

# **Langage PHP et BD (MySQL-Access-Oracle)**

**Dr Konan Marcellin BROU**

**konanmarcellin@yahoo.fr**

**2015-2016**

# Sommaire

---

- **Introduction**
- **Serveur Web**
- **Concepts de base de PHP**
- **Les tableaux**
- **Les Structures de contrôle**
- **Les Fonctions**
- **Les Fichiers**
- **PHP et MySQL**
- **PHP et Access**
- **PHP et Oracle**
- **Bibliographie**

# I. Introduction

---

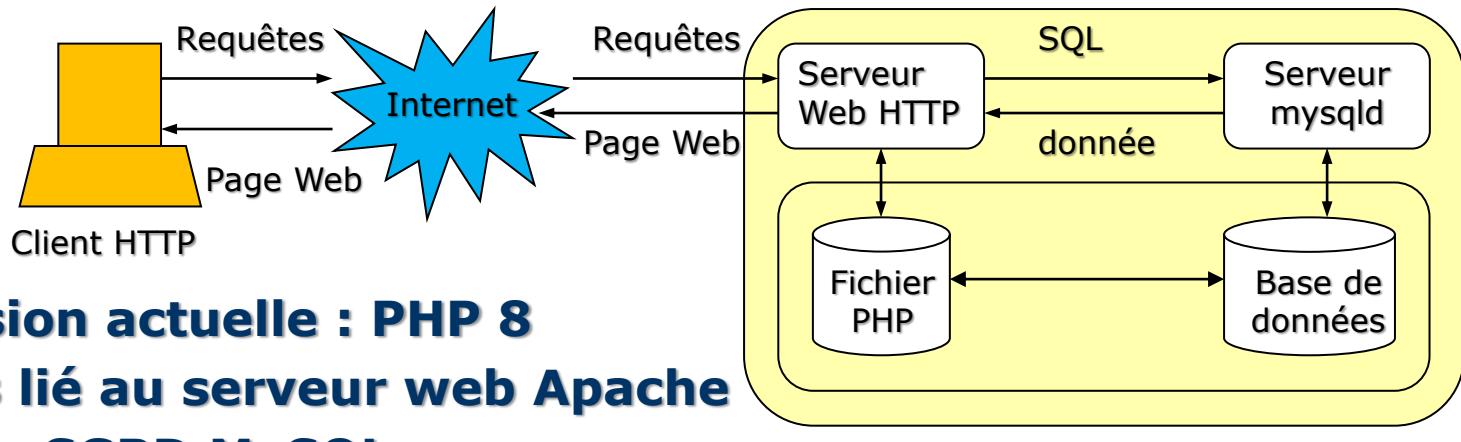
## □ 1.1. Objectifs

- **Apprendre les éléments de base de ce langage ;**
- **Savoir créer des sites web dynamiques ;**
- **Savoir comment rendre dynamique une page Web en y insérant des données issues d'une base de donnée :**
  - MySQL
  - Access
  - Oracle

# I. Introduction

## □ 1.2. Présentation

- **PHP (Hypertext PreProcessor)**
  - Anciennement **Personal Home Pages**
  - **HyperText** : s'oppose à texte linéaire comme un livre.
- **Langage de script s'exécutant sur un serveur Web.**
- **Permet de créer des sites Web dynamiques.**
- **Permet d'accéder aux données d'une BD pour les publier sur le Web.**



- **Version actuelle : PHP 8**
- **Très lié au serveur web Apache et au SGBD MySQL**

# I. Introduction

## □ 1.3. Exemple de script PHP

### ■ Code PHP délimité par les balises :

□ <?php : balise d'ouverture, et ?> : balise de fermeture

ex1Php.php sur le serveur

```
<HTML>
<HEAD>
  <title>premiers pas en PHP</title>
</HEAD>
<BODY bgcolor="#FFFFFF">
  <H1>Essai de script PHP</H1>
  <?php
    echo "Bonjour, il est ";
    echo date ("H:i:s");
  ?>
</BODY>
</HTML>
```

page envoyée au navigateur

```
<HTML>
<HEAD><title>premiers pas en PHP </title>
</HEAD>
<BODY bgcolor="#FFFFFF">
  <H1>Essai de script PHP</H1>
  Bonjour, il est 08:50:27
</BODY>
</HTML>
```



□ tout ce qui était entre les balises  
<?php et ?> a été interprété  
et transformé en code HTML.

# II. Technologies côté serveur

---

## □ 2.1. Présentation

### ■ Technologies côté client

- **Technologies de développement d'application ou de sites Web.**
- **Elles sont utilisées pour concevoir l'interface utilisateur et prendre en charge les différents événements déclenchés par les utilisateurs.**
- **Les programmes s'exécutent sur le client :**
  - Javascript, VBScript, Applet...

### ■ Trois sortes de clients :

#### □ **Client léger :**

- Navigateur Web utilisant les technologies Internet pour proposer une interface graphique à l'utilisateur.
- N'est responsables que de la partie interface utilisateur.
- Tous les traitements de l'application sont déportés sur un serveur spécifique : serveur Web, ou serveur d'application.
- Technologies utilisés : HTML, JavaScript, CSS, Applets, ActiveX.

## II. Technologies côté serveur

---

### ▣ Client lourd

- Propose une interface graphique fenêtrée telle qu'une application de traitement de texte par exemple.
- Intègre également une partie non négligeable de la logique de traitement.
- Les technologies utilisées : bibliothèques de composants graphiques AWT et Swing en Java, MFC en langage C++ sous Windows, les bibliothèques d'accès aux BD JDBC pour Java, ADO et ODBC pour Microsoft.

### ▣ Client riche

- Compromis entre client lourd et client léger.
- L'interface graphique a une ergonomie comme les clients lourds.
- Majeur partie des traitements effectués sur un serveur d'application, comme pour les clients légers.
- Possibilité d'utiliser l'interface client en mode connecté ou déconnecté du serveur d'application.
- Technologies utilisée : plateforme Microsoft .NET, environnement Eclipse RCP (Rich Client Platform) utilisant les technologies Java.

## **II. Technologies côté serveur**

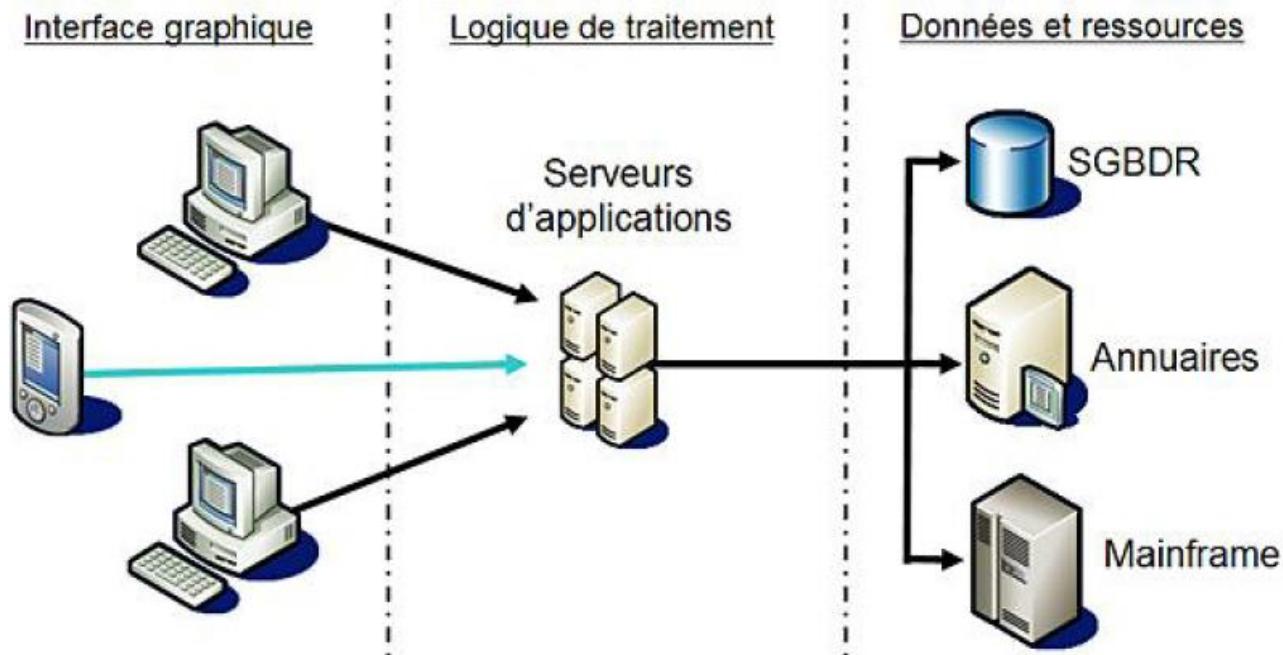
---

- **Technologies côté serveur**

- **Les programmes s'exécute sur le serveur :**
  - **CGI, ASP, PHP, JSP, Servlet...**

# II. Technologies côté serveur

## ■ Architecture 3 tiers



- Séparation de l'interface graphique utilisateur, de la logique de traitement de l'application, et des serveurs hébergeant les données et ressources utilisées par ces applications.

## II. Technologies côté serveur

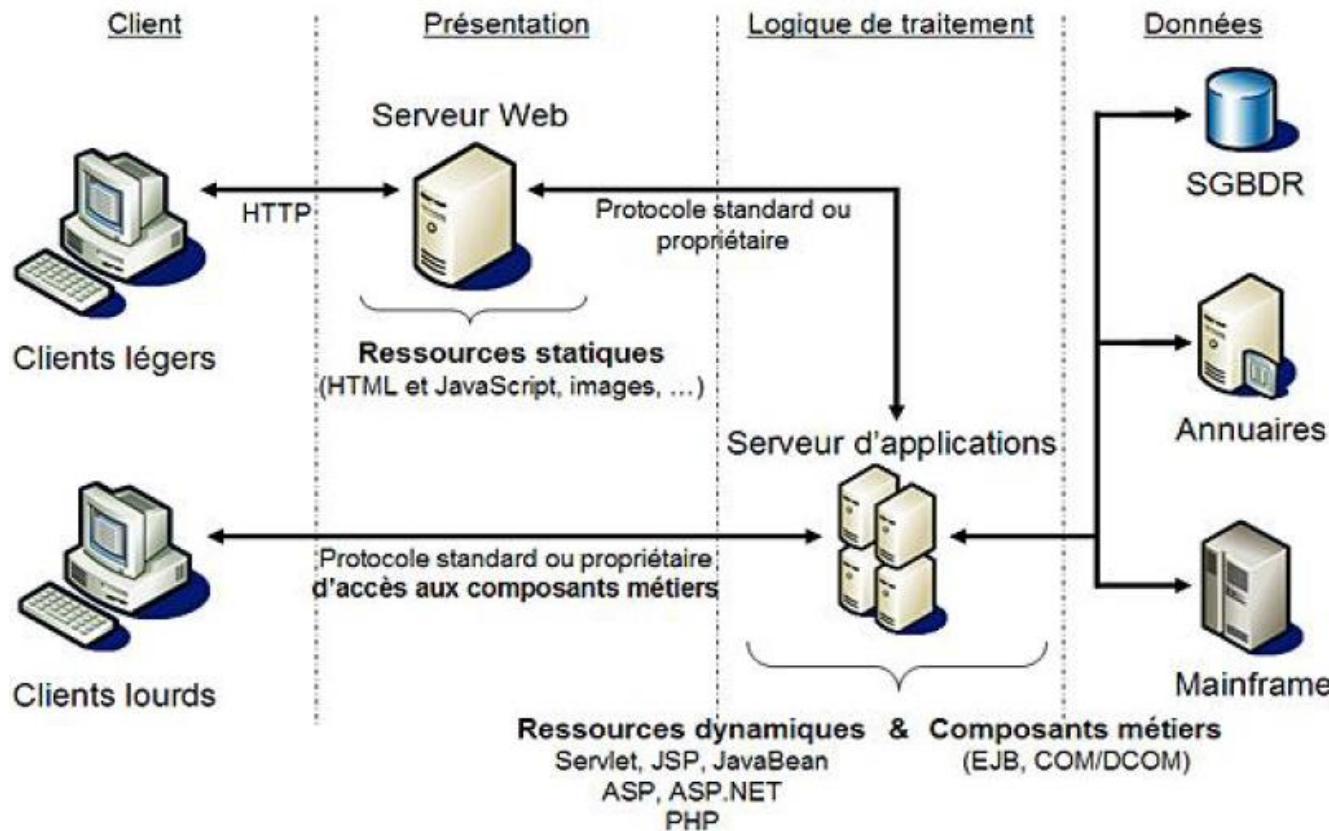
---

### □ Avantages :

- Les interfaces graphiques fonctionnant sur le poste client peuvent être allégées ;
- Le gros des traitements est réalisé sur un serveur d'application, et non plus sur le poste client ;
- La mise à jour d'un composant de traitement se fait sur le serveur et n'impose aucune mise à jour côté client.
- Cependant, si la partie cliente est un client lourd, il faut installer cette application sur chacun des postes utilisateurs.

# II. Technologies côté serveur

## ■ Architecture n tiers

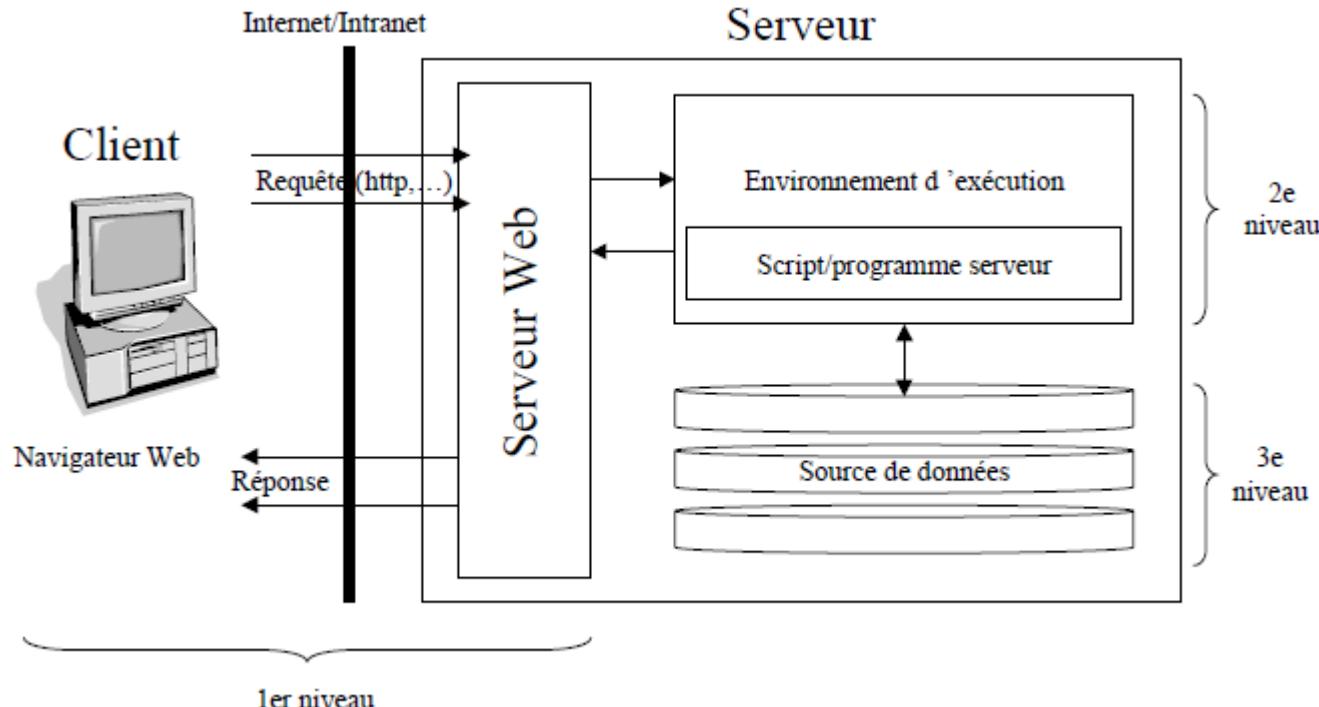


- **Serveur Web : gère les ressources statiques**
- **Serveur d'application : gère les ressources dynamiques**

## II. Technologies côté serveur

### ■ Autre façon de voir l'architecture précédente

- Le client et le serveur Web constituent le 1<sup>er</sup> niveau.



- **1<sup>er</sup> niveau : présentation (navigateur + serveur Web)**
- **2<sup>ème</sup> niveau : applicatif (script ou programme)**
- **3<sup>ème</sup> niveau : données (nécessaires au niveau 2)**

## II. Technologies côté serveur

---

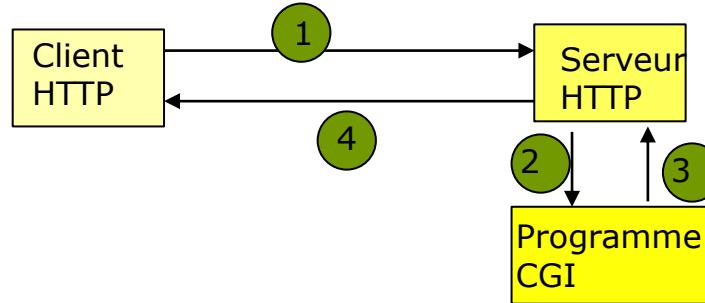
### ■ **Déroulement : une application Web**

- 1. Recueille les données utilisateur grâce à un formulaire HTML (niveau 1) ;**
- 2. Envoie d'une requête au serveur Web ;**
- 3. Exécute le programme serveur requis (niveau 2 et 3) ;**
- 4. Assemble/renvoie les données vers le navigateur (niveau 1).**

## II. Technologies côté serveur

### □ 2.2. CGI : Common Gateway Interface

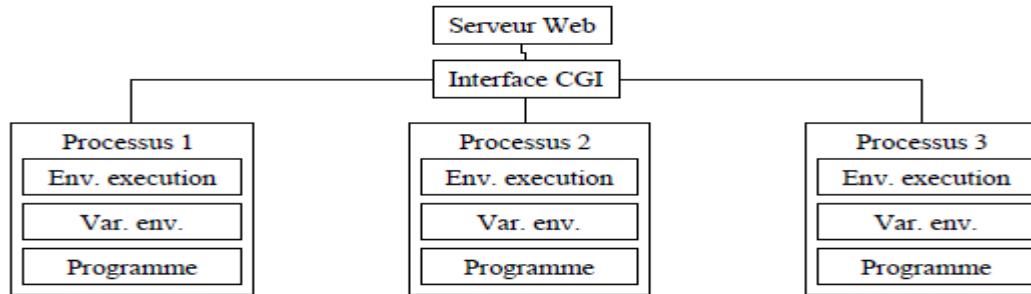
- Permet de générer par programme une page HTML
- Principe de fonctionnement



- 1. Le navigateur indique l'URL `http://www.monSite.monProgCgi`. Ce lien désigne le programme cgi sur le serveur (script ou d'un exécutable).**
- 2. Le serveur http lance l'exécution du programme.**
- 3. Le programme cgi s'exécute et construit une page html à la volée. Cette page est retournée au serveur http.**
- 4. Le serveur http transmet la page construite précédemment au navigateur. Celui-ci reçoit donc du texte html comme d'habitude. Il l'interprète et affiche le résultat.**

## II. Technologies côté serveur

- Chaque requête lance un processus sur le serveur Web



### ■ Avantages :

- gratuit, pris en charge par tous les serveurs Web actuels
- peut être écrit dans n'importe quel langage (surtout perl)

### ■ Inconvénients :

- assez lent ;
- parfois difficile à développer ;
- manque d'évolutivité (plusieurs processus créés) ;
- serveur très sollicité si plusieurs requêtes au même moment
- amélioration :
  - **FastCGI** : instance partagée des programmes CGI ;
  - **mod\_perl** (Apache) : script CGI interprété et exécuté dans le serveur Web.

## II. Technologies côté serveur

---

### □ 2.3. ISAPI et NSAPI

- Comparables aux CGI

- Mais utilisent une bibliothèque de liens dynamiques DLL (Dynamic Link Library).

- ISAPI (Microsoft IIS)

- Internet Server Application Programming Interface.

- NSAPI (Netscape)

- Netscape Server Application Programming Interface.

- Démarrage plus rapide que les programmes CGI pour 2 raisons :

- Un programme côté serveur est démarré par le serveur Web comme un thread séparé.

- Un programme CGI standard est lancée comme un processus séparé.
    - Un thread peut être démarré plus vite que d'un nouveau processus.

- Le programme côté serveur est compilé en tant que DLL

- conserve le serveur chargé après que le programme ait été exécuté une fois.
    - Le redémarrages ultérieurs sont beaucoup plus rapides.

## II. Technologies côté serveur

---

- **2.3. PHP**
  - **PHP (Hypertext PreProcessor)**

## II. Technologies côté serveur

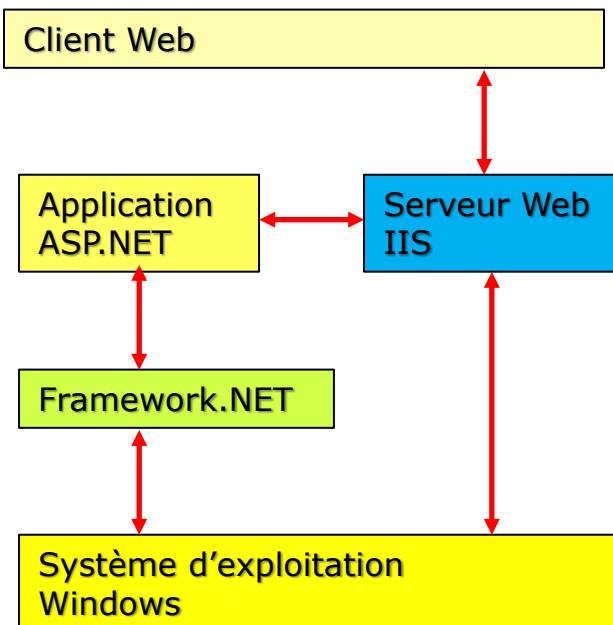
---

### □ 2.4. ASP.NET (Microsoft)

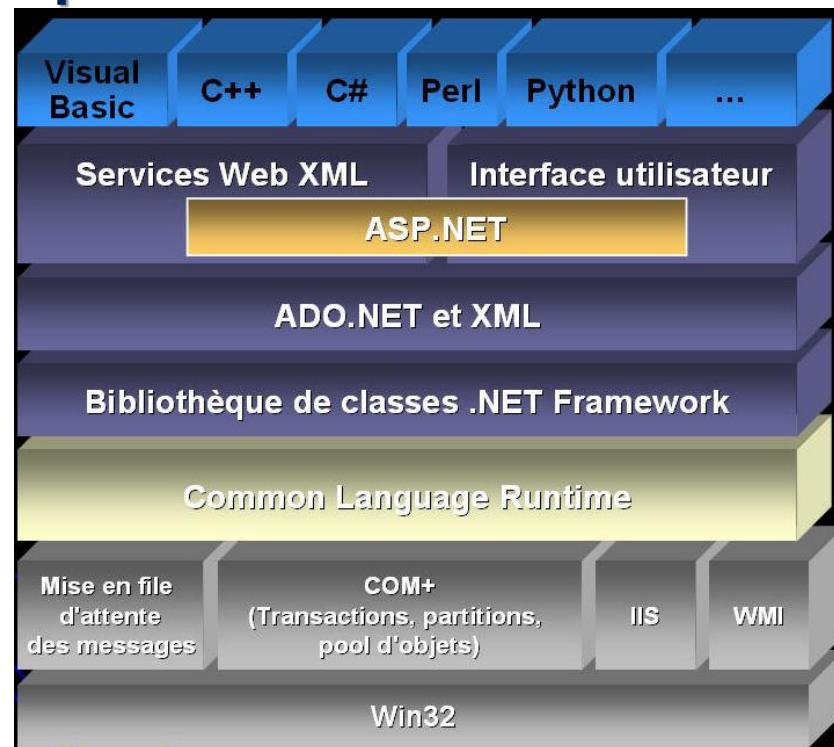
- Active Server Page .NET
- Basé sur la technologie .NET
- ASP.NET est le successeur des ASP
- Conçu pour être compatible avec davantage de types de navigateurs et de langages.
- Permet la programmation d'applications Web dynamiques, du côté du serveur.
- Contrairement à asp, où le code était inclus directement dans la partie html, asp.NET est un langage compilé.
  - Il est possible de travailler de manière séparée et d'avoir des personnes spécialisées pour chaque partie de la page.
  - Il suffit de connaître l'interface entre le code et ce qui est affiché, c'est à dire le nom des WebForms.
  - Le langage de développement utilisé peut être tout langage compatible .NET (C#, VB.NET, J#), Delphi, Perl, Python, ...).

