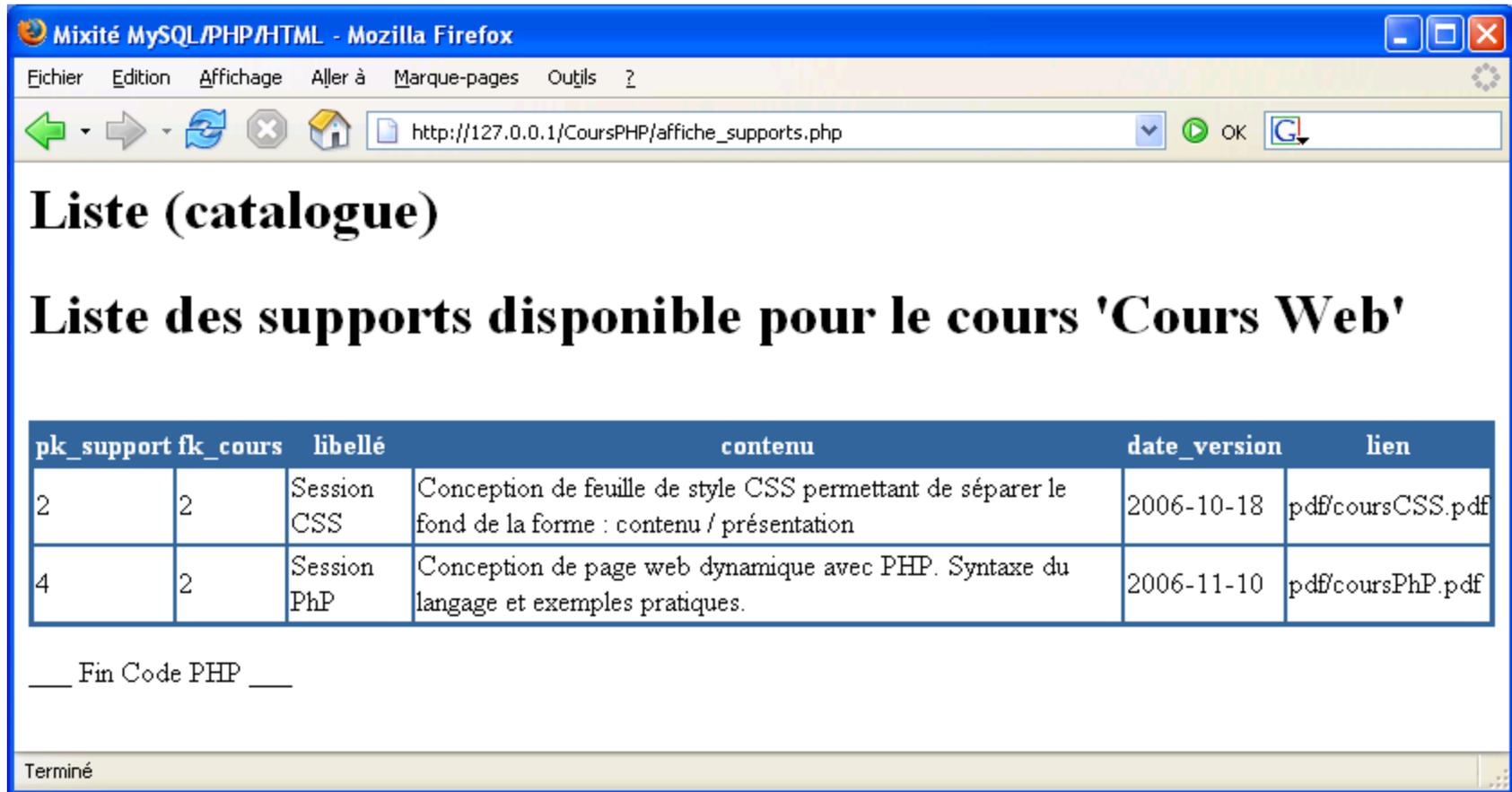
```
<P>___ Fin Code PHP ___</P>
```

```
</BODY>
```

```
</HTML>
```

# Aperçu du Résultat : succès !

69  
71



Mixité MySQL/PHP/HTML - Mozilla Firefox

Eichier Edition Affichage Aller à Marque-pages Outils ?

http://127.0.0.1/CoursPHP/affiche\_sUPPORTS.php

## Liste (catalogue)

### Liste des supports disponible pour le cours 'Cours Web'

pk_support	fk_cours	libellé	contenu	date_version	lien
2	2	Session CSS	Conception de feuille de style CSS permettant de séparer le fond de la forme : contenu / présentation	2006-10-18	pdf/coursCSS.pdf
4	2	Session PhP	Conception de page web dynamique avec PHP. Syntaxe du langage et exemples pratiques.	2006-11-10	pdf/coursPhP.pdf

\_\_\_ Fin Code PHP \_\_\_

Terminé

# Code HTML du Résultat (succès)

70  
71

```
<HTML>
```

```
<HEAD> <LINK rel="stylesheet" type="text/css" href="tableau.css">
```

```
<TITLE>Mixité MySQL/PHP/HTML</TITLE>
```

```
</HEAD> <BODY>
```

```
<H1>Liste (catalogue)</H1>
```

```
<H1> Liste des supports disponible pour le cours 'Cours Web'</H1><BR>
```

```
<TABLE><caption></caption>
```

```
<TR class="titre">
```

```
<TD>pk_support</TD><TD>fk_cours</TD><TD>libellé</TD>
```

```
<TD>contenu</TD><TD>date_version</TD><TD>lien</TD>
```

```
</TR>
```

```
<TR>
```

```
<TD>2</TD><TD>2</TD><TD>Session CSS</TD>
```

```
<TD>Conception de feuille de style CSS permettant de paramétrer le fond de la forme :  
contenu / présentation</TD><TD>2006-10-18</TD><TD>pdf/coursCSS.pdf</TD>
```

```
</TR>
```

```
<TR><TD>4</TD><TD>2</TD><TD>Session PHP</TD>
```

```
<TD>Conception de page web dynamique avec PHP. Syntaxe du langage et exemples  
pratiques.</TD>
```

```
<TD>2006-11-10</TD><TD>pdf/coursPHP.pdf</TD></TR>
```

```
</TABLE>
```

```
<P>___ Fin Code PHP ___</P>
```

```
</BODY></HTML>
```

# Aperçu du Résultat : échec !

71  
71