## II. Technologies côté serveur

### ■ Architecture d'ASP.NET



### Composants de Framework.NET



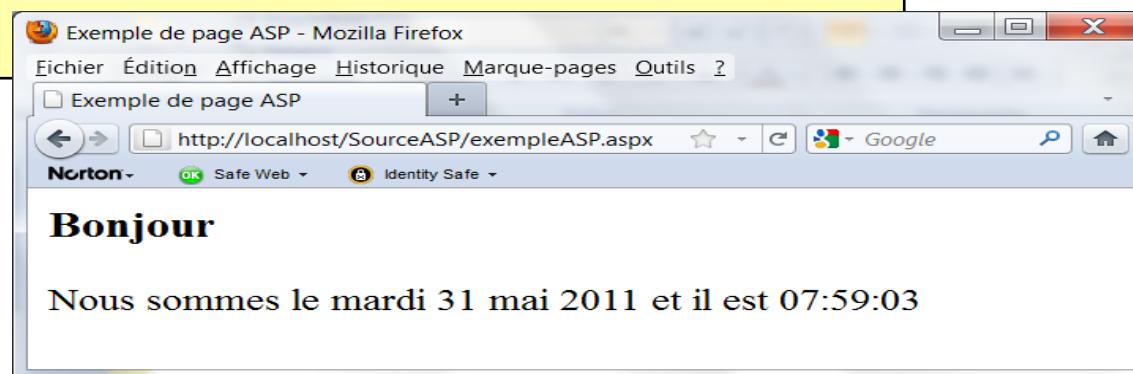
### ■ Framework .NET

- Plateforme de développement et d'exécution
- Permet à divers langages de programmation et bibliothèques de travailler ensemble de façon transparente afin de créer des applications Windows, Web, Mobile, ou Office.

# II. Technologies côté serveur

## ■ Exemple de page ASP

```
<HTML>
<HEAD> <TITLE>Exemple de page ASP</TITLE> </HEAD>
<script language="vb" runat=server>
Sub Page_Load()
    Dim aujourd'hui as DateTime = DateTime.Now
    Dim d1, d2 as String
    d1 = aujourd'hui.ToString("dddd dd MMM yyyy")
    d2 = aujourd'hui.ToString("HH:mm:ss")
    Message.Text = "Nous sommes le " & d1 & " et il est " & d2
End Sub
</script>
<BODY>
    <H3>Bonjour</H3>
    <asp:label id="Message" runat="Server" />
</BODY>
</HTML>
```



## II. Technologies côté serveur

---

### □ 2.5. Servlets et JSP (Sun)

- **Programmes Java s'exécutant côté serveur Web en tant qu'extension du serveur.**
- **Elle reçoit une requête du client, elle effectue des traitements et renvoie le résultat.**
- **Servlet**
  - **Réponse de Java aux CGI**
  - **Programme "autonome" stockés dans un fichier .class sur le serveur. S'exécute sur le serveur et retournant des pages Web dynamiques (à la volée)**
  - **Exécutées sur un serveur d'applications, elles interceptent les requêtes des navigateurs web et génèrent des réponses (souvent en DHTML) construites dynamiquement à l'aide du langage Java.**

## II. Technologies côté serveur

---

### ■ JSP

- **Réponse aux ASP, PHP, perl, etc...**
- **Programme source Java embarqué dans une page .html.**

	Côté client	Côté serveur
Classe autonome	Applet	Servlet
Embarqué dans une page HTML	Javascript	JSP

- **Extension des servlets en suivant le modèle JEE qui sépare :**
  - Les contenus statiques et dynamiques ;
  - La présentation et la logique métier, notamment grâce aux JavaBeans.
- **Exécutable avec tous les serveurs Web (IIS, Apache)**
  - Auxquels on a ajouté un moteur de Servlet/JSP tel que Tomcat
  - Les pages JSP sont converties en Servlet par le moteur de Servlets lors du premier appel à la JSP.

## II. Technologies côté serveur

---

### ■ Comparaison JSP/Servlet

- **JSP : compilé en servlet**
- **Servlet : possibilité de distinguer les requêtes HTTP**
  - **doPut, doGet, doPost, ...**
- **JSP : beaucoup de HTML, peu Java**
- **Servlet : beaucoup de Java, peu HTML**
- **contenu autre que HTML (PDF, GIF, Excel, ...) : oui**
- **session, chaînage, redirection : oui dans les 2 cas**
- **servlet : pur Java : facilement éditable dans IDE**
- **JSP : plutôt éditeur de pages HTML**
- **servlet compilation avant déploiement / JSP après**
- **JSP à redéployer si erreur**

# III. Serveur Web

---

## □ 3.1. Présentation

### ■ Synonymes :

- **Serveur Web ou serveur HTTP ou daemon HTTP ou HTTPd (HTTP daemon).**

### ■ Définition

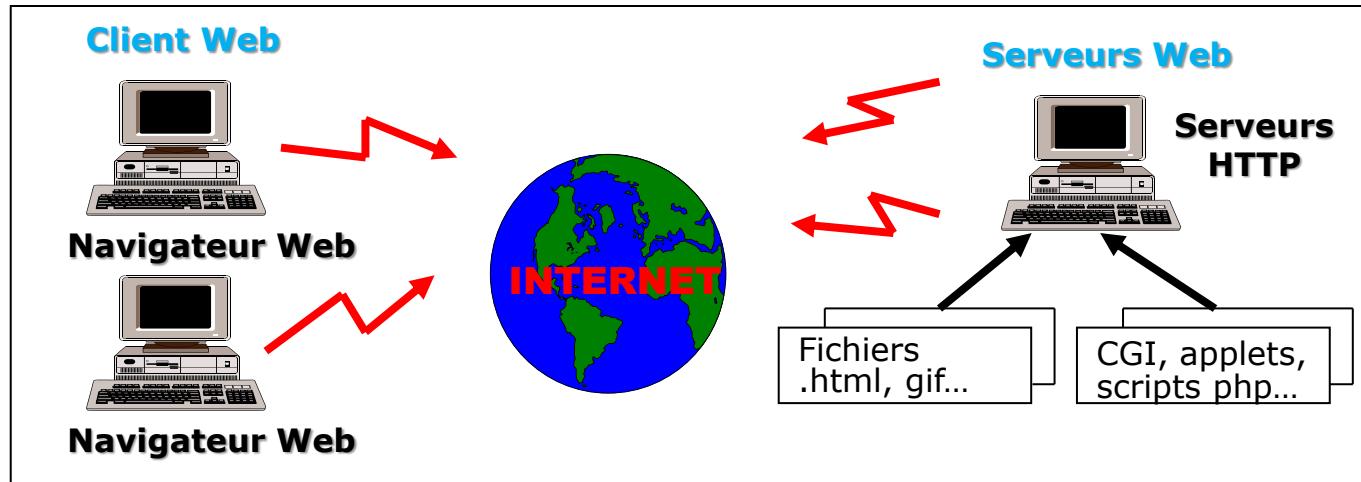
- **Logiciel servant des requêtes respectant le protocole de communication client-serveur HTTP (Hypertext Transfer Protocol) développé pour le World Wide Web.**

### ■ Serveurs Web les plus utilisés :

- **Apache HTTP Server de la Apache Software Foundation**
  - **80% des sites Web en 2021**
- **Internet Information Services (IIS) de Microsoft ;**

# III. Serveur Web

## ■ Fonctionnement client/serveur



# III. Serveur Web

---

## □ 3.2. Intégrés

- **Package gratuit d'installation facile de Apache, PHP et MySQL.**
- **EasyPHP : version**
  - Site de téléchargement :
    - [www.easypht.org](http://www.easypht.org)
- **WampServer : Version 2.0i**
  - **PHP version 5.3.0**
  - **Apache version 2.2.11**
  - **MySQL version 5.1.36**
  - Site de téléchargement :
    - [www.wampserver.com/download.php](http://www.wampserver.com/download.php)

# III. Serveur Web

---

## □ 3.3. Répertoire de publication

- Pour que les pages PHP soient interprétées, il faut que les fichiers soient dans :
  - le répertoire www ("C:\wamp\www")
  - ou dans un autre dossier avec un alias.
- Visualiser les pages à l'aide de "Web local"
  - Créer le dossier coursPHP dans www
  - Copier le fichier ex1PHP.php dans le dossier coursPHP
  - Cliquer sur l'icône de Wamp/localhost

# III. Serveur Web

The screenshot shows a Mozilla Firefox browser window displaying the WAMP5 control panel. The title bar reads "Accueil WAMP5 - Mozilla Firefox". The address bar shows "http://localhost/". The main content area displays the WAMP5 logo and version information. It includes sections for server configuration, tools, and project management.

**WAMP5**  
PHP 5 - Apache 2 - MySQL 5

Version 1.7.2 - english version

**Configuration Serveur**

Version de Apache:	Apache/2.2.4 (Win32)
Version de PHP:	5.2.3 <i>Extensions chargées:</i> <i>bcmath, calendar, com_dotnet, ctype, session, filter, ftp, hash, iconv, json, odbc, pcre, Reflection, date, libxml, standard, tokenizer, zlib, SimpleXML, dom, SPL, wddx, xml, xmlreader, xmlwriter, apache2handler, mbstring, mysql, mysqli, PDO, pdo_sqlite, SQLite</i>
Version de MySQL:	<i>MySQL n'est pas lancé ou votre configuration phpmyadmin n'est pas bonne.</i>

**Outils**

- [phpinfo\(\)](#)
- [PHPmyadmin 2.10.1](#)
- [SQLitemanager 1.2.0](#)

**Vos projets**

- [coursPHP](#)

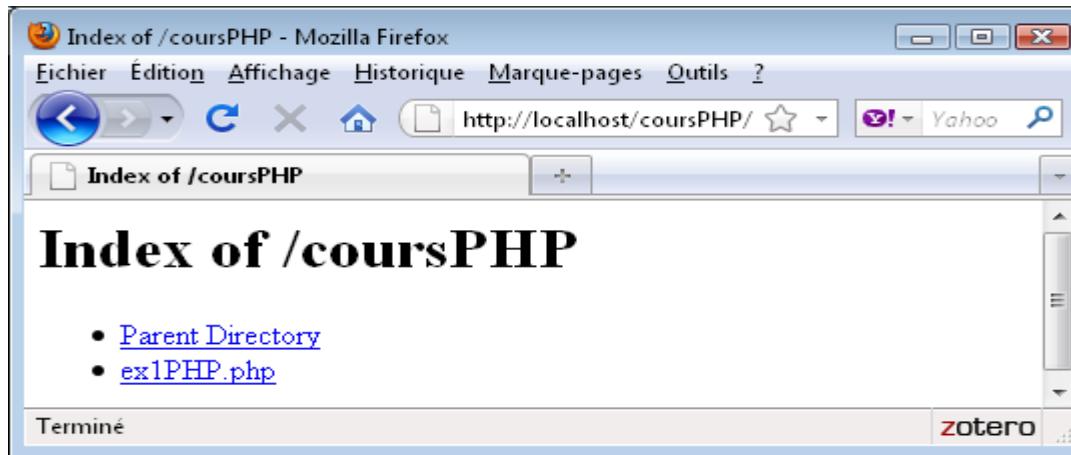
**Vos alias**

- [cnec](#)
- [coursPhp](#)
- [dmi](#)
- [MySQLPhp](#)

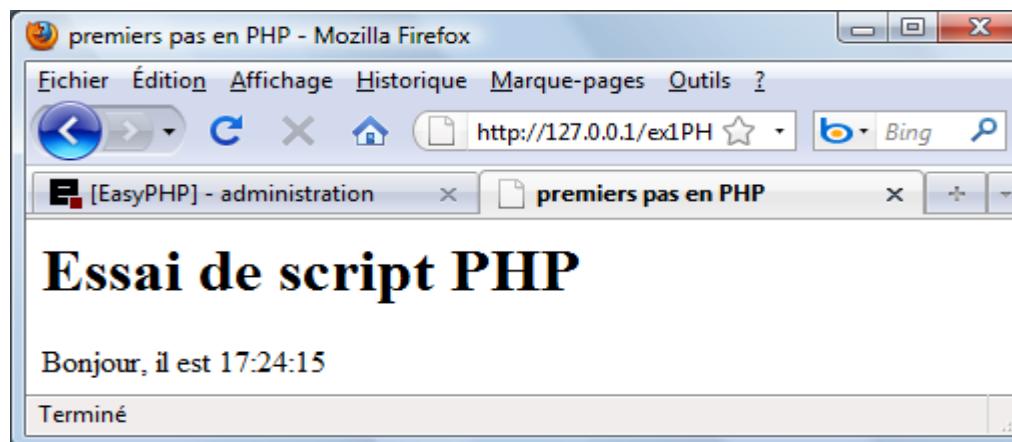
Terminé

# III. Serveur Web

## □ Cliquer sur le dossier coursPHP



## □ Cliquer sur le fichier ex1PHP.php



# III. Serveur Web

## ■ Utilisation d'un alias

- Raccourci vers un dossier
- Cliquer sur l'icône de Wamp/Apache/Alias directories/Add an alias

```
php c:\wamp\php\php.exe
Enter the destination of your alias.
For example,
'c:/test/'
would make http://localhost/coursPhp/ point to
c:/test/
: E:\Cours\Cours2010\Webmaster\Support\Source\SourcePHP
Alias created. Press Enter to exit...
```

- Saisir l'alias et appuyer sur la touche Entrée
- Saisir ou coller le chemin du dossier contenant les scripts php et appuyer sur la touche Entrée
- Cliquer sur l'icône de Wamp/localhost

# III. Serveur Web

The screenshot shows a Mozilla Firefox browser window displaying the WAMP5 control panel at <http://localhost/>. The page includes a logo for WAMP5 (PHP 5 - Apache 2 - MySQL 5), version information (Version 1.7.2 - english version), and sections for server configuration, tools, and projects.

**Configuration Serveur**

- Version de Apache: Apache/2.2.4 (Win32)
- Version de PHP: 5.2.3  
*Extensions chargées:*  
bcmath, calendar, com\_dotnet, ctype, session, filter, ftp, hash, iconv, json, odbc, pcntl, Reflection, date, libxml, standard, tokenizer, zlib, SimpleXML, dom, SPL, wddx, xml, xmlreader, xmlwriter, apache2handler, mbstring, mysql, mysqli, PDO, pdo\_sqlite, SQLite
- Version de MySQL: MySQL n'est pas lancé ou votre configuration phpmyadmin n'est pas bonne.

**Outils**

- [phpinfo\(\)](#)
- [PHPMyAdmin 2.10.1](#)
- [SQLiteManager 1.2.0](#)

**Vos projets**

- [coursPHP](#)

**Vos alias**

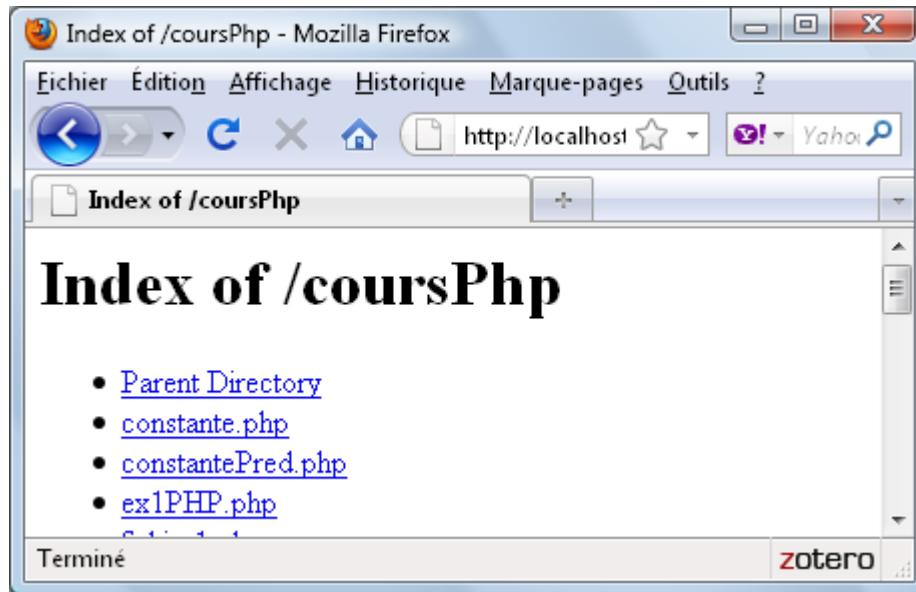
- [cnec](#)
- [coursPHP](#)
- [dmi](#)
- [MySQLPHP](#)

Terminé zotero

▣ Cliquer sur l'alias

# III. Serveur Web

---



- Cliquer sur le fichier **ex1PHP.php**

# III. Serveur Web

---

## □ 3.4. Changer le port de apache

- Port 80 occupé par un autre serveur web.
- Fichier httpd.conf de Apache  
(C:\wamp\bin\apache\Apache2.2.11\conf)
  - Remplacer :

Champ initial	Champ modifié
Listen 80	Listen 82
ServerName localhost:80	ServerName localhost:82

- Relancer le serveur Wamp

# III. Serveur Web

---

## □ 3.5. Accès au site web via un autre poste

1. Ouvrir le fichier "C:\wamp\conf\httpd.conf de la machine serveur.
2. Modifier les champs comme suit :
  - Toutes les adresse IP "127.0.0.1" par l'adresse IP de la machine;
  - Tous les noms par défaut "localhost" par le nom de la machine

Champ initial	Champ modifié
Listen 127.0.0.1:80	Listen 172.16.36.168:80
ServerAdmin admin@localhost	ServerAdmin admin@kmbrou
ServerName localhost	ServerName kmbrou
Toutes les adresses IP 127.0.0.1	172.16.36.168:80

# III. Serveur Web

---

## 3. Faire de même pour le fichier

"C:\wamp\apps\phpmyadmin3.2.0.1\config.inc.php"

### ■ NB . Dupliquer config.sample.php en config.inc.php

Champ initial	Champ modifié
\$cfg['Servers'][\$i]['host'] = 'localhost';	\$cfg['Servers'][\$i]['host'] = 'broukm';

## 5. Saisir l'adresse sur le même poste ou sur un autre poste connecté en réseau :

**http://kmbrou/coursphp/ex1PHP.php**

### □ Ne pas oublier le port s'il a été modifié

**http://kmbrou:82/coursphp/ex1PHP.php**

# IV. Concepts de base de PHP

## □ 4.1. Opérateurs

Arithmétiques		Chaînes	Booléens		Binaires	
+	addition	.	concaténation	not ou !	négation	ou
-	soustraction			or ou	ou logique	& et
*	multiplication			and &&	et logique	~ not
/	division					^ xor
%	modulo					
++	incrémentation					
--	décrémentation					

Comparaison	
\$x == \$y	vrai si \$x = \$y
\$x != \$y	vrai si \$x différent de \$y
\$x <= \$y	vrai si \$x inférieur ou égal à \$y
\$x >= \$y	vrai si \$x supérieur ou égal à \$y
\$x > \$y	vrai si \$x supérieur à \$y
\$x < \$y	vrai si \$x inférieur à \$y
\$x === \$y	vrai si \$x = \$y et qu'ils sont du même type
\$x !== \$y	vrai si \$x différent de \$y ou qu'ils n'ont pas le même type

Affectation	
\$x += \$y	\$x = \$x + \$y
\$x -= \$y	\$x = \$x - \$y
\$x *= \$y	\$x = \$x * \$y
\$x /= \$y	\$x = \$x / \$y
\$x %= \$y	\$x = \$x % \$y
\$x++	\$x = \$x + 1
\$x--	\$x = \$x - 1

# IV. Concepts de base de PHP

## □ 4.2. Variables

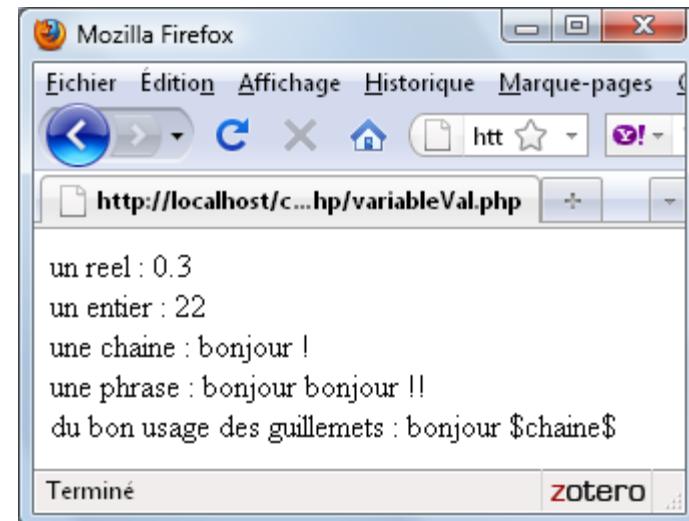
### ■ Syntaxe

- **\$nomVariable = expression;**
- **PHP respecte la casse (\$i différent de \$I) ;**
- **le nom de la variable commence avec une lettre ou un souligné et est constitué de lettres, chiffres, souligné ;**
- **pas de déclaration préalable ;**

### ■ Assignation par valeurs (variableVal.php)

```
<?php
$reel = 0.3;
$entier = 22;
$chaine = "bonjour !";
$phrase1 = "bonjour $chaine!";
$phrase2 = 'bonjour $chaine$';

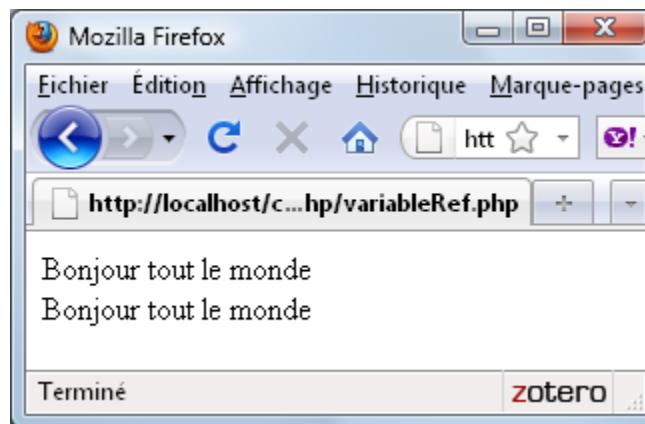
print("un reel : $reel<BR>");
print("un entier : $entier<BR>");
print("une chaine : $chaine<BR>");
print("une phrase : $phrase1<BR>");
print("du bon usage des guillemets : $phrase2");
?>
```



# IV. Concepts de base de PHP

## ■ Assignton par référence (variableRef.php)

```
<?php
$chaine1 = "Bonjour";
$chaine2 = &$chaine1; // Reference $chaine1 par $chaine2.
$chaine2 = "$chaine2 tout le monde";
print("$chaine1<BR>");
print("$chaine2");
?>
```

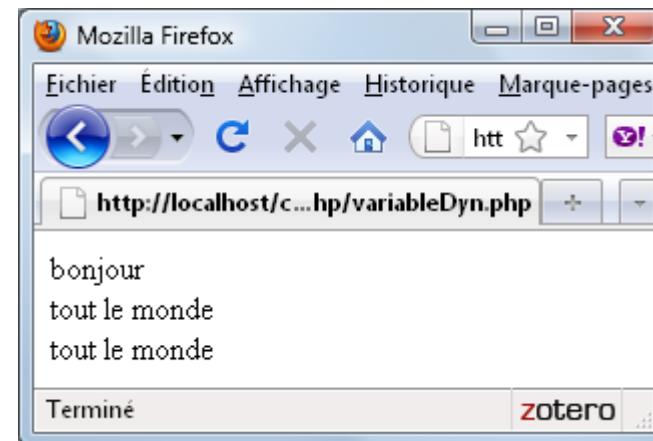


# IV. Concepts de base de PHP

## □ 4.3. Variables dynamique

- Prend comme nom la valeur d'une autre variable.
- Syntaxe
  - Affectation : **\$\$nom\_var = valeur;**
  - Lecture de la valeur : **\$\${\$nom\_var}**

```
<?php  
$ch = "bonjour";  
$$ch = "tout le monde";  
echo "$ch<BR>";  
echo "{$ch}<BR>";  
echo "$bonjour<BR>";  
?>
```



# IV. Concepts de base de PHP

## □ 4.4. Variables d'environnement

### ■ Variables prédéfinies

□ La plupart dépendent du serveur qui appelle le script php.

### ■ Liste complète des variables (variableEnv.php)

□ <?php phpinfo();?>

<b>System</b>	Windows NT KMBROU 6.0 build 6002
<b>Build Date</b>	May 31 2007 09:36:39
<b>Configure Command</b>	cscript/nologo configure.js "--enable-snapshot-build" "--with-gd=shared"
<b>Server API</b>	Apache 2.0 Handler
<b>Virtual Directory Support</b>	enabled
<b>Configuration File (php.ini) Path</b>	C:\Windows

Terminé

zotero

# IV. Concepts de base de PHP

---

## □ 4.5. Fonctions de conversion

### ■ Type donné automatiquement

- **integer, double, string, array, object**

### ■ Fonction

- **intval, doubleval :**

- \$entier = intval(\$reel);

- **Cast : (integer), (string), (double) :**

- \$entier = (integer)\$reel;

- **booleen**

- settype(var, type)

- **chaine**

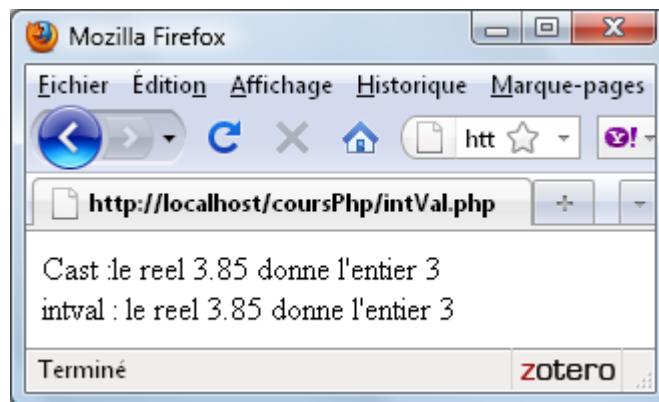
- strval(argument)

- retourne l'argument sous forme de chaine

# IV. Concepts de base de PHP

## ■ Exemple : intval.php

```
<?php
$reel = 3.85;
$entier1 =(integer)$reel;
$entier2 =intval($reel);
print("Cast :le reel $reel donne l'entier $entier1<BR>");
print("intval : le reel $reel donne l'entier $entier2");
?>
```



# IV. Concepts de base de PHP

---

## □ 4.6. Fonctions pour connaître le type

- **chaine gettype(variable)**
- **booleen is\_array(var)**
- **booleen is\_double(var)**
- **booleen is\_integer(var)**
- **booleen is\_object(var)**
- **booleen is\_string(var)**
- **booleen is\_type(var, type)**
  - renvoie 1 si var est du type : type

# IV. Concepts de base de PHP

## □ 4.7. Constantes

- définition de la valeur de la constante :
  - `define ("NomConst", valeur);`

- Utilisation :

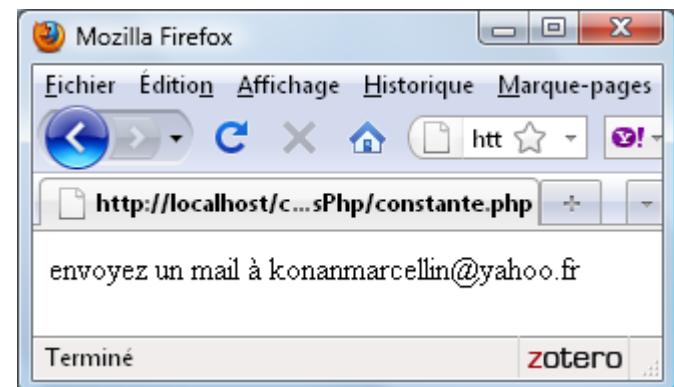
- `NomConst`

- vérification d'existence :

- `if(defined(NomConst))...`

- Exemple : Consante.php

```
<?php
define("MAILBROU", "konanmarcellin@yahoo.fr");
print("envoyez un mail à ". MAILBROU);
?>
```



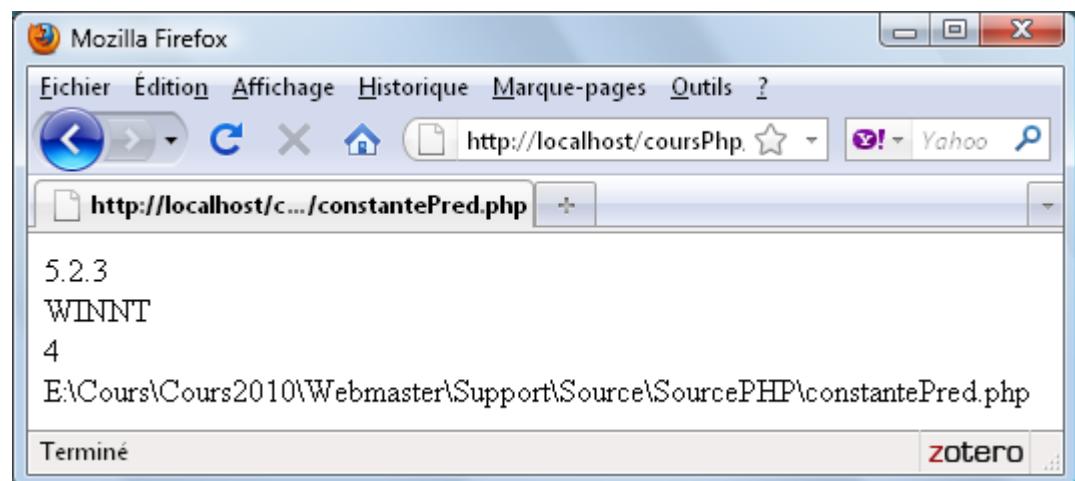
# IV. Concepts de base de PHP

## ■ Constantes prédéfinies :

Constante	Explication
TRUE	vrai
FALSE	faux
__FILE__	nom du fichier du script parsé (attention il y a deux caractères soulignés avant FILE et deux après)
__LINE__	numéro de la ligne du script
PHP_VERSION	version de PHP
PHP_OS	système sur lequel PHP exécute le script)

## ▣ Exemple : constantePred.php

```
<?php  
echo PHP_VERSION."<BR>";  
echo PHP_OS."<BR>";  
echo __LINE__."<BR>";  
echo __FILE__."<BR>";  
?>
```



# V. Tableaux

## ■ 5.1. Tableau à une dimension

### ■ Initialisation trois possibilités équivalentes :

- `$tab = array("b","o","n");`
- `$tab[0] = "b"; $tab[1] = "o"; $tab[2] = "n";`
- `$tab[] = "b"; $tab[] = "o"; $tab[] = "n";`

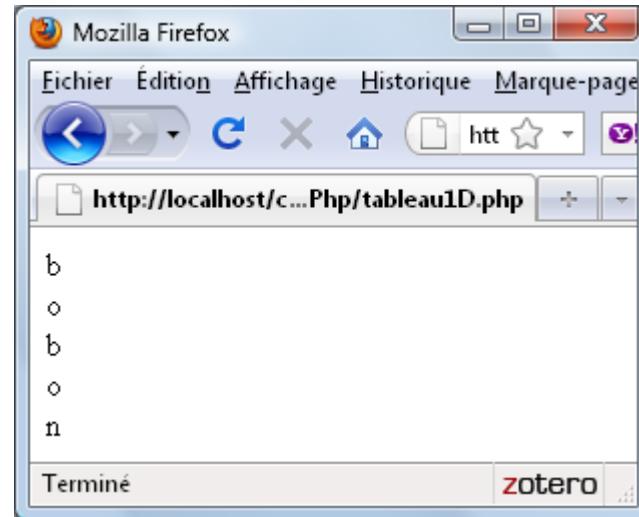
### ■ Accès à un élément d'un tableau de n cases :

- `$tab[indice]` avec indice dans l'intervalle [0,n-1]
- Exemple : **tableau1D.php**

```
<?php
//initialisation 1
$tab = array("b","o","n");

//accès a une valeur
echo $tab[0]."<BR>";
echo $tab[1];

//parcourir le tableau avec for
for($i=0 ; $i<3 ; $i++)
    echo "<BR>$tab[$i]";
K. M. B ?>
```



# V. Tableaux

---

## □ 5.2. Tableau associatifs

- Initialisation 1 : **\$nomtab["cle"] = valeur;**
- Initialisation 2 :
  - **\$nomtab = array("cle1"=>valeur1,"cle2"=>valeur2,...);**
- Accès à un élément : **\$individu["cle"];**
- parcourir tous les éléments, solution 1 (for) :

```
for(reset($nomtab) ; $cle = key($nomtab) ; next($nomtab))  
{ $val = pos($nomtab);  
  print("$cle = $val<BR>\n");  
}
```

- parcourir tous les éléments, solution 2 (while) :

```
reset($nomtab);  
while(list($cle, $valeur) = each($nomtab))  
{ echo "<BR>$cle = $valeur"; }
```

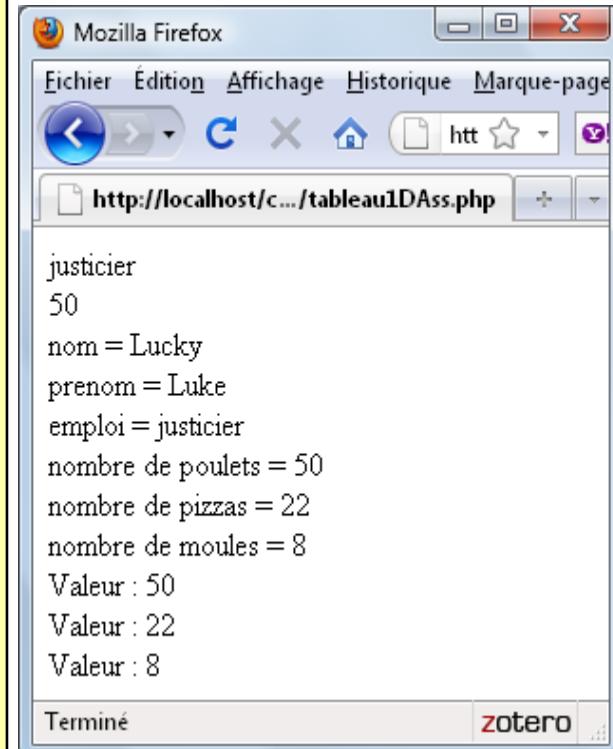
- parcourir tous les éléments, solution 3 : foreach

```
foreach ($nomtab as $value)  
  echo "<br />Valeur : $value";
```

# V. Tableaux

## ■ Exemple (tableau1DAss.php)

```
<?php
//initialisation 1
$individu["nom"] = "Lucky";
$individu["prenom"] = "Luke";
$individu["emploi"] = "justicier";
//initialisation 2
$stock= array("poulets"=>50, "pizzas"=>22, "moules"=>8);
//accès a une valeur
echo $individu["emploi"]."<BR>";
echo $stock["poulets"];
//parcourir le tableau avec for
for(reset($individu) ; $cle = key($individu) ; next($individu))
{$val = pos($individu); echo "<BR>$cle = $val"; }
//parcourir le tableau avec while
reset($stock);
while(list($cle, $valeur) = each($stock))
{echo "<BR>nombre de $cle = $valeur";}
//parcourir le tableau avec foreach
foreach ($stock as $value)
echo "<br />Valeur : $value";
?>
```



# V. Tableaux

## □ 5.3. Tableau plusieurs dimensions

### ■ Syntaxe

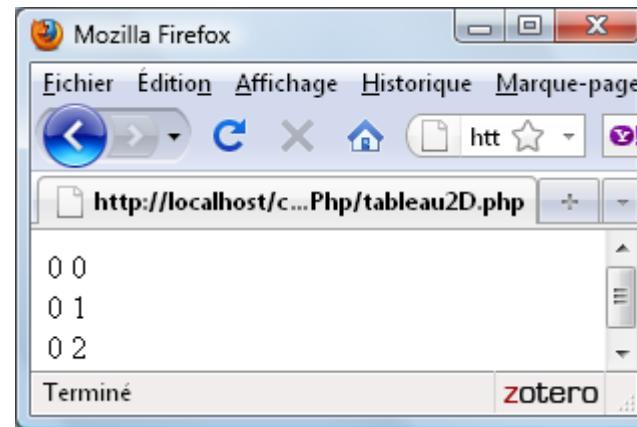
□ **\$nomtab = array(array());**

### ■ Exemple : tableau2D.php

```
<?php
$t = array(array());

//saisir le tableau avec for
for($i=0 ; $i<3 ; $i++)
    for($j=0 ; $j<2 ; $j++)
        $t[$i][$j] = $i*$j;

//parcourir le tableau avec for
for($i=0 ; $i<3 ; $i++)
    {for($j=0 ; $j<2 ; $j++)
        echo $t[$i][$j] . " ";
        echo "<BR>";
    }
?>
```



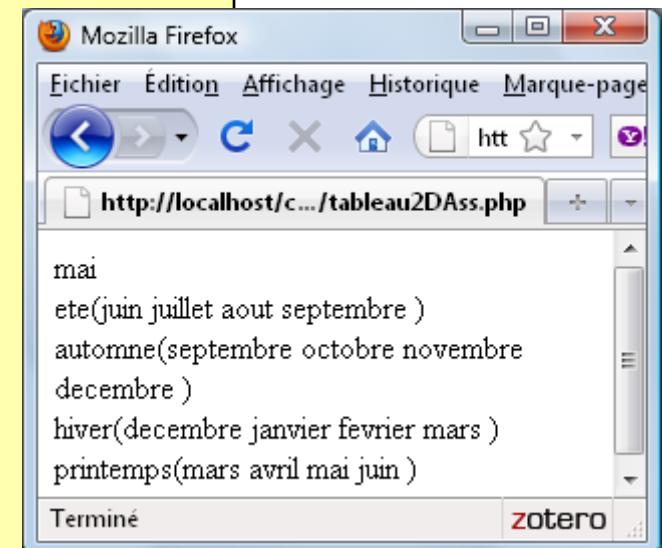
# V. Tableaux

- **Tableau associatif à plusieurs dimensions**
  - Exemple : **tableau2DAss.php**

```
<?php
$saison = array(
    "ete"=>array("juin","juillet","aout","septembre"),
    "automne"=>array("septembre","octobre","novembre","decembre"),
    "hiver"=>array("decembre","janvier","fevrier","mars"),
    "printemps"=>array("mars","avril","mai","juin"));

print($saison["printemps"][2]."<BR>");

//parcourir le tableau avec for
for(reset($saison) ; $cle = key($saison) ; next($saison))
{$val = pos($saison);
echo "$cle(";
for($i=0 ; $i<4 ; $i++)
echo "$val[$i] ";
echo ")<BR>";
}
?>
```



# V. Tableaux

---

## □ 5.4. Fonctions pour les Tableaux

### ■ Fonctions classiques

count(tab)	nb d'éléments
sizeof(tab)	nb d'éléments
explode(separateur, chaine)	transforme une chaîne en tableau en fonction du sép.
implode(tab, séparateur)	opération inverse
max(tab)	retourne la plus grande valeur d'un tableau
min(tab)	retourne la plus petite valeur du tableau
shuffle(tab)	change aléatoirement l'ordre des éléments

### ■ Fonction de type pointeurs

current(tab)	valeur de l'élément courant (endroit où est le pointeur), NB pos(tab) est un alias de current.
each(tab)	clé et valeur pour l'endroit où est le pointeur sur le tableau (fait avancer de 1 le pointeur)
end(tab)	place le pointeur sur le dernier élément du tableau
key(tab)	index de l'élément courant
next(tab)	avance le pointeur d'une case, retourne sa valeur
prev(tab)	recule d'une case et retourne sa valeur
reset(tab)	pointeur sur le premier élément du tableau

# V. Tableaux

---

## ■ Fonctions de tri

arsort(tab)	trie par ordre décroissant, l'index associé se déplace avec les valeurs
asort(tab)	trie par ordre croissant, l'index associé se déplace avec les valeurs
krsort(tab)	trie le tableau par ses clés
ksort(tab)	trie le tableau en ordre inverse par ses clés
rsort(tab)	trie par ordre décroissant, efface les valeurs des clés
sort(tab)	trie par ordre croissant, efface la valeur des clés
uasort(tab, comp)	trie selon la fonction comp (définie par l'utilisateur), conserve la valeur des clés
usort(tab, comp)	trie selon la fonction, efface la valeur des clés
uksort(tab, comp)	trie les clés selon la fonction

## ■ Autres fonctions

array_count_values (tab)	retourne un tableau associatif qui donne en clé les valeurs du tableau et en valeur la fréquence <pre>\$tab=array("hi", "hi", "bonjour", "hi") \$tabfreq = array_count_values (\$tab); // \$tabfreq contient "hi"=3, "bonjour"=1.</pre>
array_diff (tab1, tab2, ...)	retourne un tableau contenant toutes les valeurs de tab1 qui ne sont pas dans tab2, ...
array_keys (tab)	retourne un tableau contenant toutes les clés de tab.
array_values (tab)	retourne un tableau contenant toutes les valeurs de tab.
array_pop (tab)	dépile et retourne le dernier élément
array_push (tab, el1, el2, ...)	empile les éléments 1, 2, ... à la fin du tableau
in_array (val, tab)	retourne vrai si la valeur est dans le tableau

# VI. Structures de contrôle

---

## □ 6.1 L'alternation

- C'est une instruction de prise de décision.

- Syntaxe 1 :

```
if(condition)
    action1 ;
[else
    action2 ;]
```

- Si la valeur de condition est différente de 0 alors l'action1 est exécutée sinon, si sa valeur est égale à 0 l'action 2 est exécutée.

- Le else est facultatif.

- Syntaxe 2 :

```
condition ? action1 : action2 ;
```

# VI. Structures de contrôle

---

## □ 6.2 Le choix multiple

- Généralisation de l'alternative
- Syntaxe

```
switch(expression)
{case valeur1 : action1 ; break ;
case valeur2 : action2 ; break ;
...
case valeurn : actionn ; break ;
default : actionn+1 ;
}
```

- Expression : sélecteur du switch, de type entier ou de type caractère.
- Si sa valeur correspond à l'une des valeuri, l'actioni est exécutée
- break permet de sortir du switch
- default (facultatif) correspond au cas où la valeur d'expression n'a aucune correspondance

# VI. Structures de contrôle

---

## □ 6.3 Les boucles

- **Une instruction itérative ou boucle permet de spécifier l'exécution d'un bloc d'instruction plusieurs fois.**
- **On distingue 3 types de boucles :**
  - **Boucle for (POUR)** : on connaît à priori le nombre d'itération à faire
  - **Boucle do-while (REPETER..JUSQUA)** : on ne connaît pas le nombre d'itération à faire
  - **Boucle while (TANTQUE)** : on ne connaît pas le nombre d'itération à faire.
- **Boucle while** : on l'utilise quand on n'est pas sûr que le corps de la boucle doit être exécutée au moins une fois.
  - **Syntaxe**  
**while(condition)**  
**action;**
    - Tant que la condition est vérifiée (i.e. différente de 0), le corps de la boucle (action) est exécutée.

# VI. Structures de contrôle

---

- **Boucle while** : on l'utilise quand on est sûr que le corps de la boucle doit être exécutée au moins une fois.

- **Syntaxe**

- do**

- action;**

- while(condition);**

- Le corps de la boucle est exécuté tant que la condition est vérifiée.

- **Boucle for :**

- **Instruction itérative à borne définies.**

- **Syntaxe**

- for(expr1 ; expr2 ; expr3)**

- action ;**

- expr1 : indique la condition de départ (plusieurs initialisation possibles séparées par des virgules).
      - expr2 : indique la condition d'arrêt. l'action est exécutée tant que cette condition est vérifiée. (la condition est testée avant l'exécution de l'action).
      - expr3 : indique ce qu'il faut faire à chaque itération (plusieurs instruction possibles séparées par des virgules).

# VI. Structures de contrôle

---

## ■ Boucle foreach

### □ Syntaxe

**foreach(\$tab as \$value)**

**{ instructions; }**

- Pour tout le tableau tab, à chaque itération la valeur de l'élément courant est assignée à la variable \$value et le pointeur sur le tableau est avancé d'une case.
- NB : au départ le pointeur sur le tableau est automatiquement mis sur la première case par le foreach.

## ■ Interruption des boucles : break et continue

- L'instruction **break** permet de sortir d'une boucle lorsqu'une condition est vérifiée.
- L'instruction **continue** permet de ne pas exécuter parfois le corps de la boucle lorsqu'une condition est vérifiée.

# VII. Fonctions

---

## □ 6.1. Déclaration et appel

### ■ Déclaration

```
function nomFonction($arg1, &$arg2,  
                      $arg3="defaultValue",...,$argn)  
{ instructions ; }
```

### □ Avec :

- arg1 est passé par valeur ;
- arg2 est passé par adresse : si sa valeur est modifiée dans la fonction, elle est modifiée pour tout le programme ;
- arg3 est optionnel : il peut ne pas être passé lors de l'appel de la fonction puisqu'il a une valeur par défaut.

### ■ Appel :

□ **nomfonction(param1,param2,param3..., paramn);**

### ■ Pour retourner une valeur : return

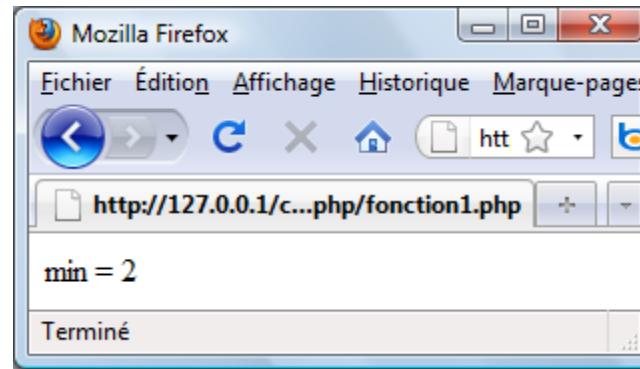
### ■ Remarques

□ **Chaque appel de fonction demande du temps CPU.**

# VII. Fonctions

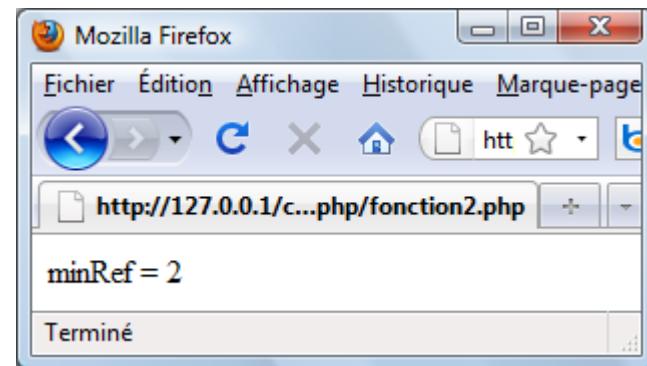
## ■ Exemple 1 : mode de passage par valeur (Fonction1.php)

```
<?php
function minimum($x, $y)
{if($x < $y)
 $z = $x;
else
 $z = $y;
return $z;
}
$r = minimum(2,4);
echo "min = " . $r;
?>
```



## ■ Exemple 2 : mode de passage par référence (Fonction2.php)

```
<?php
function minimum($x, $y, &$z)
{if($x < $y)
 $z = $x;
else
 $z = $y;
}
minimum(2,4, $r);
echo "min = " . $r;
?>
```



# VII. Fonctions

---

## ■ Exercices

- Ecrire une fonction pour calculer  $n!$
- Ecrire une fonction pour calculer  $x = n! + C(n,p) = n!/(p!*(n-p)!)$

# VII. Fonctions

---

## □ 7.2 Variable locale, globale et statique

### ■ Variable locale

- Déclarée dans une fonction
- Visible uniquement dans cette fonction
- Durée de vie : celle de la fonction

### ■ Variable globale

- Déclarée en dehors de toute fonction
- Visible dans une fonctions si elle est précédée de global
- Durée de vie : celle du programme

### ■ Variable statique

- Déclarée dans une fonction précédée de static
- Visible uniquement dans cette fonction
- Durée de vie : celle du programme

# VII. Fonctions

## ■ Exemple

### □ Fonction3.php

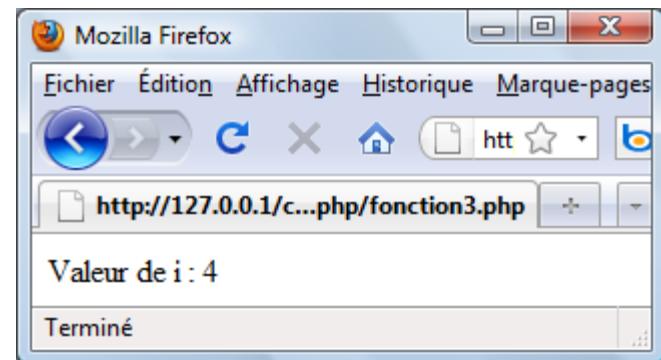
```
<?php
$chaine = "Valeur de i : ";

function inc($mode="")
{global $chaine;

static $nb=0;

$nb++; // on incrémente $nb
if($mode == "affiche")
    echo $chaine . $nb; // on affiche $nb
}

inc(); // nb == 1
inc(); // nb == 2
inc(); // nb == 3
inc("affiche"); // affiche Valeur de i : 4
?>
```



# VII. Fonctions

## □ 7.3 Retourner plusieurs valeurs

- On utilise un tableau
- Exemple : retourner le min et max de 2 valeurs

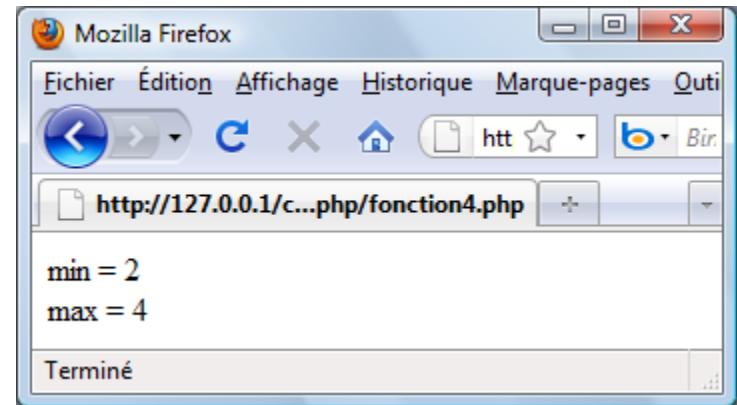
```
<?php
function minimum($x, $y)
{
$z = $x < $y ? $x : $y;
return $z;
}

function maximum($x, $y)
{
$z = $x > $y ? $x : $y;
return $z;
}

function plusPetit($x, $y)
{
$min = minimum($x, $y);
$max = maximum($x, $y);
return array($min, $max);
}

$r = array();
$r = plusPetit(2,4);
echo "min = " . $r[0] . "<BR>";
echo "max = " . $r[1];
?>
```

Fonction4.php



# VII. Fonctions

## □ 7.4 Appel dynamique

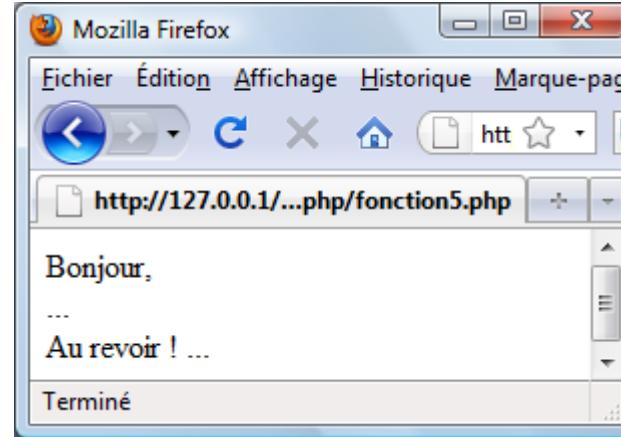
- Paramétriser une variable avec un nom de fonction.
- Utile si on ne sait pas exactement quelle fonction devra être appelée.
  - car ça dépend des données que vous allez recevoir.
  - fonction5.php

```
<?php
function affiche($chaine)
{print($chaine);}

function afficheln($chaine)
{print("$chaine<BR>");}

// corps
$afficher = "afficheln";
$afficher("Bonjour, ");
$afficher("...");
$afficher = "affiche";
$afficher("Au revoir ! ");
$afficher("...");

?>
```



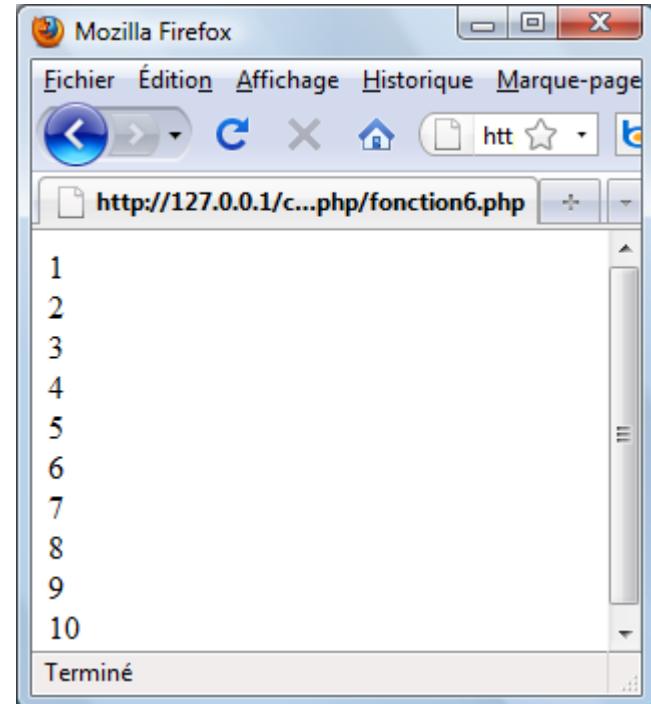
# VII. Fonctions

## □ 7.5 Récursivité

- Fonction qui s'appellent elle-même.
- fonction6.php

```
<?php
function affiche($n=0)
{$n++;
echo "$n <br>";
if($n < 10) // si n est inférieur à 10 on continue
affiche($n);
}

affiche(); // affiche les nb de 1 à 10
?>
```



# VII. Fonctions

---

## □ 7.6 Inclusion de scripts

■ Il peut être intéressant de mettre dans un fichier séparé un script qui peut servir à plusieurs autres fichiers de script.

■ Instruction include

□ Syntaxe :

**include("nom\_fichier");**

- PHP analyse le fichier à l'endroit où **include** est appelé.
- Le code inséré et interprété uniquement si l'instruction est exécutée (sauté si la valeur testée dans un if est false par exemple).

■ Instruction require

□ Syntaxe :

**require("nom\_fichier");**

□ PHP analyse le fichier à l'endroit où **require** est appelé

- Inconvénient : insère toujours le code, même si l'instruction n'est pas exécutée (ne pas mettre dans un if, switch, while, for)
- Avantage : le code est inséré au premier passage de l'analyseur PHP -> rapidité.

# VII. Fonctions

## ■ Exemple

**PhpSource.php**

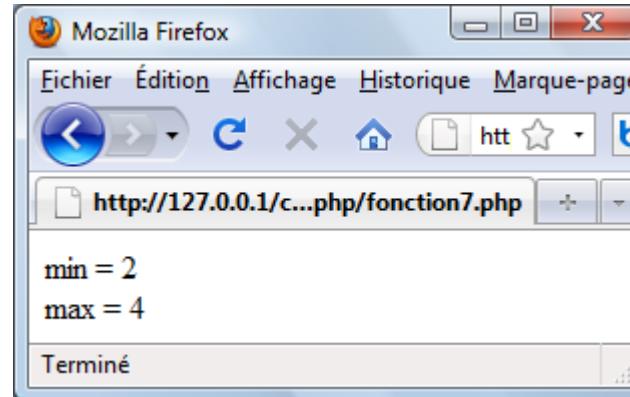
```
<?php
function minimum($x, $y)
{$z = $x < $y ? $x : $y;
return $z;
}

function maximum($x, $y)
{$z = $x > $y ? $x : $y;
return $z;
}

function plusPetit($x, $y)
{$min = minimum($x, $y);
$max = maximum($x, $y);
return array($min, $max);
}
?>
```

**fonction7.php**

```
<?php
include('phpSource.php');
$r = array();
$r = plusPetit(2,4);
echo "min = " . $r[0] . "<BR>";
echo "max = " . $r[1];
?>
```



# VII. Fonctions

## □ 7.7 Fonctions prédéfinies

### ■ Fonctions pour les chaînes

Séquences d'échappement	
\n	Retour à la ligne
\t	tabulation
\\	\
\\$	\$
\"	"

eval(chaine)	Evalue la chaine comme si c'etait du code PHP
entier strlen(ch)	retourne la longueur de ch
tab count_chars(ch)	retourne le nombre d'occurrences de chaque caractère de la chaîne (PHP4)
chaine strtok(ch, sep)	Separe ch en fonction de sep
chaine quotemeta(ch)	retourne la chaîne avec un \ devant les caractères . \\ + * ? [ ^ ] ( \$ )
chaine str_repeat (ch, n)	Répète n fois la chaîne (PHP4)

conversion	
chaine sprintf(format, arg)	retourne une chaine formatée
chaine chr(ascii)	donne le caractere qui correspond au code ascii
int ord(car)	donne le code ascii qui correspond au caractère
chaine pack(format,arg)	\$ch = pack("ccc", 66, 79, 78); affichera bon
chaine sql_regcase(exp)	convertit une exp. rationnelle. en une exp rationnelle non sensible à la casse

# VII. Fonctions

---

<b>position, sous-chaîne</b>	
entier strpos(ch, ssch)	retourne la position de la 1ère occurrence de ssch dans ch
entier strrpos(ch, car)	retourne la position de la dernière occurrence du caractère
entier strspn(ch, enscar)	longueur de la sous-chaîne dont les caractères sont entièrement dans enscar
chaine strstr(ch, ssch)	retourne la portion de ch à partir de la 1ère occurrence de ssch et jusqu'à la fin
chaine stristr(ch, ssch)	strstr non sensible à la casse
chaine substr(ch, deb, l)	renvoie la sous-chaîne de taille l qui commence à l'indice deb

<b>comparaison</b>	
entier strcasecmp(ch1, ch2)	comparaison sans tenir compte de la casse
entier strcmp(ch1, ch2)	retourne 0 si ch1 = ch2

<b>casse</b>	
chaine strtolower(ch)	convertit en minuscules
chaine strtoupper(ch)	convertit en majuscules
chaine ucfirst(ch)	met la première lettre en majuscule
chaine ucwords(ch)	met tous les mots de ch en majuscule

# VII. Fonctions

---

<b>url, html</b>	
chaine htmlentities(ch)	convertit les car de la chaine au format HTML
chaine htmlspecialchars (ch)	convertit les caractères spéciaux en entités html (< -> &lt; ;)
chaine nl2br(ch)	ajoute   devant chaque nouvelle ligne du texte
parse_str(requete)	analyse la requête comme si elle venait d'un formulaire posté par get (crée les variables et leur valeur)
tableau parse_url(requete)	stocke une URL dans un tableau associatif (scheme, host, path, query).
chaine rawurlencode(URL)	convertit en texte la chaine au format URL (%)
chaine rawurldecode(ch)	convertit ch au format URL (%)
chaine urldecode(URL)	conversion en texte de l'URL (pas pour les données binaires)
chaine urlencode(ch)	conversion de la chaine au format URL (par pour bin)

<b>espaces</b>	
chaine chop(ch)	retourne la chaine sans les espaces
chaine trim(ch)	supprime les espaces de début et fin de chaîne

# VII. Fonctions

## ■ Expressions rationnelles

	ou
*	0 ou +
+	au moins une fois
?	0 ou 1 fois
{n}	n fois
{n,}	n fois ou plus
{n, m}	au moins n et au plus m

.	un caractère quelconque
^	correspondance au début
\$	correspondance en fin
[a-z]	tout caractère minuscule
[ab]	a ou b
[^ab]	tout sauf a et b
:alpha:]	type de caractère (alnum, blank, digit, punct)
[:<:]c	mot commence par c
[>:]c	mot finit par c

## ■ Fonctions pour les expressions rationnelles

booleen ereg(exp, ch, tabocc)	évalue l'expression et met les occurrences rencontrées dans ch dans le tableau tabocc la case 0 contient l'occurrence de l'expression, les autres cases les sous-occurrences. Retourne 0 si pas de correspondance
chaine ereg_replace(exp, rep, ch)	remplace les sous-chaînes de ch qui correspondent à l'expression exp par la chaîne rep. exp peut être un code ascii ereg_replace(10, " ", ch) remplace les \n par des  
booleen eregi(exp, ch, tabocc)	comme ereg sans tenir compte de la casse
booleen eregi_replace(exp, rep, ch)	comme ereg_replace sans tenir compte de la casse
tableau split(exp, ch, limite)	retourne un tableau des sous-chaînes

# VII. Fonctions

---

## ■ Exemple

```
<?php
// scinde la phrase grâce aux virgules et espacements
// ce qui inclus les " ", \r, \t, \n et \f
$t = preg_split("/[\s,]+/", "langage hypertexte, programmation");
print_r($t);
?>
```

t[0] => langage  
t[1] => hypertexte  
t[2] => programmation

# VII. Fonctions

---

## ■ Fonctions mathématiques

trigonométrie	
décimal acos(x)	arc cosinus
décimal asin(x)	arc sinus
décimal atan(x)	arc tangente
décimal cos(x)	cosinus
décimal sin(x)	sinus
décimal tan(angle)	tangente de l'angle

arrondi	
nb abs(x)	valeur absolue
entier ceil(x)	retourne le plus petit entier > x
entier floor(x)	partie entière
entier round(val)	arrondi à l'entier le plus proche

autres	
nb abs(x)	valeur absolue
décimal exp(x)	exponentielle
décimal log(x)	logarithme
décimal pi()	retourne la valeur de pi
décimal pow(base, puissance)	élève la base à la puissance
décimal sqrt(x)	racine carrée de x

# VII. Fonctions

---

<b>conversion</b>	
chaine decbin(entier)	retourne la représentation binaire de entier
int bindec(chaine)	retourne la représentation décimale de la chaîne binaire
chaine dechex(entier)	retourne la représentation hexadécimale de entier
chaine decoct(entier)	retourne la représentation octale de entier
décimal deg2rad(angle)	donne le radian pour l'angle en degré
entier hexdec(chainehex)	conversion chaine hexadécimale en un entier
décimal octdec(chaineoct)	conversion chaine octale en un entier
décimal rad2deg(angle)	retourne les degrés pour l'angle en radian

<b>nombres aléatoires</b>	
entier getrandmax()	retourne le nb aléatoire max pour rand
entier rand(inf, sup)	retourne un nb aléatoire compris entre les entiers inf et sup
entier srand(entier)	pour initialiser le générateur de nb aléatoires

# VII. Fonctions

---

## ■ Fonctions de temps et de date

- **booleen checkdate(m, j, a) :**
  - Retourne 1 si la date passée est valide, 0 sinon
- **chaine date(format) :**
  - retourne la date

a	am ou pm	d	jour du mois sans les zéros
D	jour de la semaine en 3 lettres	F	nom du mois
h	heure de 1 à 12	H	heure de 0 à 23
i	minutes	j	jour du mois avec les zéros
I	jour de la semaine	m	chiffre du mois
M	nom du mois en abrégé	y	année sur deux unités
Y	année sur 4 unités	z	jour de l'année

# VII. Fonctions

---

- ▣ **tableau getdate() :**

- retourne un tableau associatif

clé	valeur
hours	heure format 24
minutes	heure en minutes
month	nom du mois
mday	jour du mois
mon	mois
wsay	jour (0..6)
weekday	nom du jour
yday	jour de l'année
year	année
seconds	minutes en secondes
0	Timestamp (nb de secondes depuis le 01/01/70)

- ▣ **tableau gmdate(format) :**

- comme date mais au format Greenwich

- ▣ **entier gmmktime(h, m, s, mois, j, a) :**

- comme mktime mais Greenwich

- ▣ **entier mktime(h, m, s, mois, j, a) :**

- retourne la valeur Timestamp qui correspond aux arguments

- ▣ **entier time() : retourne la valeur de Timestamp**

# VII. Fonctions

---

- **Exercice 1 :**
  - Ecrire un programme qui compte le nombre de voyelles d'une chaîne de caractères et les affiche.
- **Exercice 2 :**
  - Ecrire un programme qui compte le nombre de consonnes d'une chaîne de caractères et les affiche.
- **Exercice 3 :**
  - Ecrire un programme qui supprime les espaces d'une chaîne de caractères.
- **Exercice 4 :**
  - Ecrire un programme qui donne le mot miroir d'un mot.
  - Mot miroir de konan est nanok
- **Exercice 5 :**
  - Ecrire un programme qui test si un mot est palindrome (identique à son mot miroir) ou non.

# VIII. Fichiers

---

- **8.1 Présentation**
  - **Structure de données homogène mémorisée sur un support**
- **8.2. Ouverture et fermeture**
  - **Syntaxe**
    - **\$pointeur = fopen("nomFichier","mode");**

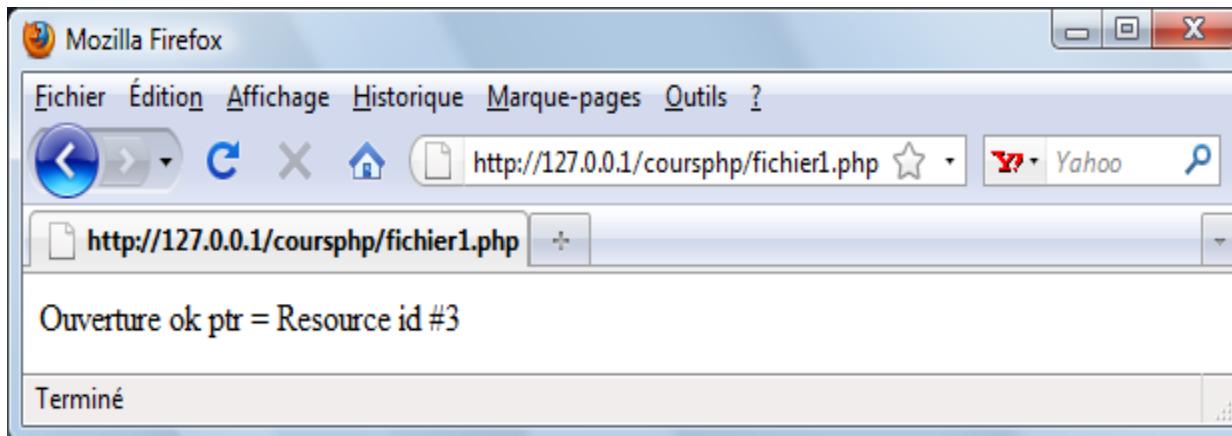
Mode	Explication
r	lecture
w	écriture
a	ajout
r+	lecture et écriture
w+	lecture et écriture, supprime le contenu précédent
a+	lecture et écriture en fin de document

- **NB : pour un fichier binaire il faut ajouter après le mode un b**

# VIII. Fichiers

## ■ Exemple : fichier1.php

```
<?php  
$ptr = fopen('toto.txt','r');  
if(!$ptr)  
{print("erreur ouverture fichier");  
 exit;  
}  
print("Ouverture ok ptr = " . $ptr);  
?>
```



# VIII. Fichiers

## ■ Ouverture et stockage dans un tableau

- **\$fichier = file("nomFichier");**
- **Exemple : fichierTableau.php**

```
<?php  
$f = file('toto.txt');  
for($i = 0; $i < count($f); $i++)  
{  
    print($f[$i]);  
}  
?>
```



# VIII. Fichiers

---

## ■ Fermeture

- **booleen fclose(\$pointeur);**
- **Exemple : fichierTableau.php**

```
<?php
$ptr = fopen('toto.txt','r');
if(!($ptr))
{print("erreur ouverture fichier");
exit;
}
print("Ouverture ok ptr = " . $ptr);
fclose($ptr);
?>
```

# VIII. Fichiers

## ■ 8.2. Lecture

### ■ Syntaxe

lecture d'un caractère	\$car = fgetc(\$pointeur);
lecture de n caractères	\$chaine = fgets(\$pointeur, n);
pour se replacer au début d'un fichier	rewind(\$pointeur);

### ■ Lecture jusqu'à la fin du fichier

```
while(!feof($pointeur))
{ $ligne = fgets($pointeur, n); }
```

- **n est le nombre de caractère par ligne**

- **feof : end of file**

- **Exemple : fichier2.php**

```
<?php
$ptr = fopen('toto.txt','r');
if(!$ptr)
{print("erreur ouverture fichier"); exit; }
print("Ouverture ok ptr = " . $ptr . "<br>");
while(!feof($ptr))
{$ligne = fgets($ptr,255);
print($ligne);
}
fclose($ptr);
?>
```

# VIII. Fichiers

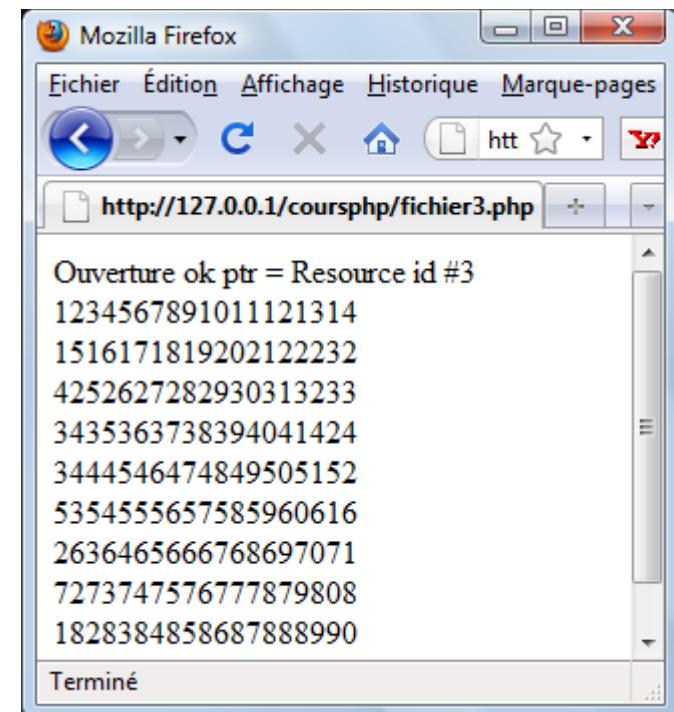
## □ 8.3. Ecriture

### ■ Syntaxe

□ **fputs(\$pointeur, "chaine");**

### ■ Exemple : fichier3.php

```
<?php
$ptr = fopen('tata.txt','w+');
if(!$ptr)
{print("erreur ouverture fichier"); exit; }
print("Ouverture ok ptr = " . $ptr . "<br>");
//écriture
for($i=1;$i<100;$i++)
    fputs($ptr, $i);
fclose($ptr);
//lecture
$ptr = fopen('tata.txt','r');
while(!feof($ptr))
{$c = fgets($ptr,20);
 print($c . "<br>");
}
fclose($ptr);
?>
```



# VIII. Fichiers

## □ 8.4. Gestion des fichiers et des répertoires

### ■ Modification

booleen chgrp("nomfich","nomgroupe")	change le groupe d'appartenance
booleen chmod("nomfich",mode)	change les droits d'accès
booleen chown("nomfich","owner")	change le propriétaire.
int rename("nomfich", "newnomfich")	change le nom du fichier.

### ■ Informations

chaine basename(chemin)	retourne le nom de fichier d'un chemin
entier filectime("nomfich")	temps écoulé depuis la dernière modification du fichier.
entier filesize("nomfich")	taille en octets du fichier.
booleen file_exists("nomfich")	1 si le fichier existe.
booleen is_file("nomfich")	1 si nomfich est un fichier.
booleen is_executable("nomfich")	1 si fichier existe et est en mode exécutable.
booleen is_readable("nomfich")	1 si fichier existe et est en mode lecture.
booleen is_writeable("nomfich")	1 si fichier existe et est en mode écriture.
chaine tempnam(rep, ch)	création d'un fichier temporaire unique dans le répertoire rep. Le préfixe est optionnel, retourne le nom du fichier temporaire.

# VIII. Fichiers

---

## ■ Répertoire

chaine dirname(chemin)	retourne le répertoire d'un chemin
booleen is_dir("nom")	1 si nom est un répertoire.
booleen chdir("repertoire")	change de repertoire, renvoie 1 si ok.
closedir(pointeur)	ferme un répertoire ouvert par opendir
booleen copy(fichsrce, repdest)	copie d'un fichier vers un répertoire.
booleen mkdir("nomrep")	crée un nouveau répertoire
entier opendir("nomrep")	ouvre un répertoire.
booleen rmdir("nomrep")	supprime le répertoire.
chaine readdir(pointeur)	pour lire un élément du répertoire (retourne un nom de fichier ou de répertoire).
rewinddir(pointeur)	revenir au début du répertoire.

- Exemple : affichage du contenu du répertoire courant
- **repCourant.php**

```
<?php
$REP = opendir(".");
while($entree = readdir($REP))
    echo "$entree<BR>";
closedir($REP);
?>
```

# VIII. Fichiers

## □ 8.5. Commandes du shell

chaine exec(commande, sortie, retour)	exécute la commande, la dernière ligne du résultat de la commande est retournée, n'envoie rien au navigateur. Argument sortie optionnel, s'il y est, chaque ligne du résultat est ajoutée à sortie comme un élément de tableau.
chaine getenv(variable_env)	retourne la valeur de la variable d'environnement.
putenv(variable_env)	pour paramétriser une variable d'environnement.
chaine system(commande, retour)	exécute la commande passée, envoie le résultat vers le navigateur, renvoie la dernière ligne de ce résultat.

### ■ Exemple : cmdSysteme.php

#### □ Contenu de la variable d'environnement PATH

```
<?php  
$chaine = getenv("path");  
echo "Variable Path = " . $chaine;  
?>
```

# VIII. Fichiers

---

## ■ TP : Crédation d'un fichier d'élèves

### ■ Interface d'ajout

The screenshot shows a web browser window with the title bar 'Konan Marcellin'. The main content area has a header 'Ajout d'un élève' and a URL bar showing 'localhost:82/tpFichier/formAjoi'. Below the URL bar, there are fields for 'Nom' (with an empty input field) and 'Age' (with an empty input field). At the bottom are two buttons: 'Ajouter' and 'Annuler'. The 'Age' input field is highlighted with a blue border.

### ■ Affichage des données dans un tableau HTML

# IX. MySQL et PHP

---

## □ 9.1. Présentation

- PHP permet d'accéder au données d'une BD :
  - Sans ODBC avec MySQL
  - avec ODBC avec les autres BD
- Les fonctions mysql\_\*
  - Devenue vieilles et ne supportent pas certains concepts des BD : requêtes préparées, procédures stockées, transactions...
- PDO : PHP Data Object
  - Classe PHP permettant à PHP de communiquer avec un serveur de données.
  - Couche d'abstraction orientée objet permettant de communiquer avec n'importe quel serveur de BD : MySQL, Oracle, PostgreSQL, etc...
  - Permet de sécuriser les requêtes et de favoriser la réutilisation du code grâce aux requêtes préparées.

# IX. PHP et MySQL

---

## □ 9.2. Fonctions de connexion au serveur MySQL

### ■ Constructeur PDO()

- Permet d'établir une connexion avec MySQL, pour un compte utilisateur, et un serveur donné.
- Retourne une valeur qui peut être utilisé pour dialoguer avec le serveur.
- Syntaxe
  - **\$idCnx = PDO(\$DSN, \$nomUtilisateur, \$motDePasse);**
- En cas d'erreur, une exception est levée
- Exemple

```
$DSN = 'mysql:host=localhost;dbname='bdgsf';charset=utf8';
$idCnx = new PDO($DSN, 'root', 'bkm');
```

# IX. PHP et MySQL

---

## ■ Erreurs levées par PDO

- PDO étant orienté objet, il lève des exceptions en cas de problème.
- Les erreurs levée sont des PDOException
- On utilise try...catch
- Erreurs rencontrées à l'instanciation d'un objet PDO.

Erreur	Explication
"could not find driver"	Le moteur de base de données est mal renseigné dans la DNS ou si le driver choisi n'est pas supporté par votre serveur.
"Unknown MySQL server host"	Le nom du serveur est mal renseigné ou indisponible (MySQL dans notre cas).
"Can't connect to MySQL server"	Dans le cadre des accès distants (le serveur MySQL n'est pas sur la même machine que le serveur Web), ce message indique que le serveur est planté ou indisponible, ou si le serveur n'est pas un serveur de données MySQL.

# IX. PHP et MySQL

## ■ Exemple : connexion.php

```
<?php
$nomUtilisateur = "root";
$motDePasse = "";
$serveur = "localhost";
$BD = "bdgsf";
$DSN = 'mysql:host=' . $serveur . ';dbname=' . $BD . ';charset=utf8';
try {
    $idCnx = new PDO( $DSN, $nomUtilisateur, $motDePasse );
}
catch ( Exception $e ) {
    echo "Connection à MySQL impossible : ", $e->getMessage();
    die();
}
echo 'Connexion au serveur ' . $serveur . ' et à la BD ' . $BD . ' réussie.';
?>
```

# IX. PHP et MySQL

---

## ■ Exécuter des requêtes

- Plusieurs façon d'exécuter des requêtes avec PDO.
- Deux types de requêtes :
  - Requêtes de sélection des enregistrements : **SELECT**
  - Requêtes de mise à jour les enregistrements : **UPDATE, INSERT, DELETE**

# IX. PHP et MySQL

---

## □ 9.3. Exécuter des requêtes SELECT

### ■ Méthode query()

- Permet d'exécuter un ordre SELECT
- Syntaxe
  - **\$resultat = \$idCnx->query(\$sql);**
- La variable **\$resultat** contient le résultat de la requête, sous la forme d'un **PDOStatement**.
- C'est un tableau associatif de clé les colonnes
- On peut parcourir les ligne avec une boucle foreach
- Exemple

```
$resultat = $idCnx->query('SELECT * FROM Client');
foreach ($resultat as $row){
    echo $row['numcl'] . ' ' . $row['nom'] . '<br>';
}
```

# IX. PHP et MySQL

---

## ■ Méthodes rowCount() et columnCount()

- rowCount() permet d'avoir le nombre de lignes résultat
- columnCount() permet d'avoir le nombre de colonne d'une lignes résultat
- Exemples
  - `$nbClient = $resultat->rowCount();`
  - `$nbColonne = $resultat->columnCount();`

## ■ Méthodes getColumnMeta()

- Permet de récupérer le nom d'une colonne dont l'index est passé en paramètre
- Syntaxe
  - `public array PDOStatement::getColumnMeta (int $column);`
- Exemple : nom de la 1<sup>ère</sup> colonne
  - `$meta = $resultat->getColumnMeta(0);`
  - `$nomColonne = $meta['name'];`

# IX. PHP et MySQL

---

## ■ Paramètre possibles du résultat \$meta

Name	Value
native_type	Le type PHP utilisé pour représenter la valeur de la colonne.
driver:decl_type	Le type SQL utilisé pour représenter la valeur de la colonne dans la BD.
flags	Tout flag positionné pour cette colonne.
name	Le nom de la colonne retournée par la BD.
table	Le nom de la table retournée par la BD.
len	La taille de la colonne. -1 pour les types autres que les nombre réels.
precision	La précision de la colonne. 0 pour les types autres que les nombre réels.
pdo_type	Le type de la colonne représenté par PDO::PARAM_* constants.

# IX. PHP et MySQL

## ■ Exemples : selectEx1.php

```
<?php
$nomUtilisateur = "root";
$motDePasse = "";
$serveur = "localhost";
$BD = "bdgsf";
$DSN = 'mysql:host=' . $serveur . ';dbname=' . $BD . ';charset=utf8';
try {
    $idCnx = new PDO( $DSN, $nomUtilisateur, $motDePasse );
}
catch ( Exception $e ) {
    echo "Connection à MySQL impossible : ", $e->getMessage();
    die();
}
$resultat = $idCnx->query('SELECT * FROM Client');
$nbClient = $resultat->rowCount();
$nbColonne = $resultat->columnCount();
echo 'Nombre de ligne : ' . $nbClient . '<br>';
echo 'Nombre de colonne : ' . $nbColonne . '<br>';
foreach ($resultat as $row){
    echo $row['numcl'] . ' ' . $row['nom'] . '<br>';
}
?>
```

# IX. PHP et MySQL

## ■ Méthode **fetch()**

- Permet de traiter le résultat renvoyé par la méthode **query()**
- **Syntaxe**

```
while($rows = $resultat->fetch($parametre)) {  
    //traitement  
}
```

- PDO permet de traiter la ligne résultat à l'aide de plusieurs format spécifiés par **\$paramètre** : tableaux, objets typés, etc...

Paramètre	Explication
PDO::FETCH_ASSOC	Tableau associatif avec les noms des colonnes comme clé
PDO::FETCH_NUM	Tableau d'indice numérique.
PDO::FETCH_BOTH	Combine PDO::FETCH_ASSOC et PDO::FETCH_NUM (par défaut)
PDO::FETCH_OBJ	Collection d'objet
PDO::FETCH_CLASS	

# IX. PHP et MySQL

---

## □ Remarque

- Si la réponse ne contiens pas de résultats, `$select->fetch()` retourne NULL/FALSE.
- Pas besoin d'utiliser une boucle si le select renvoie une seule ligne.

# IX. PHP et MySQL

- Exemple : **fetch(PDO::FETCH\_OBJ)**
  - **selectEx2.php**

```
<?php
$nomUtilisateur = "root";
$motDePasse = "";
$serveur = "localhost";
$BD = "bdgsf";
$DSN = 'mysql:host=' . $serveur . ';dbname=' . $BD . ';charset=utf8';
try {
    $idCnx = new PDO( $DSN, $nomUtilisateur, $motDePasse );
}
catch ( Exception $e ) {
    echo "Connection à MySQL impossible : ", $e->getMessage();
    die();
}
$resultat = $idCnx->query('SELECT * FROM Client');
while ($rows = $resultat->fetch(PDO::FETCH_OBJ)) {
    echo $rows->numcl . ' ' . $rows->nom . '<br>';
}
?>
```

# IX. PHP et MySQL

- Exemple : **fetch(PDO::FETCH\_ASSOC)**
  - **selectEx3.php**

```
<?php
$nomUtilisateur = "root";
$motDePasse = "";
$serveur = "localhost";
$BD = "bdgsf";
$DSN = 'mysql:host=' . $serveur . ';dbname=' . $BD . ';charset=utf8';
try {
    $idCnx = new PDO( $DSN, $nomUtilisateur, $motDePasse );
}
catch ( Exception $e ) {
    echo "Connection à MySQL impossible : ", $e->getMessage();
    die();
}
$resultat = $idCnx->query('SELECT * FROM Client');
while ($rows = $resultat->fetch(PDO::FETCH_ASSOC)){
    echo $rows['numcl'] . ' ' . $rows['nom'] . '<br>';
}
?>
```

# IX. PHP et MySQL

- Exemple : **fetch(PDO::FETCH\_NUM)**
  - **selectEx4.php**

```
<?php
$nomUtilisateur = "root";
$motDePasse = "";
$serveur = "localhost";
$BD = "bdgsf";
$DSN = 'mysql:host=' . $serveur . ';dbname=' . $BD . ';charset=utf8';
try {
    $idCnx = new PDO( $DSN, $nomUtilisateur, $motDePasse );
}
catch ( Exception $e ) {
    echo "Connection à MySQL impossible : ", $e->getMessage();
    die();
}
$resultat = $idCnx->query('SELECT * FROM Client');
while($rows = $resultat->fetch(PDO::FETCH_NUM)){
    echo $rows[0] . ' - ' . $rows[1] . '<br>';
}
?>
```

# IX. PHP et MySQL

---

## ■ Méthode **fetchALL()**

- Convertit l'objet résultats en un tableau d'objet,
- le tableau peut être traité comme un tableau classique
- Syntaxe

```
$rows = $resultat->fetchAll(PDO::FETCH_NUM);
```

# IX. PHP et MySQL

## ■ Exemple : selectEx5.php

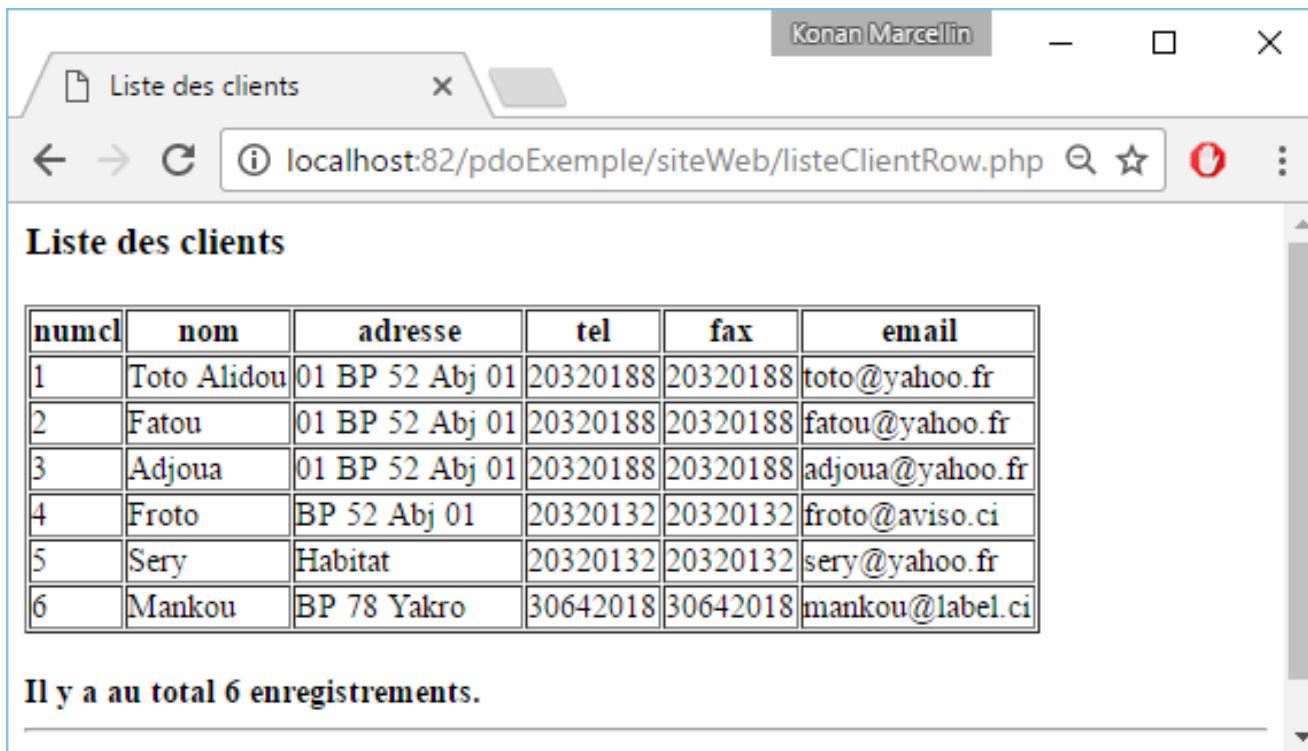
```
<?php
$nomUtilisateur = "root";
$motDePasse = "";
$serveur = "localhost";
$BD = "bdgsf";
$DSN = 'mysql:host=' . $serveur . ';dbname=' . $BD . ';charset=utf8';
try {
    $idCnx = new PDO( $DSN, $nomUtilisateur, $motDePasse );
}
catch ( Exception $e ) {
    echo "Connection à MySQL impossible : ", $e->getMessage();
    die();
}

$resultat = $idCnx->query('SELECT * FROM Client');
$rows = $resultat->fetchAll(PDO::FETCH_NUM);
$nbCol = $resultat->columnCount();
foreach ($rows as $rs){
    for ($i = 0; $i < $nbCol; $i++){
        echo $rs[$i] . ' ';
    }
    echo '<br>';
}
?>
```

# IX. PHP et MySQL

## Exercice

- Créer un tableau HTML pour afficher le résultat de l'exécution de la requête : "SELECT \* FROM Client;" comme le montre l'image écran ci-dessous.



The screenshot shows a web browser window titled "Konan Marcellin". The address bar displays "localhost:82/pdoExemple/siteWeb/listeClientRow.php". The main content area is titled "Liste des clients" and contains a table with 6 rows of client data. The table has columns: numcl, nom, adresse, tel, fax, and email. The data is as follows:

numcl	nom	adresse	tel	fax	email
1	Toto Alidou	01 BP 52 Abj 01	20320188	20320188	toto@yahoo.fr
2	Fatou	01 BP 52 Abj 01	20320188	20320188	fatou@yahoo.fr
3	Adjoua	01 BP 52 Abj 01	20320188	20320188	adjoua@yahoo.fr
4	Froto	BP 52 Abj 01	20320132	20320132	froto@aviso.ci
5	Sery	Habitat	20320132	20320132	sery@yahoo.fr
6	Mankou	BP 78 Yakro	30642018	30642018	mankou@label.ci

Il y a au total 6 enregistrements.

# IX. PHP et MySQL

---

## □ 9.4. Exécuter des requêtes mise à jour

- Méthode **query** : pour faire des **SELECT**,
- Méthode **exec()** : pour faire des **UPDATE, INSERT, DELETE.**

- **Syntaxe**

- **Exemple**

- **\$res = \$idCnx->exec(\$sql);**
    - \$res contient le nombre de mise à jour effectué

# IX. PHP et MySQL

---

## ▫ Exemple : insert.sql

```
<?php
//instruction de connexion
$sql = "INSERT INTO Client VALUES (79,'Moussa','UGB',221, 221,
'mlo@ugb.sn')";
try {
    $res = $idCnx->exec($sql);
    echo 'Insertion réalisée : ' . $res . '<br>' . $sql;
}
catch ( Exception $e ) {
    echo "Insertion impossible : ", $e->getMessage();
    die();
}
?>
```

# IX. PHP et MySQL

---

## □ Exemple : update.sql

```
<?php
//instruction de connexion
$sql = "UPDATE Client SET nom = 'Moussa Lo' WHERE numcl=78";
try {
    $res = $idCnx->exec($sql);
    echo 'Mise à jour réalisée : ' . $res . '<br>' . $sql;
}
catch ( Exception $e ) {
    echo "Mise à jour impossible : ", $e->getMessage();
    die();
}
?>
```

# IX. PHP et MySQL

## ▫ Exemple : delete.sql

```
<?php
//instruction de connexion
$sql = "DELETE FROM Client WHERE numcl>=78";
try {
    $res = $idCnx->exec($sql);
    echo 'Suppression réalisée : ' . $res . '<br>' . $sql;
}
catch ( Exception $e ) {
    echo "Suppression impossible : ", $e->getMessage();
    die();
}
?>
```

# IX. PHP et MySQL

---

## □ 9.5. Les requêtes préparées

- Réutilisables et plus sûr.
- Deux avantages majeurs :
  - Chaque exécution est plus rapide en requête préparée qu'en envoyant la totalité de la requête à chaque fois ;
  - Les paramètres sont transmis à la requête sous forme binaire, ce qui évite tout risque d'injection SQL.
    - Les paramètres sont transmis à la requête préparée soit dans l'ordre, soit sous forme nommée.
    - Les paramètres nommés peuvent être envoyés dans n'importe quel ordre.
- Principe
  - La requête n'est pas exécutée directement
  - On définit un gabarit de requête avec des options.

# IX. PHP et MySQL

## ■ Attaques par injection SQL

□ **Une injection SQL est un type d'exploitation d'une faille de sécurité d'une application Web interagissant avec une BD, en injectant une requête SQL non prévue par le système et pouvant compromettre sa sécurité.**

### □ Principe

- Envoyer, en utilisant les formulaires à disposition des utilisateurs du site, des commandes SQL manipulées au serveur.

### □ Exemple :

- Utiliser un formulaire php de connexion pour accéder à une BD.

The image shows a simple HTML form for logging in. It consists of two text input fields. The first field is labeled "Login :" and the second is labeled "Password :". Both input fields have a red border around them. The entire form is contained within a light green rectangular area.

- La requête à exécuter en PHP :

```
$sql = "SELECT * FROM user WHERE login = '$login' AND password = '$password'";
mysql_query($sql);
```

- Les variables \$login et \$password contiennent respectivement le login et le password saisis

# IX. PHP et MySQL

## ■ Cas 1 : Attaque sur le login

- Un pirate saisit dans le formulaire ci-dessus :

Login : ' OR 1 = 1 ;//

Password : N'importe quoi

- La requête générée sera :

```
$sql = "SELECT * FROM user WHERE user = " OR 1=1 ; // ' AND password =  
'4e383a1918b432a9bb7702f086c56596e";
```

- // indique début d'un commentaire en PHP
- La requête devient alors

```
$sql = SELECT * FROM user WHERE user = " OR 1 = 1;
```

- Le nom de l'utilisateur est égal à la chaîne vide ou si  $1 = 1$  qui toujours vrai.
- Le pirate se retrouvera connecté à la BD, avec toutes les chances d'être le DBA !

# IX. PHP et MySQL

## □ Cas 2 : Attaque sur le password

- Un pirate désirant se connecter avec le login toto pourra saisir dans le formulaire ci-dessus :

Login :	toto
Password :	' OR 1 = 1 OR '

- La requête générée sera :

```
SELECT *
FROM users
WHERE user = 'toto'
AND password = " OR TRUE OR " ;
```

- Le mot de passe est égal à la chaîne vide ou  $1 = 1$  est toujours vrai ou la chaîne vide
- Le pirate se retrouvera connecté à la BD, avec toutes les chances d'être le DBA !

# IX. PHP et MySQL

---

## □ Comment éviter ces attaques

- Utiliser des procédures stockées, à la place du SQL dynamique. Les données entrées par l'utilisateur sont alors transmises comme paramètres, sans risque d'injection.
- Vérifier de manière précise et exhaustive l'ensemble des données venant de l'utilisateur. On peut, par exemple, utiliser une expression rationnelle afin de valider qu'une donnée entrée par l'utilisateur est bien de la forme souhaitée.
- Utiliser des comptes utilisateurs SQL à accès limité (en lecture-seule) quand cela est possible.
- Utiliser des requêtes SQL paramétrées (requêtes à trous envoyées au serveur SQL, serveur à qui l'on envoie par la suite les paramètres qui boucheront les trous), ainsi c'est le SGBD qui se charge d'échapper les caractères selon le type des paramètres.

# IX. PHP et MySQL

---

## ■ Solution

### □ Création de rôles et d'utilisateur

- Définition des différentes sortes d'actions
- Définition des droits des différents types d'utilisateurs de la BD

### □ Utilisation de vues

- Restriction et vérification dynamique pour la consultation comme pour la modification des données

### □ Utilisation de procédures stockées

- Contrôle des opérations effectuées
- Sécurisation des requêtes effectuées sur la base

# IX. PHP et MySQL

---

## ■ Paramètre mystère

- Le caractère **?** est utilisé pour transmettre des valeurs au moment de l'exécution de la requête préparée.

### □ Syntaxe

#### 1. Préparation de la requête

```
$sqlPrepare = $idCnx->prepare('SELECT * FROM Client  
WHERE numCl=?');
```

#### 2. Envoi de la requête

```
$sqlPrepare->execute(array(1));
```

#### 3. Traitement

```
$row = $sqlPrepare->fetch(PDO::FETCH_OBJ);  
echo $row->numcl, ' ', $row->nom;
```

- On peut réutiliser la requête préparée en modifiant la valeur utilisée par une autre lors de l'envoie de la requête (2).

# IX. PHP et MySQL

## ▫ Exemple 1 : un seul paramètre

- selectPrepareEx1.php

```
<?php
//instructions de connexion
$sqlPrepare = $idCnx->prepare('SELECT * FROM Client WHERE numCl=?');
try {
    // On envois la requète
$sqlPrepare->execute(array(1));
    // Traitement
    if( $row = $sqlPrepare->fetch(PDO::FETCH_OBJ)){
        echo $row->numcl, ' ', $row->nom;
    }
}
catch( Exception $e ){
    echo 'Erreur de selection: ', $e->getMessage();
}
?>
```

# IX. PHP et MySQL

## ▫ Exemple 2 : plusieurs paramètres

- selectPrepareEx2.php

```
<?php
//instructions de connexion
$sqlPrepare = $idCnx->prepare('SELECT * FROM Client
WHERE numCl>? AND nom LIKE ?');
try {
    // On envois la requète
    $sqlPrepare->execute(array(1, 'F%'));
    // Traitement
    while( $rows = $sqlPrepare->fetch(PDO::FETCH_OBJ)){
        echo $rows->numcl, ' ', $rows->nom, '<br>';
    }
}
catch( Exception $e ){
    echo 'Erreur de selection: ', $e->getMessage();
}??>
```

## ▫ L'ordre des paramètre effectifs doit être respecté

# IX. PHP et MySQL

---

## ■ Paramètre nommé

- Le caractère **?** est utilisé pour transmettre des valeurs au moment de l'exécution de la requête préparée.
- **Syntaxe**

### 1. Préparation de la requête

```
$sqlPrepare = $idCnx->prepare('SELECT * FROM Client  
WHERE numCl>:numClt AND nom LIKE :nomClt');
```

### 2. Envoi de la requête

```
$sqlPrepare->execute(array('numClt'=>1, 'nomClt'=>'F%'));  
Ou  
$sqlPrepare->execute(array('nomClt'=>'F%', 'numClt'=>1));
```

On peut ne pas respecter l'ordre des paramètres

### 3. Traitement

```
$row = $sqlPrepare->fetch(PDO::FETCH_OBJ);  
echo $row->numcl, ' ', $row->nom;
```

# IX. PHP et MySQL

- Requête préparée pour mise à jour
  - Même processus que pour un SELECT
  - **updatePrepareEx1.php**

```
<?php
//instructions de connexion
$sql = 'UPDATE Client SET nom = :nomClt WHERE numcl = :numClt';
$maj = $idCnx->prepare($sql);
try {
    $maj->execute(array('numClt'=>1, 'nomClt'=>'Toto Ali'));
    echo 'Modification réalisée : <br>' . $sql;
}
catch ( Exception $e ) {
    echo "Modification impossible : ", $e->getMessage();
    die();
}
?>
```

# IX. PHP et MySQL

## ■ Paramètres Bindés

- La méthode **bindParam()** permet de remplir la requête préparée avec le contenu d'une variable.
- on a peut aussi préciser un type de donnée et une taille.

### 1. Préparation de la requête

```
$sql = 'UPDATE Client SET nom = :nomClt WHERE numcl = :numClt';  
$maj = $idCnx->prepare($sql);
```

### 2. Bind

```
$maj->bindParam(':nomClt', $nom, PDO::PARAM_STR, 15);
```

- Autres types : PDO::PARAM\_INT, PDO::PARAM\_DATE

### 3. Envoi de la requête

```
$maj->execute();
```

### 4. Traitement

# IX. PHP et MySQL

## □ Exemple : updatePrepareEx2.php

```
<?php
$sql = 'UPDATE Client SET nom = :nomClt WHERE numcl = :numClt';
$maj = $idCnx->prepare($sql);
$nom = 'Toto Alidou';
$numcl = 1;
try {
    $maj->bindParam(':nomClt', $nom, PDO::PARAM_STR, 15);
    $maj->bindParam(':numClt', $numcl, PDO::PARAM_INT);
    $maj->execute();
    echo 'Modification réalisée : <br>' . $sql;
}
catch ( Exception $e ) {
    echo "Modification impossible : ", $e->getMessage();
    die();
}
?>
```

# IX. PHP et MySQL

---

## □ 9.5. Utilisation des sessions php

### ■ Session

- Permet de stocker des informations de l'utilisateur sur le serveur (identifiants de connexion).
  - A l'inverse des cookies qui stockent les informations sur le poste du client.
- Temporaire et effacé très rapidement du serveur.
  - Temps moyen par défaut = 30 minutes.
- Informations stockées dans des fichiers, sur le serveur
  - Un fichier par session.
- Chaque session possède un nom et un identifiant.
- Normalement, la session est détruite ou à la fermeture du navigateur, ou au bout de 30 minutes.
  - Cette valeur peut être modifiée dans le fichier php.ini du serveur.

### ■ Remarque

- Session déclarée dans la page tout en haut de la page web.
  - Le cache du navigateur doit être vide pour démarrer une session.
  - Aucun code HTML avant le démarrage d'une session.

# IX. PHP et MySQL

## ■ Fonctionnement d'une session

Fonction	Explication
Démarrer une session	<?php session_start(); ?>
Numéro de la session	<?php \$session_id(); ?>
Créer une variable de session	<?php \$_SESSION['variable']=\$valeur ;?>
Savoir si une variable de session existe	<?php if(isset(\$_SESSION['variable'])) echo 'La variable "variable" existe et vaut: ' . \$_SESSION['variable']; ?>
Supprimer une variable de session	<?php unset(\$_SESSION['variable']); echo 'La variable de session "variable" est détruite'; ?>
Détruire toutes les variables de session	<?php session_unset(); ?>
Détruire une session	<?php session_destroy(); ?>

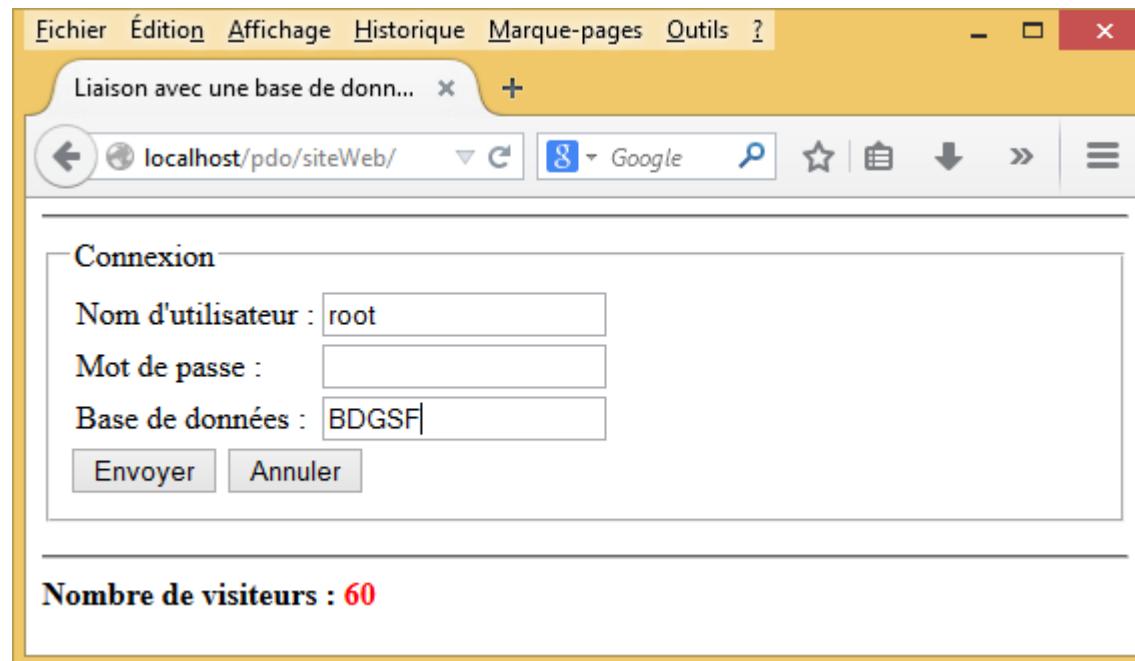
# IX. PHP et MySQL

## □ 9.5. Manipulation d'une table

### ■ Page de garde : index.php

#### □ Fenêtre de connexion

#### □ Source\PDO\siteWeb



# IX. PHP et MySQL

```
<html>
<head><title>Liaison avec une base de données</title></head>
<body>
<HR size=2>
<FORM NAME=frmRech ACTION=menu.php METHOD=POST>
<FIELDSET>
    <LEGEND>Connexion</LEGEND>
    <TABLE>
        <TR><TD><LABEL>Nom d'utilisateur :</LABEL></TD>
            <TD><INPUT TYPE=TEXT NAME="NU" VALUE=""></TD></TR>
        <TR><TD><LABEL>Mot de passe :</LABEL></TD>
            <TD><INPUT TYPE=PASSWORD NAME="MP" VALUE=""></TD>
        </TR><TR><TD><LABEL>Base de données :</LABEL></TD>
            <TD><INPUT TYPE=TEXT NAME="BD" VALUE=""></TD></TR>
    </TABLE>
    <Input Type=Submit Value=Envoyer NAME=submit>
    <Input Type=Reset Value=Annuler>
</FIELDSET>
</form>
</body>
<HR size=2>
</html>
```

# IX. PHP et MySQL

## ▢ Compteurs des visiteur

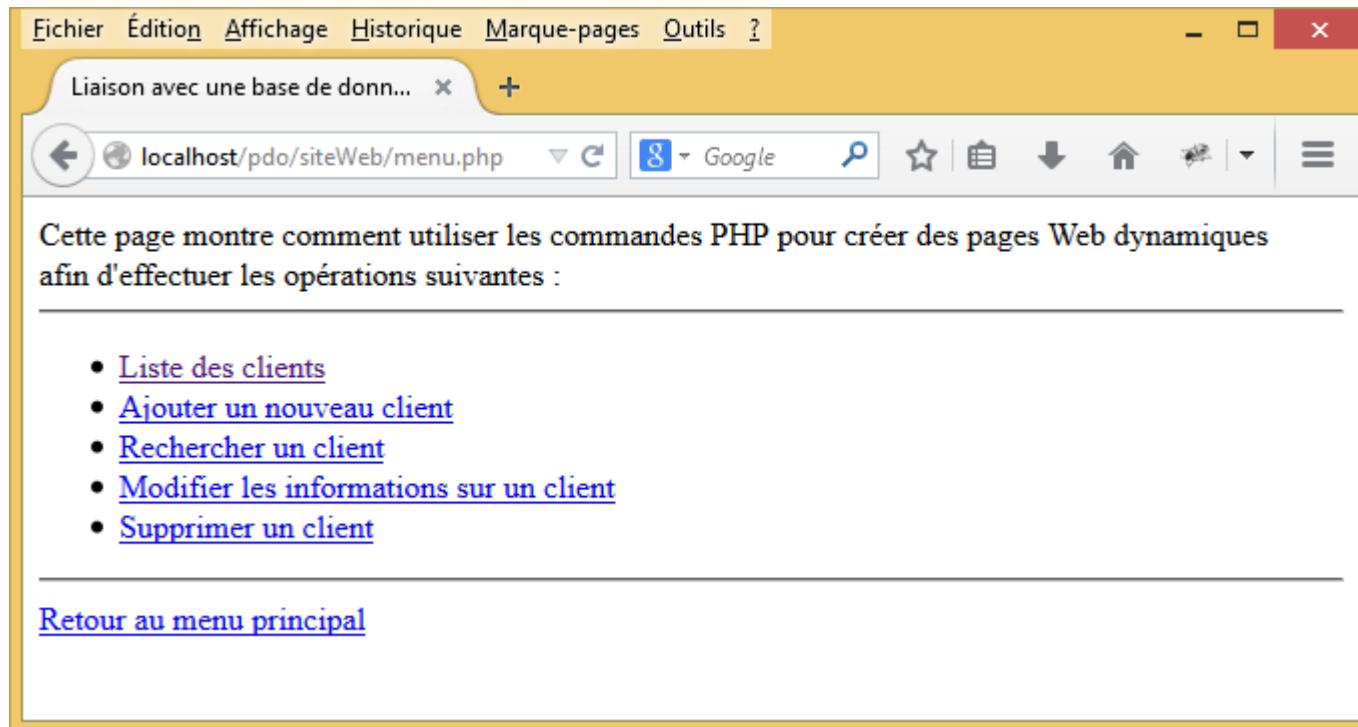
- en bas de la page de garde index.php

```
<html>
...
<b>Nombre de visiteurs : </b>
<?php include("codePhp.php");
echo "<b><font color=red>" . majCmpt("fichierCmpt.txt") . "</font></b>";
?>
</body>
</html>
```

```
function majCmpt($nomFichier)
{$ptr = fopen($nomFichier,'r');
 $cmpt = fgets($ptr,255);
 fclose($ptr);
 $ptr = fopen($nomFichier,'w');
 $cmpt = $cmpt + 1;
 fputs($ptr, $cmpt);
 fclose($ptr);
 return $cmpt;
}
```

# IX. PHP et MySQL

- **Page de menu : menu.htm**
  - **Propose tous les traitements sur une tables**



# IX. PHP et MySQL

- **Gestion de sessions**
  - Contient les paramètres d'accès à la BD

```
<?php
session_start();
//definition des variables d'une session
if(isset($_POST["submit"])) //Bouton Envoyer appuyé
{
$_SESSION['Serveur'] = 'localhost';
$_SESSION['Nom'] = $_POST["NU"];
$_SESSION['MPasse'] = $_POST["MP"];
$_SESSION['BD'] = $_POST["BD"];

global $serveur, $nom, $mpasse, $bd;
$serveur = $_SESSION['Serveur'];
$nom = $_SESSION['Nom'];
$mpasse = $_SESSION['MPasse'];
$bd = $_SESSION['BD'];
}
?>
```

# IX. PHP et MySQL

```
<html>
<head><title>Liaison avec une base de données MySQL</title></head>
<body>
<?php
include("codePhp.php");
if($_POST["submit"]) $idCnx = connexionBD($nom, $mpasse, $bd);
if($idCnx = 1)
{echo "Cette page montre comment utiliser les commandes PHP " ;
echo "pour créer des pages Web dynamiques <br>";
echo "afin d'effectuer les opérations suivantes :";
echo "<HR size=2>";
echo "<ul>";
echo " <li><a href='listeClientRow.php'>Liste des clients</a></li>";
echo " <li><a href='ajoutClient.htm'>Ajouter un nouveau client</a></li>";
echo " <li><a href='RechercherClient.htm'>Rechercher un client</a></li>";
echo " <li><a href='majClient.htm'>Modifier les informations sur un client</a></li>";
echo " <li><a href='supprimeClient.htm'>Supprimer un client</a></li>";
echo "</ul>";
}
else
die("Insuffisance de privilèges");
?> <HR size="2"><A href="index.html">Retour au menu principal</A>
</body>
</html>
```

# IX. PHP et MySQL

## ❑ Fichier auth.php

- ❑ Entête de chaque client qui doit se connecter à la BD

```
<?php
session_start() ; // initialisation
//tester la présence des variables de la session
if(isset($_SESSION['Serveur']) && isset($_SESSION['Nom']) &&
isset($_SESSION['MPasse']) && isset($_SESSION['BD']))
{ //lecture des variables de session
    global $serveur, $nom, $mpasse, $bd;
    $serveur = $_SESSION['Serveur'] ;
    $nom = $_SESSION['Nom'];
    $mpasse = $_SESSION['MPasse'];
    $bd = $_SESSION['BD'];
}
else
{die('Erreurs dans les paramètres de connexion');}
?>
```

# IX. PHP et MySQL

## ■ Lister une table

- **SELECT \* FROM Client**
- **listeClientRow.php et listeClientObject.php**

The screenshot shows a web browser window with a yellow header bar. The title bar contains menu items: Fichier, Édition, Affichage, Historique, Marque-pages, Outils, and a question mark icon. Below the title bar is a toolbar with various icons. The main content area displays a table titled "Liste des clients". The table has columns labeled: numcl, nom, adresse, tel, fax, and email. The data in the table is as follows:

numcl	nom	adresse	tel	fax	email
1	Toto Alidou	01 BP 52 Abj 01	20320188	20320188	toto@yahoo.fr
2	Fatou	01 BP 52 Abj 01	20320188	20320188	fatou@yahoo.fr
3	Adjoua	01 BP 52 Abj 01	20320188	20320188	adjoua@yahoo.fr
4	Froto	BP 52 Abj 01	20320132	20320132	froto@aviso.ci
5	Sery	Habitat	20320132	20320132	sery@yahoo.fr
6	Mankou	BP 78 Yakro	30642018	30642018	mankou@label.ci

Below the table, a message states: "Il y a au total 6 enregistrements." At the bottom of the page, there is a link: "Retour au menu principal".

# IX. PHP et MySQL

## □ Utilisation des indices de colonnes

```
function afficher($resultat) {
    $nbCol = $resultat->columnCount(); // nbre de colonne
    $nbClient = $resultat->rowCount(); // nbre de ligne retourné
    echo "<TABLE border='1'>";
    $tete="";
    for ($i = 0 ; $i < $nbCol; $i++) {
        $meta = $resultat->getColumnMeta($i);
        $field = $meta['name']; // nom de la colonne
        $tete = $tete . "<TH>" . $field . "</TH>"; }
    echo "<TR>" . $tete . "</TR>" ;
    $rows = $resultat->fetchAll(PDO::FETCH_NUM);
    foreach ($rows as $rs) {
        $tete="";
        for ($i = 0 ; $i < $nbCol; $i++) {
            if($rs[$i])
                $tete = $tete . "<TD>" . $rs[$i] . "</TD>";
            else
                $tete = $tete . "<TD>" . "&nbsp;" . "</TD>";}
        echo "<TR>" . $tete . "</TR>" ;
    }
    echo "</TABLE>";
    echo "<BR><B>Il y a au total " . $nbClient . " enregistrements.</B>";
}
```

# IX. PHP et MySQL

- Ajouter un enregistrement dans une table
  - **INSERT INTO VALUES(...)**
  - **ajoutClient.htm**

The screenshot shows a web browser window with a yellow header bar containing menu items: Fichier, Édition, Affichage, Historique, Marque-pages, Outils, and a question mark icon. Below the header is a toolbar with icons for back, forward, search, and other browser functions. The main content area has a title bar that says "Ajouter un nouveau client à la ...". The page content is as follows:

**Ajout d'un nouvel enregistrement à la table client.**

Saisissez les informations sur le client et cliquez sur le bouton Envoyer.

numCl:   
nom:   
adresse:   
tel:   
fax:   
email:

[Retour au menu principal](#)

# IX. PHP et MySQL

```
<html>
<head><title>Ajouter un nouveau client à la BD</title></head>
<body>
<H3>Ajout d'un nouvel enregistrement à la table client.</H3>
<FORM NAME=ajout ACTION=traiterAjoutClient.php METHOD=POST>
<H4>Saisissez les informations sur le client et cliquez sur le bouton Envoyer.</H4>
<table>
<tr> <td>numCl: </td> <td><Input Type=Text Name=Ch1></td> </tr>
<tr> <td>nom: </td> <td><Input Type=Text Name=Ch2></td> </tr>
<tr> <td>adresse: </td> <td><Input Type=Text Name=Ch3></td> </tr>
<tr> <td>tel: </td><td><Input Type=Text Name=Ch4></td> </tr>
<tr> <td>fax: </td> <td><Input Type=Text Name=Ch5></td> </tr>
<tr> <td>email: </td> <td><Input Type=Text Name=Ch6></td> </tr>
</table>
<br>
<Input Type=Submit Value=Envoyer>
<Input Type=Reset Value=Annuler>
</FORM>
<HR size="2"><A href="index.html">Retour au menu principal</A>
</body>
</html>
```

# IX. PHP et MySQL

## ▣ traiterAjoutClient.php

```
<?php
include("codePhp.php");
$idCnx = connexionBD($serveur, $nom, $mpasse, $bd);

$sql = "INSERT INTO Client VALUES(" . $_POST["Ch1"] . ",";;
$sql .= $_POST["Ch2"] . "" . "," . $_POST["Ch3"] . "" . "";
$sql .= $_POST["Ch4"] . "" . $_POST["Ch5"] . "" . $_POST["Ch6"] . ")";
try {
    $res = $idCnx->exec($sql);
    echo 'Insertion réalisée : <br>' . $sql;
}
catch ( Exception $e ) {
    echo "Insertion impossible : ", $e->getMessage();
    die();
}
?>
```

- ▣ **`$_POST["Ch1"]`** : permet de récupérer le contenu de la zone de saisie dont le nom est Ch1.

# IX. PHP et MySQL

## ■ Exécution

The screenshot shows a web application interface for managing clients. The main window title is "Ajout d'un client". The URL in the address bar is "localhost/pdo/siteWeb/traiterAjoutClient.php". The page content includes a success message: "Insertion réalisée : INSERT INTO Client VALUES(80,'Aicha','Abidjan','22','22','aicha@yahoo.ci')". Below this, there is a link "Retour au menu principal". On the left, a sidebar titled "Liste des clients" shows a table with 7 rows of client data. At the bottom, a message states "Il y a au total 7 enregistrements." and a link "Retour au menu principal".

numcl	nom	adresse	tel	fax	email
1	Toto Alidou	01 BP 52 Abj 01	20320188	20320188	toto@yahoo.fr
2	Fatou	01 BP 52 Abj 01	20320188	20320188	fatou@yahoo.fr
3	Adjoua	01 BP 52 Abj 01	20320188	20320188	adjoua@yahoo.fr
4	Froto	BP 52 Abj 01	20320132	20320132	froto@aviso.ci
5	Sery	Habitat	20320132	20320132	sery@yahoo.fr
6	Mankou	BP 78 Yakro	30642018	30642018	mankou@label.ci
80	Aicha	Abidjan	22	22	aicha@yahoo.ci

Il y a au total 7 enregistrements.

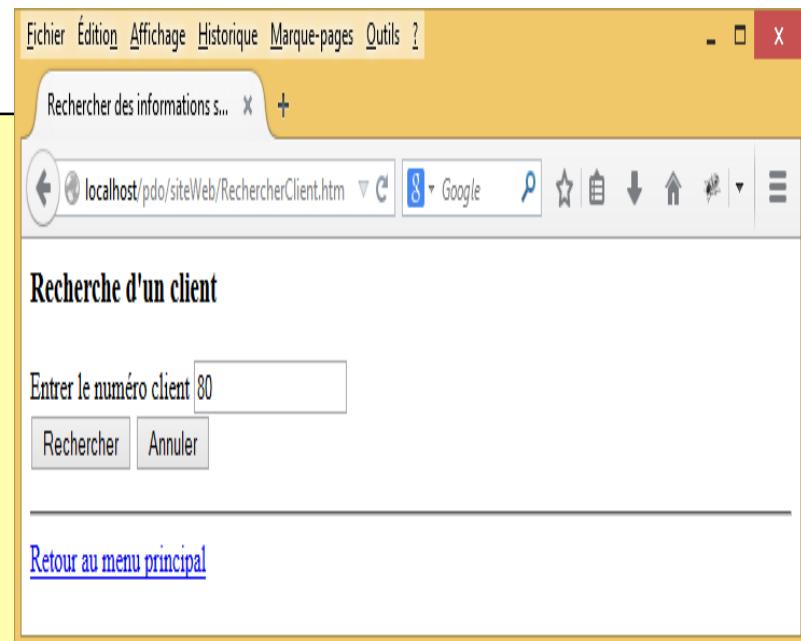
[Retour au menu principal](#)

# IX. PHP et MySQL

## ■ Rechercher un enregistrement d'une table

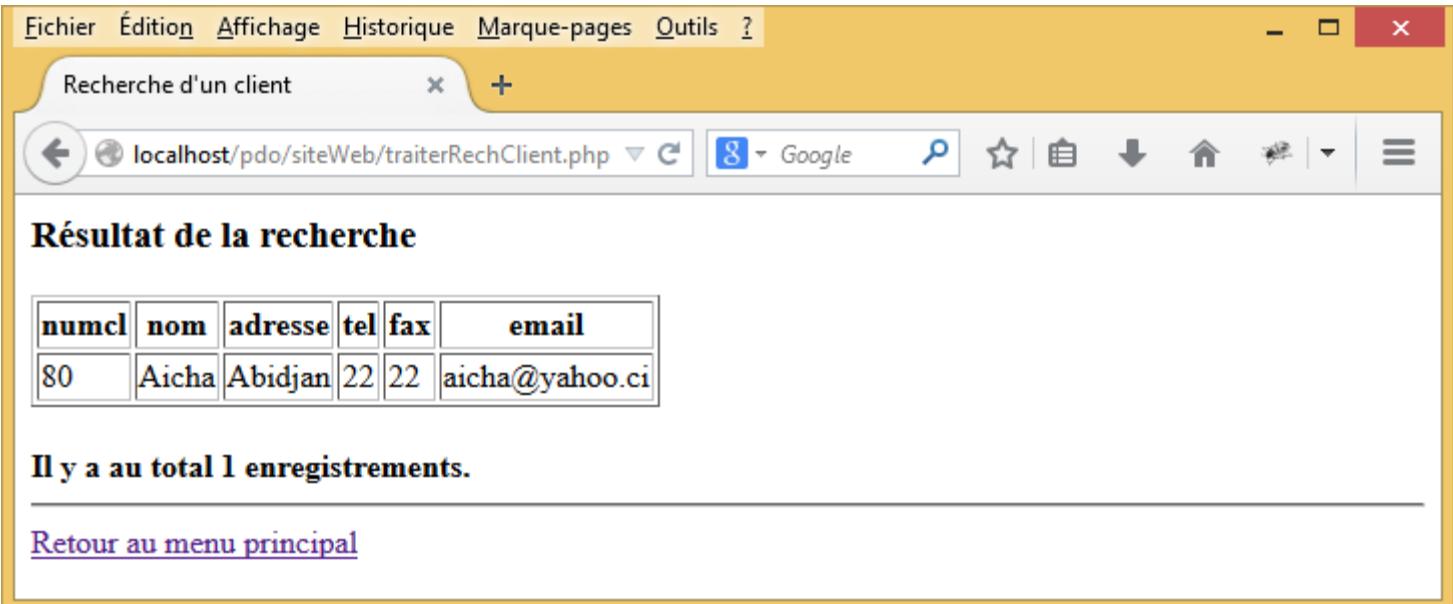
- **SELECT \* FROM Client WHERE numCl = " . \$\_POST["Ch1"]**
  - Requête paramétrée
- **rechercheClient.htm**

```
<html>
<head><title>Rechercher des informations sur un
client</title></head>
<body>
<H3>Recherche d'un client</H3>
<FORM NAME=frmRech
ACTION=traiterRechClient.php METHOD=POST>
Entrer le numéro client <INPUT TYPE=TEXT
NAME="Ch1" VALUE=""><br>
<Input Type=Submit Value=Rechercher>
<Input Type=Reset Value=Annuler>
</FORM>
<HR size="2"><A href="index.html">Retour au menu
principal</A>
</body>
</html>
```



# IX. PHP et MySQL

## ■ traiterRechClient.php



The screenshot shows a web browser window with the following details:

- Menu Bar:** Fichier, Édition, Affichage, Historique, Marque-pages, Outils, ?
- Title Bar:** Recherche d'un client
- Address Bar:** localhost/pdo/siteWeb/traiterRechClient.php
- Toolbar:** Back, Forward, Stop, Refresh, Google search bar, Favorites, Home, Downloads, etc.
- Content Area:**
  - Section Header:** Résultat de la recherche
  - Table:** A table with columns: numcl, nom, adresse, tel, fax, email. The data row is: 80, Aicha, Abidjan, 22, 22, aicha@yahoo.ci
  - Text:** Il y a au total 1 enregistrements.
  - Link:** Retour au menu principal

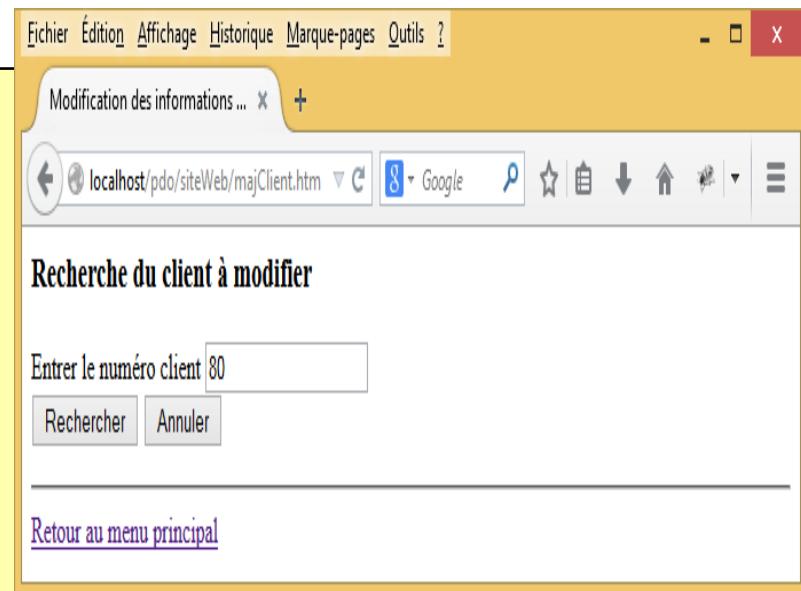
```
$sql = "SELECT * FROM Client WHERE numCl = " . $_POST["Ch1"] . ";" ;
$resultat = mysql_query($sql, $connexion);
if($resultat)
{...}
```

# IX. PHP et MySQL

## ■ Modifier un enregistrement

- **UPDATE Client SET ... WHERE numcl= " . \$\_POST["Ch1"]**
- **majClient.htm**

```
<html>
<head><title>Modification des informations sur un
client</title></head>
<body>
<H3>Recherche du client à modifier</H3>
<FORM NAME=frmMaj
ACTION=rechercherClientMaj.php METHOD=POST>
    Entrer le numéro client <INPUT TYPE=TEXTBOX
NAME="Ch1" VALUE=""><br>
    <Input Type=Submit Value=Rechercher>
    <Input Type=Reset Value=Annuler>
</FORM>
<HR size="2"><A href="index.html">Retour au menu
principal</A>
</body>
</html>
```



- **Il faut rechercher l'enregistrement à modifier**

# IX. PHP et MySQL

## ▫ rechercherClientMaj.php

```
<?php
include("codePhp.php");
$idCnx = connexionBD($serveur, $nom, $mpasse, $bd);
$sql = "SELECT * FROM Client WHERE numcl = " . $_POST["Ch1"] . ";" ;
$resultat = $idCnx->query($sql);
if($resultat){
    $nbCol = $resultat->columnCount();
    $nbClient = $resultat->rowCount();
    echo "<TABLE border='1'>";
    $tete="";
    echo "<TABLE BORDER=1>";
    $rows = $resultat->fetchAll(PDO::FETCH_NUM);
    foreach ($rows as $row) {
        for ($i = 0 ; $i < $nbCol; $i++) {
            $meta = $resultat->getColumnMeta($i);
            $field = $meta['name'];
            $data = "<INPUT TYPE='Text' NAME='Ch" . ($i+1) . "' VALUE = '" . $row[$i] . "'>";
            $tete = "<TH>" . $field . "</TH><TD>" . $data . "</TD>";
            echo "<TR>" . $tete . "</TR>" ;
        }
    }
    echo "</table>";
}
else {
    echo "Erreur dans l'exécution de
la requête<br>";
}
?>
```

# IX. PHP et MySQL

## ■ Exécution

The screenshot shows a web browser window with the following details:

- Menu Bar:** Fichier, Édition, Affichage, Historique, Marque-pages, Outils, ?
- Title Bar:** Recherche d'un client
- Address Bar:** localhost/pdo/siteWeb/rechercherClientMaj.php
- Toolbar:** Back, Forward, Stop, Refresh, Home, etc.
- Content Area:**
  - Section Header:** Résultat de la recherche
  - Text:** Faites les modifications et cliquer sur le bouton Modifier.
  - Form:** A table with input fields for modifying a client record.

<b>numcl</b>	80
<b>nom</b>	Aicha
<b>adresse</b>	Abidjan
<b>tel</b>	22
<b>fax</b>	22
<b>email</b>	aicha@yahoo.ci
  - Buttons:** Modifier, Annuler
  - Link:** Retour au menu principal

# IX. PHP et MySQL

## ■ **traiterMajClient.php**

```
<?php
include("codePhp.php");
$idCnx = connexionBD($serveur, $nom, $mpasse, $bd);
$sql = "UPDATE Client SET numcl=" . $_POST["Ch1"] . ", nom=\"" . $_POST["Ch2"] .
",adresse=\"";
$sql= $sql . $_POST["Ch3"] . "",tel=" . $_POST["Ch4"] . ",fax=" . $_POST["Ch5"] . ",
email=\"";
$sql= $sql . $_POST["Ch6"] . "" WHERE numcl=" . $_POST["Ch1"] ;
try {
$res = $idCnx->exec($sql);
echo 'Mise à jour réalisée : ' . $res . '<br>' . $sql;
}
catch ( Exception $e ) {
echo "Mise à jour impossible : ", $e->getMessage();
die();
}
?>
```

# IX. PHP et MySQL

## ■ Exécution

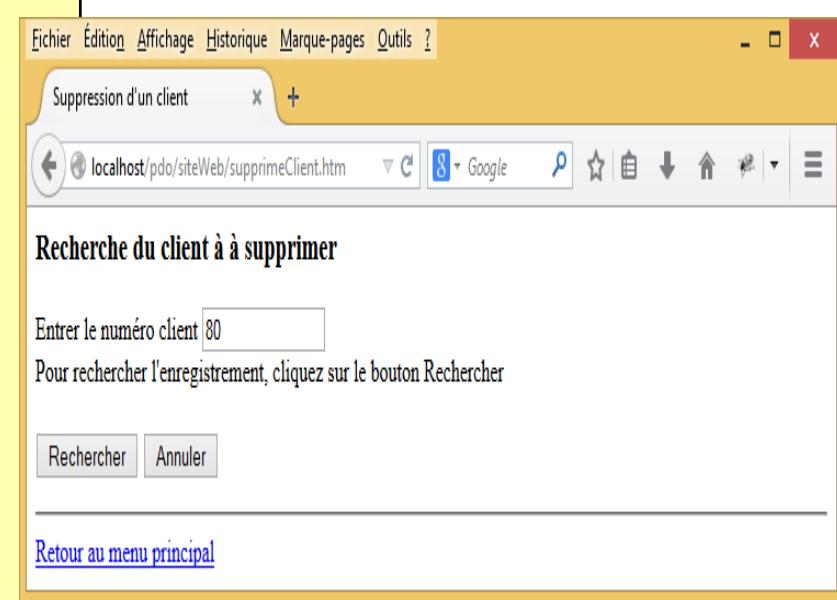


# IX. PHP et MySQL

## ■ Supprimer un enregistrement d'une table

- ▣ **\$sql = "DELETE FROM Client WHERE numCl=" . \$\_POST["Ch1"] . ";" ;**
- ▣ Requête paramétrée
- ▣ **supprimeClient.htm**

```
<html>
<head><title>Suppression d'un
client</title></head>
<body>
<H3>Recherche du client à supprimer</H3>
<FORM NAME=frmMaj
ACTION=rechercherClientSuppr.php
METHOD=POST>
    Entrer le numéro client <INPUT TYPE=TEXTBOX
NAME="Ch1" VALUE="" Size=15><br>
    Pour rechercher l'enregistrement, cliquez sur le
bouton Rechercher<br><br>
    <Input Type=Submit Value=Rechercher>
    <Input Type=Reset Value=Annuler>
</FORM>
</body>
</html>
```



# IX. PHP et MySQL

## □ rechercherClientSuppr.php

```
<FORM NAME=frmRech ACTION=traiterSupprClient.php  
METHOD=POST>  
$sql = "SELECT * FROM Client WHERE numcl = " . $_POST  
";";  
$resultat = mysql_query($sql, $connexion);  
if($resultat)  
{$nbCol = mysql_num_fields ($resultat );  
$nbClient = mysql_affected_rows();  
echo "<TABLE border='1'>";  
$tete="";  
echo "<TABLE BORDER=1>";  
if($row = mysql_fetch_row($resultat))  
{for ($i = 0 ; $i < $nbCol; $i++)  
{$field = mysql_field_name($resultat, $i);  
$data = "<INPUT TYPE='Text' NAME='Ch" . ($i+1) .  
"' . $row[$i] . "'>";  
$tete = "<TH>" . $field . "</TH><TD>" . $data . "</TD>";  
echo "<TR>" . $tete . "</TR>" ;  
}  
}  
}  
} ...  
</FORM>
```

Recherche d'un client

Résultat de la recherche

Cliquer sur le bouton Supprimer.

numcl	80
nom	Aicha Diallo
adresse	Abidjan
tel	22
fax	22
email	aicha@yahoo.ci

Supprimer Annuler

Retour au menu principal

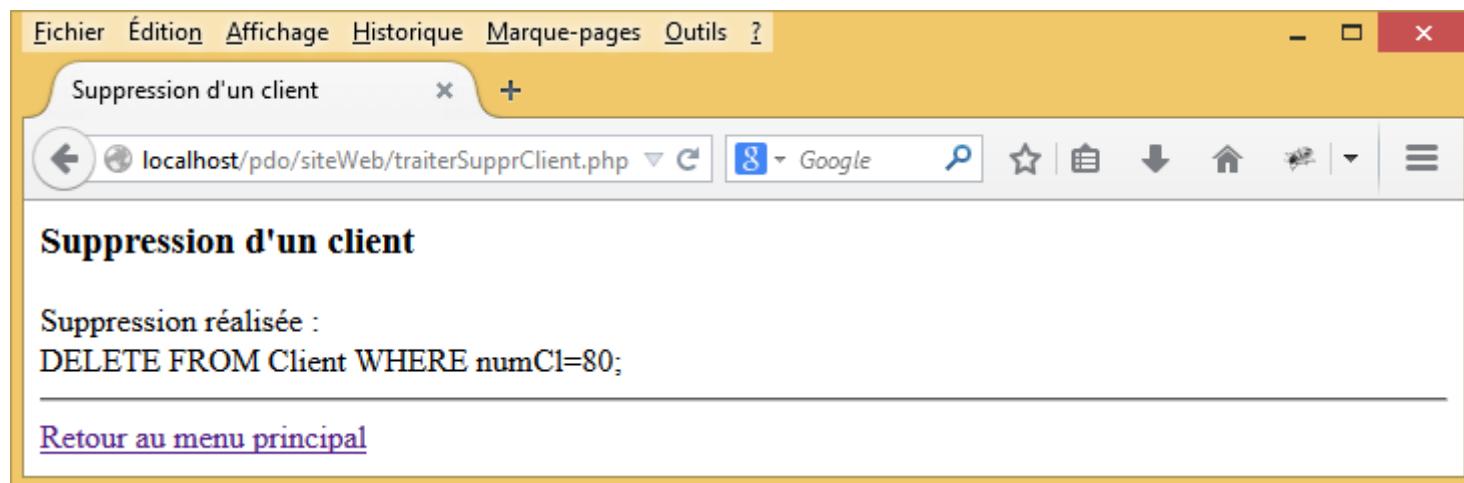
# IX. PHP et MySQL

## ❑ traterSupprClient.php

```
<?php
include("codePhp.php");
$idCnx = connexionBD($serveur, $nom, $mpasse, $bd);
$sql = "DELETE FROM Client WHERE numCl=" . $_POST["Ch1"] . ";" ;
try {
    $res = $idCnx->exec($sql);
    echo 'Suppression réalisée : <br>' . $sql;
}
catch ( Exception $e ) {
    echo "Mise à jour impossible : ", $e->getMessage();
    die();
}
?>
```

# IX. PHP et MySQL

## ■ Exécution



# IX. PHP et MySQL

---

## □ 9.6. Accès à la BD via un autre poste

1. Ouvrir le fichier " C:\wamp\apache2\conf\httpd.conf de la machine serveur.
2. Modifier les champs comme suit :
  - Toutes les adresse IP "127.0.0.1" par l'adresse IP de la machine serveur.
  - Tous les noms par défaut "localhost" par le nom de la machine serveur.

Champ initial	Champ modifié
Listen 127.0.0.1:80	Listen 172.16.36.168:80
ServerAdmin admin@localhost	ServerAdmin admin@broukm
ServerName localhost	ServerName broukm
Toutes les adresses IP 127.0.0.1	172.16.36.168:80

# IX. PHP et MySQL

## 4. Faire de même pour le fichier

" C:\Program Files\EasyPHP 3.0\phpmyadmin\config.inc.php"

- NB . Dupliquer config.sample.php en config.inc.php

Champ initial	Champ modifié
\$cfg['Servers'][\$i]['host'] = 'localhost';	\$cfg['Servers'][\$i]['host'] = 'broukm';

## 5. Saisir l'adresse sur le même poste ou sur un autre poste connecté en réseau :

<http://kmbrou/accessphp/webDyn.htm>

# IX. PHP et MySQL

## ■ TP : /Source/TDetTP/client.php

- Créer une application permettant de gérer les commandes des clients comme le montre l'image écran ci-dessus.

The screenshot shows a web browser window titled "Konan Marcellin" displaying a PHP application. The URL in the address bar is "localhost:82/pdoExemple/2012/executer.php". The page content is as follows:

**Client**

<b>numcl</b>	1
<b>nom</b>	Toto Alidou
<b>adresse</b>	01 BP 52 Abj 01
<b>tel</b>	20320188
<b>fax</b>	20320188
<b>email</b>	toto@yahoo.fr

**Commandes**

<b>numCom</b>	<b>dateCom</b>
1	2005-11-15
2	2005-11-17
6	2005-11-18

**Action buttons:** Afficher, Nouveau, Ajouter, Rechercher, Modifier, Supprimer

# X. PHP et Access

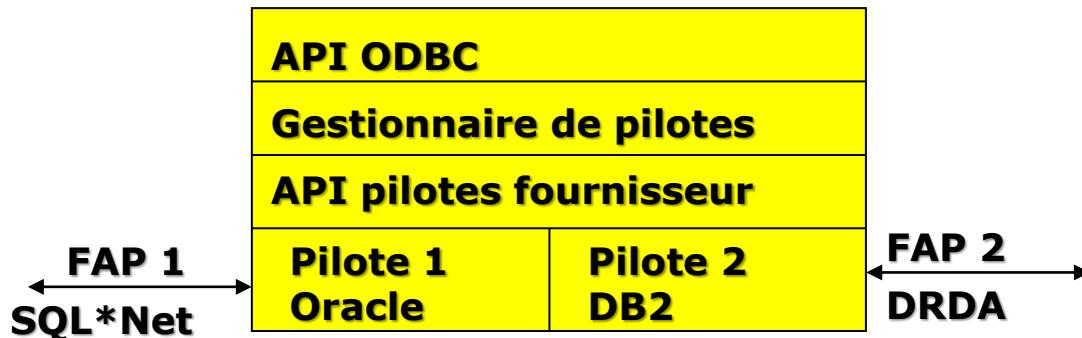
---

## □ 10.1 Présentation

- Pour accéder à Access, il faut définir une source de données ODBC.
- ODBC : Open DataBase Connectivity
  - Protocole standard d'accès aux données.
  - Il permet à une application de se connecter de la même façon à un ensemble de BD relationnelles différentes.
- Pilote ODBC (ou driver ODBC)
  - DLL qui répond à un ensemble de spécifications.
  - C'est le driver ODBC qui va implémenter les accès spécifiques à une BD particulière.
  - Il existe ainsi des drivers ODBC pour des BD : Access, SQL Server, Oracle, DB2...
  - Le driver s'appuie sur les librairies de bas niveau pour accéder aux BD :
    - DBLib pour SQL Server
    - SQLNet pour Oracle...

# X. PHP et Access

## ■ Architecture d'ODBC



- **API d'ODBC :**
  - Couche de réception des requêtes ODBC.
- **Gestionnaire de pilote :**
  - Couche de gestion des pilotes (à quel pilote passer une requête?) selon un format standard pour les pilotes ODBC.
- **Pilote i :**
  - chaque pilote de BD encapsule les requêtes dans le format propre à chaque fournisseur de BD.
- **FAP i (Format And Protocol) :**
  - Structure des messages et codage des données pour un SGBD.

# X. PHP et Access

---

## ■ Noms de source de données

- Ou Data Source Name (DSN)
- Permet de manipuler une BD dans une application
- Plusieurs types de DSN :
  - DSN Système : permet à tous les utilisateurs connectés à un serveur défini de se connecter à une BD ;
  - DSN Utilisateur : limite la connectivité à la BD à un utilisateur spécifique disposant des informations d'authentification nécessaires.
  - DSN fichier : fichier texte, fournit un accès à plusieurs utilisateurs et se transfère aisément d'un serveur à un autre par copie.

# X. PHP et Access

---

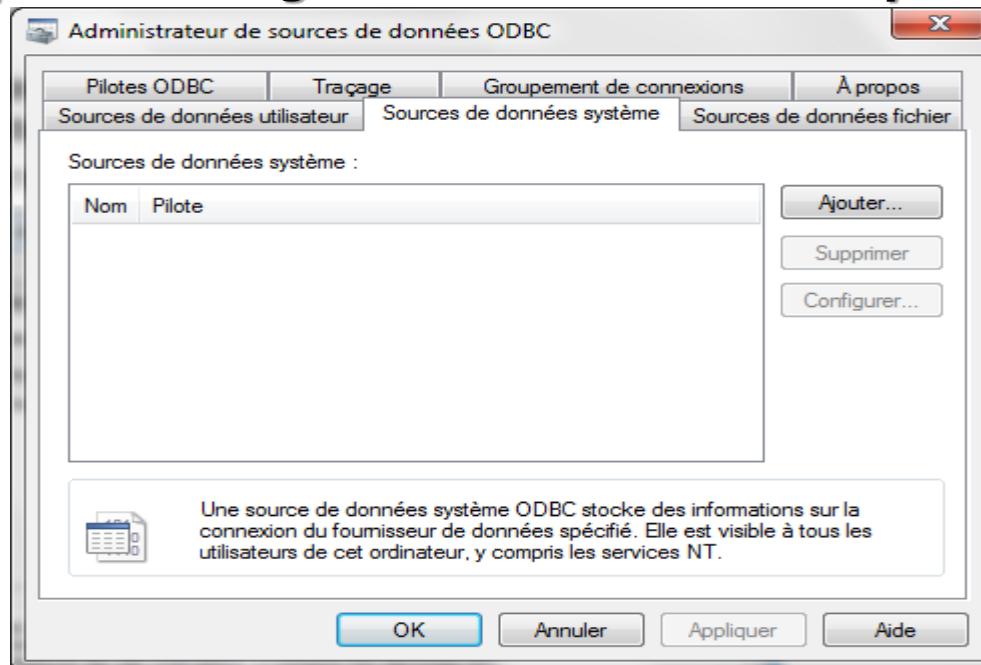
## ■ Comment Activer ODBC sur machine 64 bits

- Les DLLs x86 sont situées dans C:\Windows\SysWow64
- Les DLLs x64 sont situées dans C:\Windows\System32
- **Le nommage est trompeur, en effet, SysWow64 signifie "Windows On Windows" - et permet donc de lancer un processus 32-bits dans un OS 64-bits.**
- **Activer ODBC 32 bits**
  - **C:\Windows\syswow64\odbcad32.exe**

# X. PHP et Access

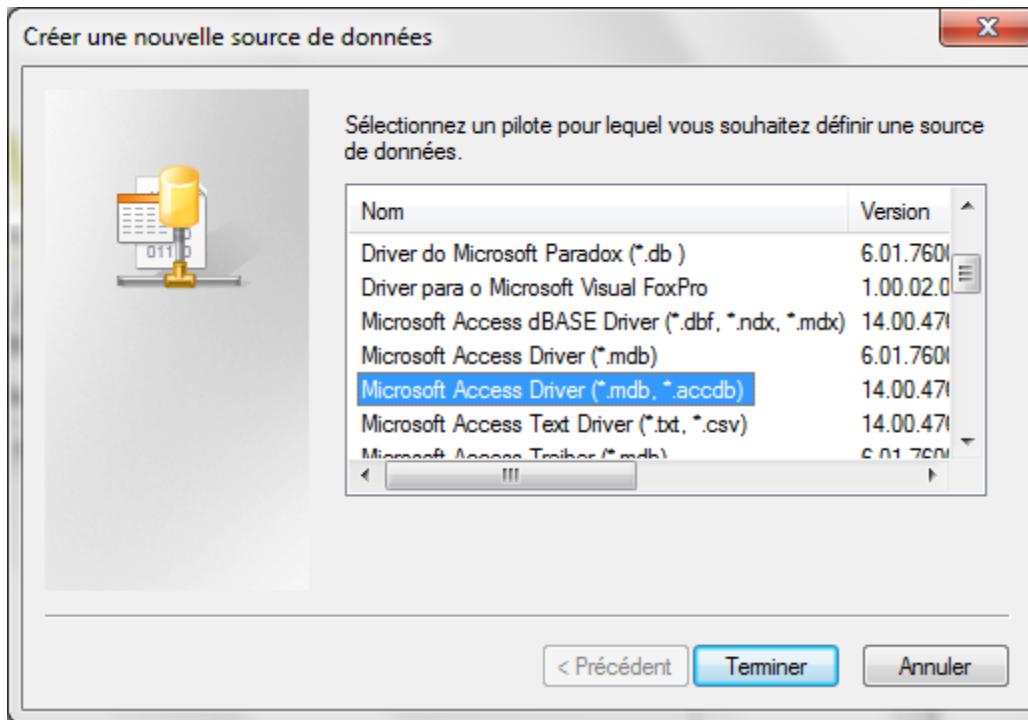
## □ 10.2. Configuration d'une source de données

- Panneau de configuration/Système et maintenance/Outils d'administration/Sources de données ODBC
- Cliquer sur l'onglet Source de données système



- Cliquer sur le bouton Ajouter

# X. PHP et Access



- ▣ Sélectionnez Microsoft Access Driver (\*.mdb, \*.accdb) puis cliquez sur Terminer.

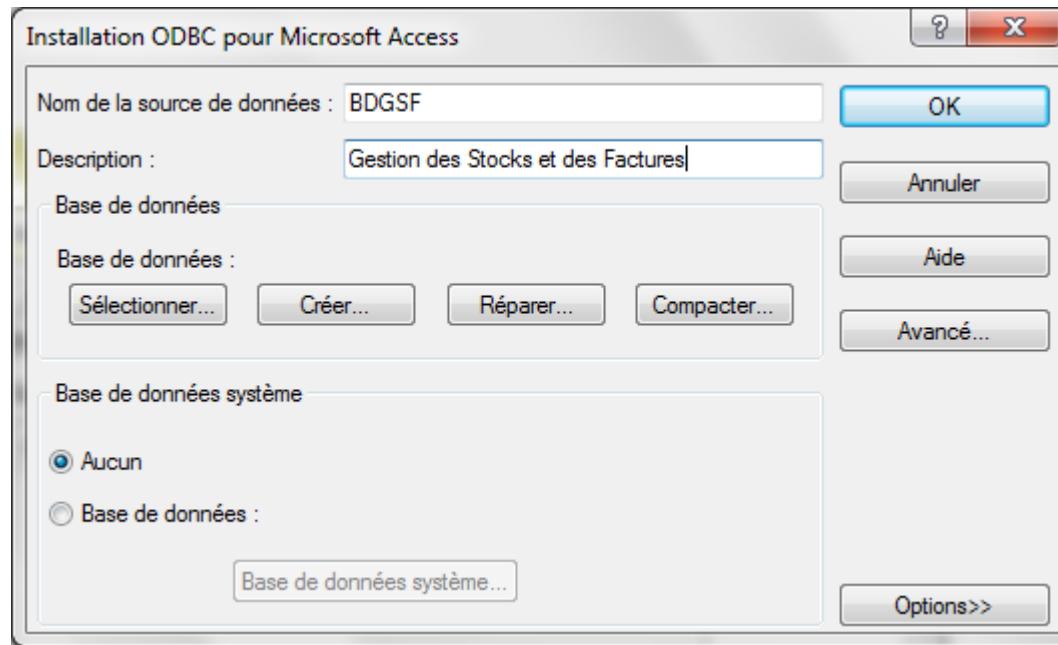
# X. PHP et Access

---

## ■ Remarque

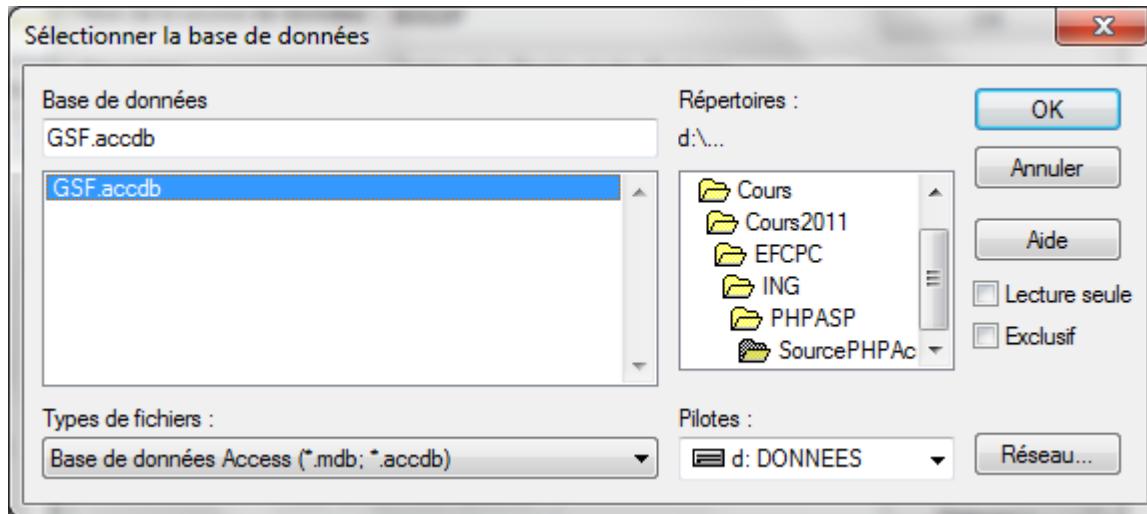
- ❑ Si le driver d'Access ne figure pas dans la liste, exécuter l'utilitaire suivante :
- ❑ Pour machine 32 bits : **AccessDatabaseEngine.exe**
- ❑ Pour machine 64 bits : **AccessDatabaseEngine\_X64.exe**

# X. PHP et Access



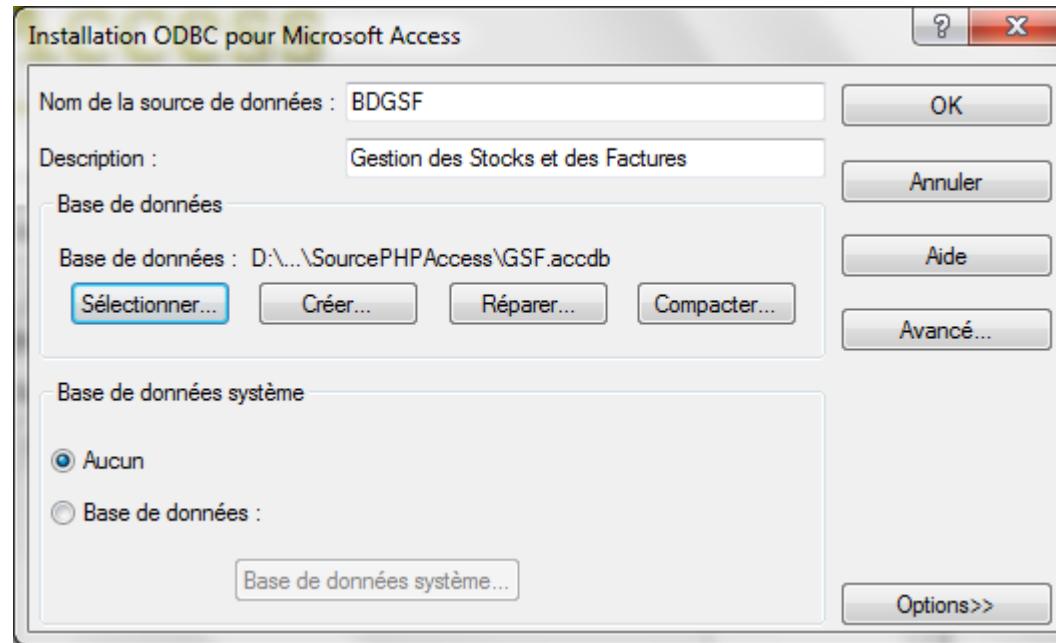
- ▣ **Saisir le nom de la source de données" (BDGSF)**
- ▣ **La description de la BD est facultative**
- ▣ **Cliquez sur le bouton Sélectionner**

# X. PHP et Access



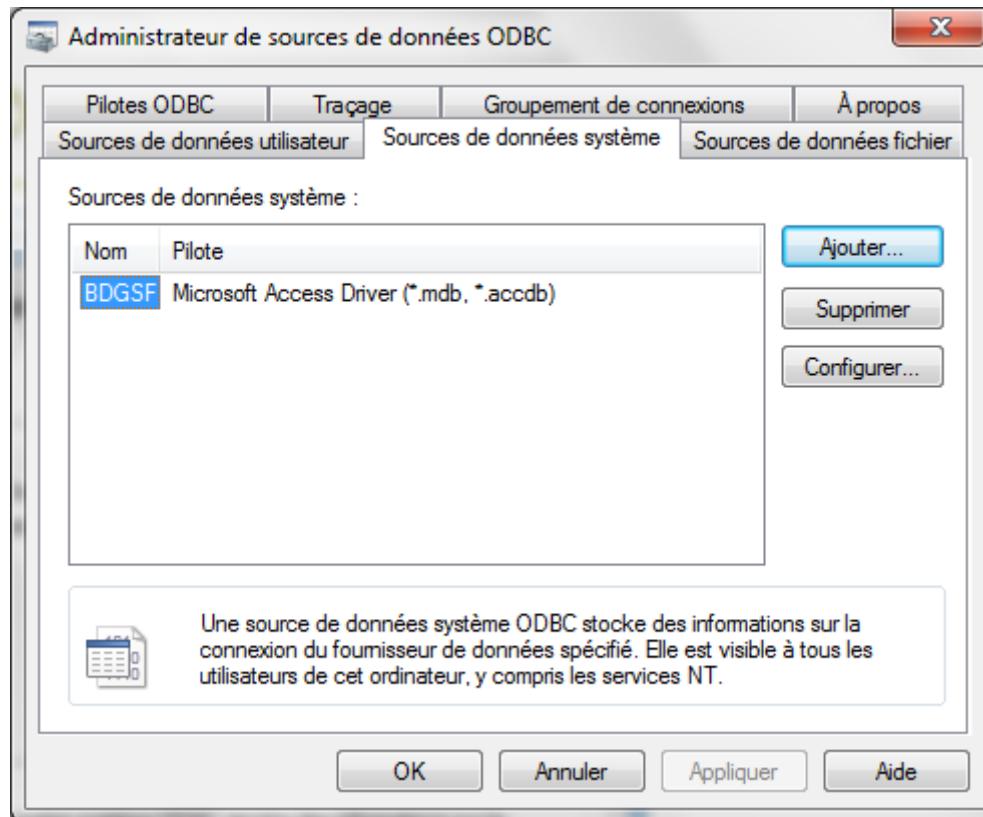
- ▣ **Sélectionner l'emplacement de la BD sur le disque dur de l'ordinateur**
- ▣ **Cliquer sur le bouton Ok**

# X. PHP et Access



- ▣ Cliquer sur le bouton Ok

# X. PHP et Access



- Cliquer sur le bouton Ok

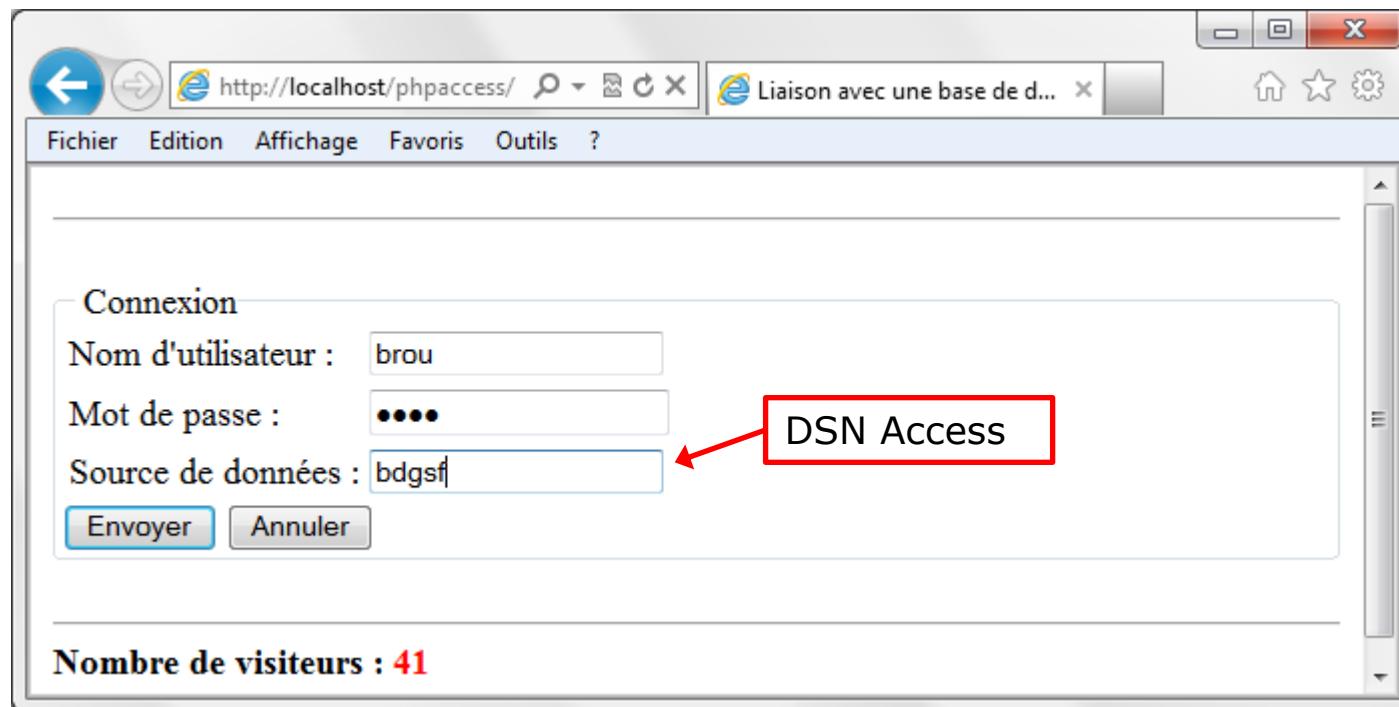
# X. PHP et Access

## □ 10.3. Crédation des scripts PHP

### ■ Page de garde : index.php

#### □ Fenêtre de connexion

□ \Webmaster\Support\Source\PDO\siteWeb\Access



#### □ Peu importe le nom d'utilisateur et son mot de passe

# X. PHP et Access

- **Scripts identiques à ceux de PHP/MySQL**
  - **C'est la chaîne de connexion qui change**

```
function connexionBD($serveur, $nom, $mpasse, $bd){  
    try {  
        $dsn = 'odbc:' . $bd ;  
        $db = new PDO($dsn, $nom, $mpasse);  
    }  
    catch ( Exception $e ) {  
        echo "Connection à la BD impossible : <BR>", $e->getMessage();  
        die();  
    }  
    return $db;  
}
```

# X. PHP et Access

## ■ ODBC ne supporte pas la méthode getColumnMeta

```
function afficher($resultat) {
    $nbCol = $resultat->columnCount();
    $nbClient = $resultat->rowCount();
    echo "<TABLE border='1'>";
    $tete="";
    for ($i = 0 ; $i < $nbCol; $i++) {
        $tete = $tete . "<TH>Col " . $i . "</TH>";
    }
    echo "<TR>" . $tete . "</TR>" ;
    $rows = $resultat->fetchAll(PDO::FETCH_NUM);
    foreach ($rows as $row) {
        $tete="";
        for ($i = 0 ; $i < $nbCol; $i++) {
            if($row[$i])
                $tete = $tete . "<TD>" . $row[$i] . "</TD>";
            else
                $tete = $tete . "<TD>" . "&nbsp;" . "</TD>";
        }
        echo "<TR>" . $tete ."</TR>" ;
    }
    echo "</TABLE>";
    echo "<BR><B>Il y a au total " . $nbClient . " enregistrements.</B>";
}
```

# X. PHP et Access

---

- ▣ **Activer l'extension PHP : php\_pdo\_odbc**
- ▣ **Créer un alias pour \PHPASP\SourcePHAccess :**
  - Alias : phpAccess
  - Chemin : \PHPASP\SourcePHAccess
- ▣ **Lancer un navigateur**
  - Saisir l'adresse : <http://localhost/phpAccess/index.html>

# X. PHP et Access

The screenshot shows a Microsoft Internet Explorer browser window. The address bar displays the URL `http://localhost/phpaccess/menu.php`. The title bar of the browser window also shows the text "Liaison avec une base de d...". The menu bar includes "Fichier", "Edition", "Affichage", "Favoris", "Outils", and "?". The main content area of the browser contains the following text:

Cette page montre comment utiliser les commandes PHP pour créer des pages Web dynamiques via ODBC afin d'effectuer les opérations suivantes :

---

- [Liste des clients](#)
- [Ajouter un nouveau client](#)
- [Rechercher un client](#)
- [Modifier les informations sur un client](#)
- [Supprimer un client](#)

---

[Retour au menu principal](#)

## ▣ Tester les liens

# XI. PHP et Oracle

---

## □ 11.1. Présentation

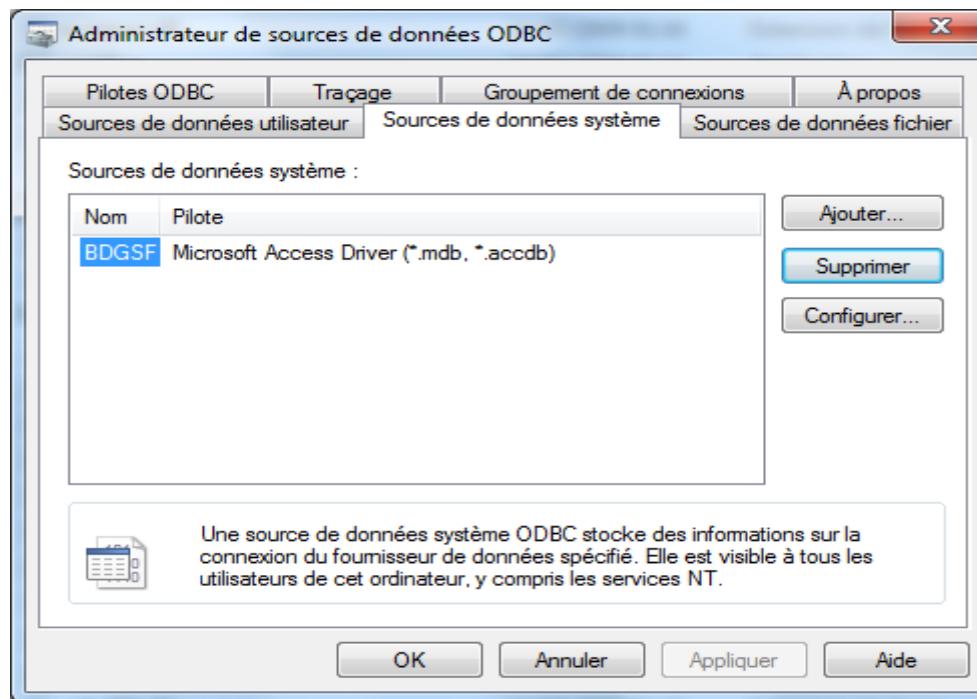
- Scripts identiques à ceux phpAccess
- Sauf :
  - Crédit à Oracle

# XI. PHP et Oracle

## ■ 11.2. Configuration d'une source de données

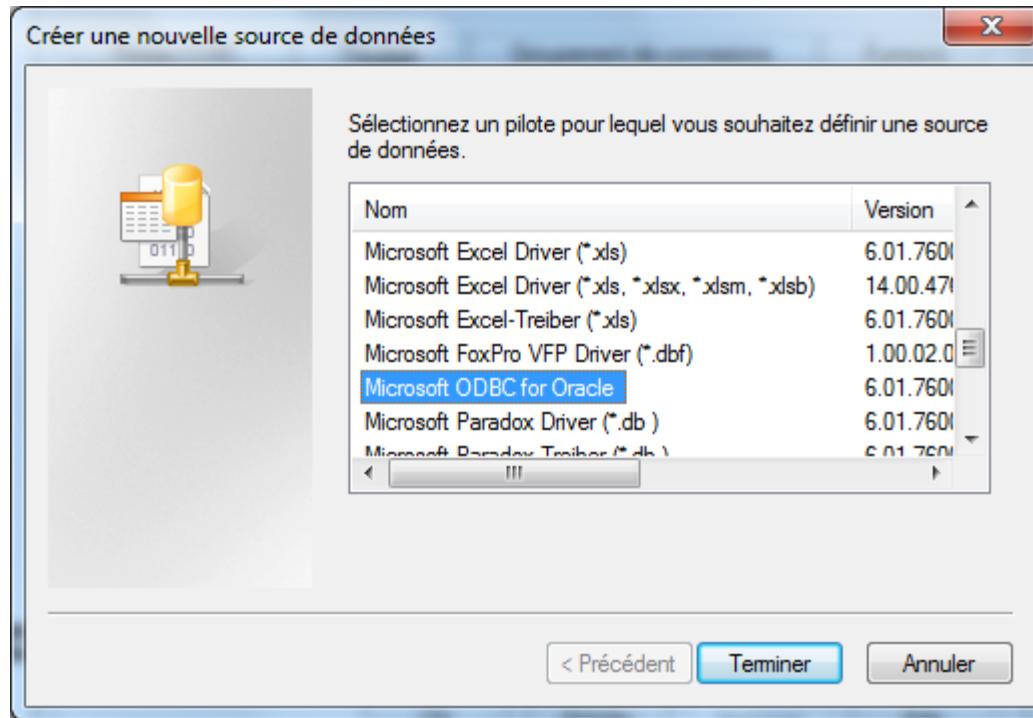
### ■ Processus

#### □ C:\Windows\SysWOW64\odbcad32.exe



#### □ Cliquer sur Ajouter

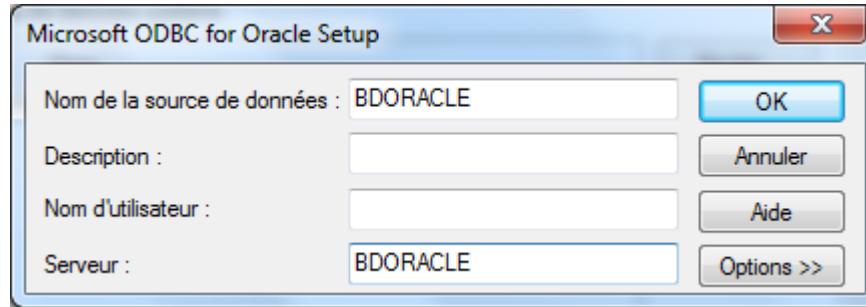
# XI. PHP et Oracle



- ▣ **Sélectionner Microsoft ODBC for Oracle**
- ▣ **Cliquer sur Terminer**

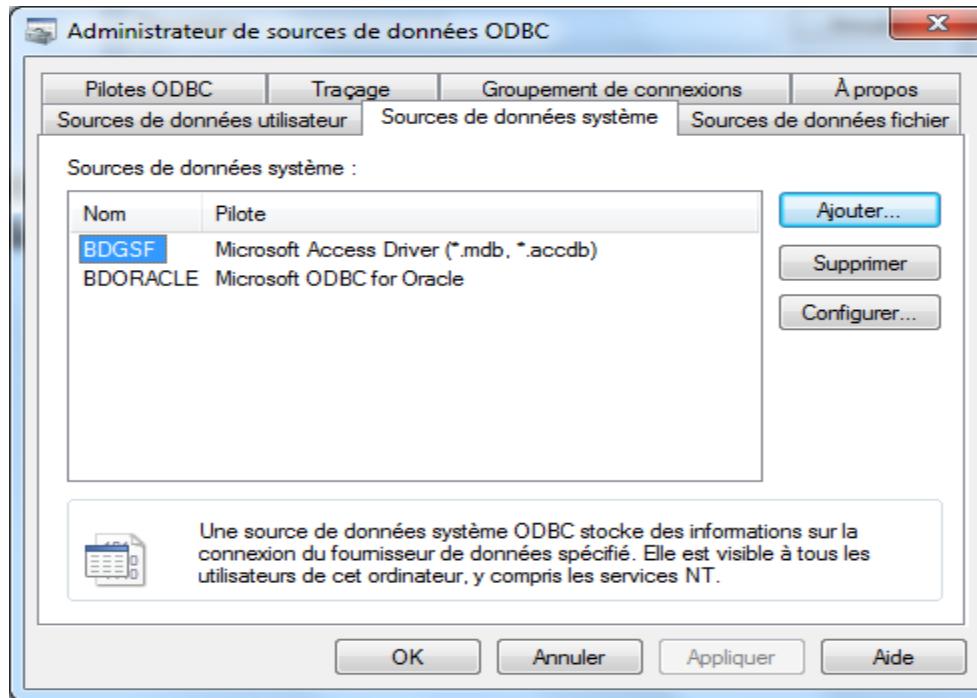
# XI. PHP et Oracle

---



- ▣ **Remplir les champs**
  - Nom de la source de données : nom de la DSN
  - Serveur : Nom de la BD Oracle
- ▣ **Cliquer sur OK**

# XI. PHP et Oracle



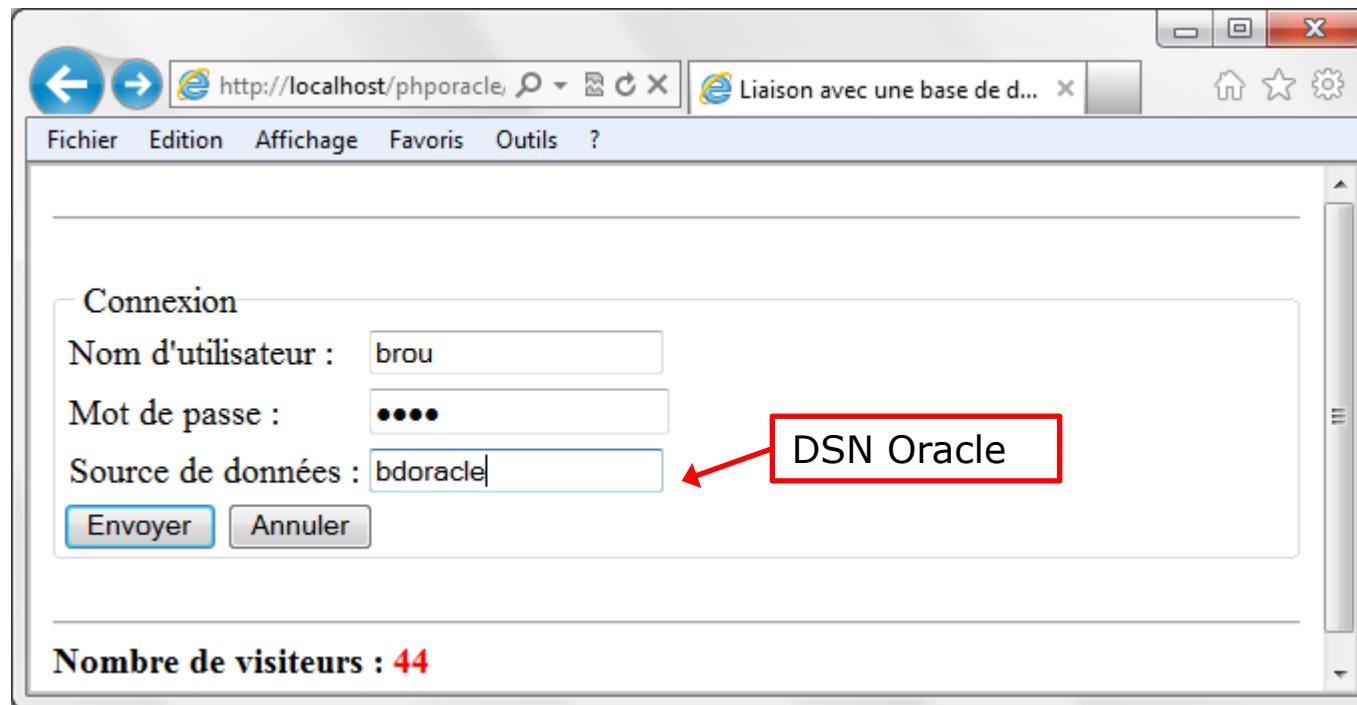
- ▣ **Cliquer sur OK**

# XI. PHP et Oracle

---

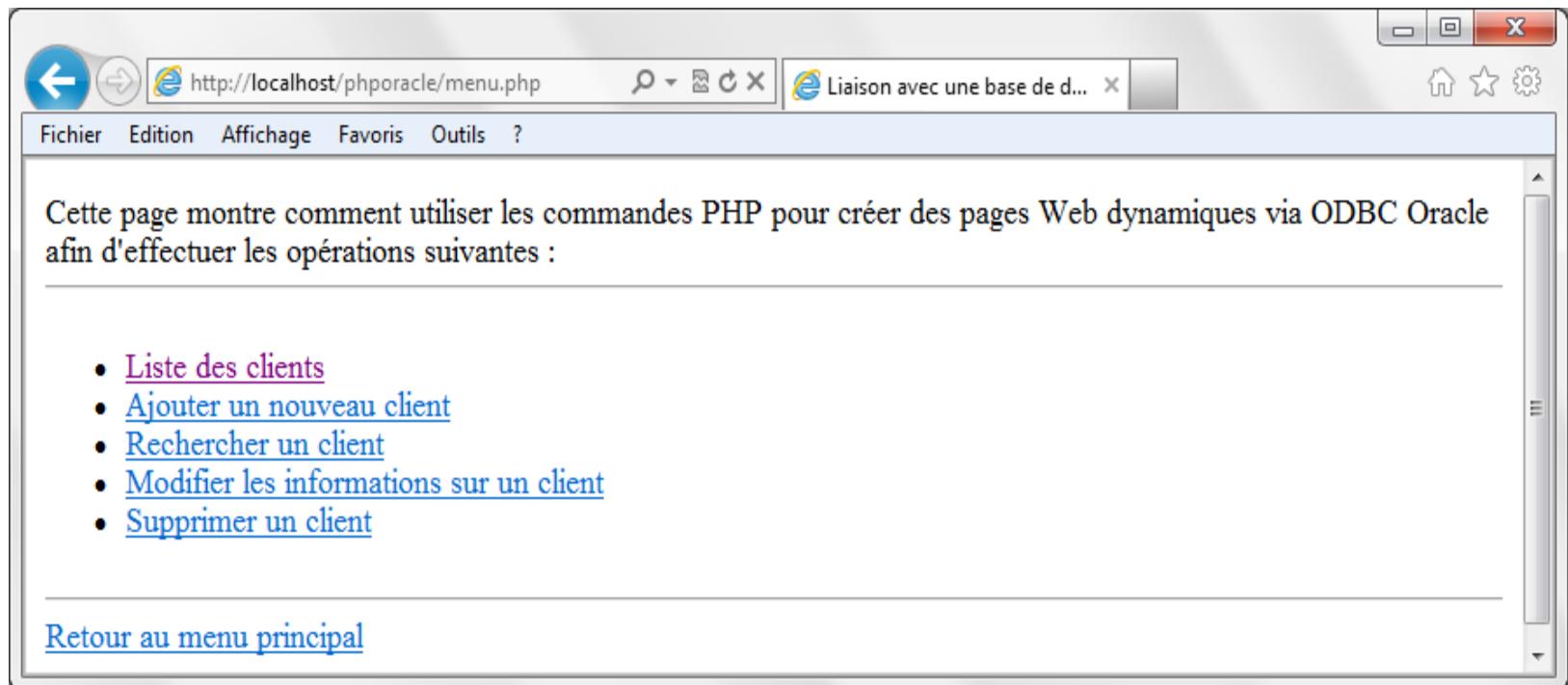
- **Scripts identiques à ceux de PHP/Access**
  - **Toutes les fonctions commencent par **odbc** et non **mysql****
    - odbc\_connect, odbc\_do ...
  - **Créer un alias pour \PHPASP\SourcePHPOracle :**
    - Alias : phporacle
    - Chemin : \PHPASP\SourcePHPOracle
  - **Lancer un navigateur**
    - Saisir l'adresse : <http://localhost/phpOracle/index.html>

# XI. PHP et Oracle



- **Le nom d'utilisateur et le mot de passe sont obligatoire**

# XI. PHP et Oracle



## □ Tester les liens

# Bibliographie

---

## □ Livres

- **Supports de cours Oracle Formation :**
  - ▣ **Programmer avec PL/SQL**
  - ▣ **Le langage SQL & l'outil SQL\*Plus**

## □ Webographie

- <http://cyberzoide.developpez.com>
- [http://maximilian.developpez.com/mysql/nouveautes\\_mysql5/](http://maximilian.developpez.com/mysql/nouveautes_mysql5/)