



<html>

HTML



HTML



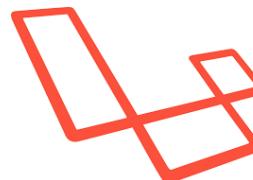
CSS



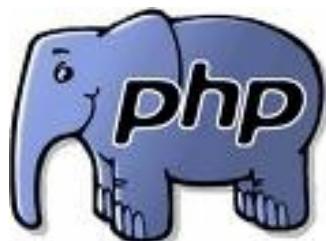
Frameworks du Web



JavaScript



MariaDB
MySQL



Ahmed ZELLOU

ENSIAS, 2019-2020.

Plan global

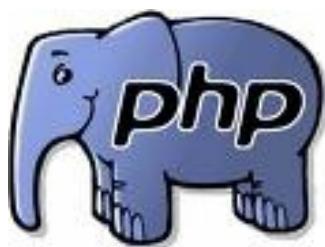
- Le Web Statique
- Le Web dynamique
- PHP
- PHP Objet
- MVC
- Sécurisation d'applications Web
- Le framework Symfony
- Le framework Laravel



<html>



Développement des Applications Web



Ahmed ZELLOU



IWIM, eMBI, ENSIAS, 2019-2020.

Plan du Semestre

Semaine	Contenu Pédagogique
1	PHP Partie 1
2	PHP Partie 2
3	TP
4	PHP Objet
5	TP
6	Framework Symfony
7	TP

Quelques Règles



Pas de GSM



Pas de Bavards



Poser des Questions

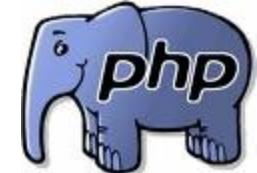


Partager votre expérience



Tout est relatif

$$E = mc^2$$



Développement des Applications Web

PHP, la Programmation

Ahmed ZELLOU

IWIM, eMBI, ENSIAS, 2019-2020.

Plan PHP

- Introduction à PHP
- Support de PHP
- Premier Script
- Installation
- Les Variables & Constantes
- Les Opérateurs
- Les Tableaux
- Tests, boucles et fonctions

Introduction à PHP

- 1994: Crée par l'ingénieur canadien d'apache Rasmus Lerdorf.
- PHP 1.0 (Personnal Home Page), 1994-1995.
- PHP/FI 2.0. 1995-1997, Personal Home Page Tool/Form Interpreter.
- PHP3, 1997-2000.
- PHP4, 2000, Zend, ZEev et aNDy.
- PHP5 2005, PDO.
- PHP6, 2010, version abandonnée.
- PHP7, décembre 2015, support de l'unicode.
- Last Version : 7.4.1 Déc 2019.



Introduction à PHP

- Signifie PHP: Hypertext Preprocessor.
- Langage de script Open source qui s'exécute côté serveur, code protégé.
- Extension utilisée sur la plupart des serveurs Web.
- Syntaxe basée sur C, Java et Perl.
- Supporte la programmation procédurale et orientée objet.
- Langage de type Server-Side Scripting comme :
 - Practical extraction and Report Language (Perl) ;
 - Active Server Pages (ASP) ;
 - Java server pages (JSP) ;
 - Ruby.

Introduction à PHP

- Performance [Zdnet Statistics]^{*} :
 - PHP interprète près de **47** pages/secondes ;
 - Microsoft ASP interprète près **43** pages/secondes ;
 - Allaire ColdFusion interprète près **29** pages/secondes ;
 - Sun Java JSP interprète près de **13** pages/secondes.

* selon une configuration basique.

Comparaison

https://w3techs.com

rie de composan W Bulletin officiel du Ma Google Translate Nouvel onglet G Goog

Server-side Programming Languages

Most popular server-side programming languages

© W3Techs.com	usage	change since 1 December 2018
1. PHP	78.9%	
2. ASP.NET	11.8%	-0.1%
3. Java	4.0%	+0.1%
4. Ruby	2.4%	+0.1%
5. static files	2.1%	

percentages of sites

Web Servers

Most popular web servers

© W3Techs.com	usage	change since 1 December 2018
1. Apache	44.6%	-0.2%
2. Nginx	40.7%	+0.3%
3. Microsoft-IIS	9.0%	-0.1%
4. LiteSpeed	3.7%	+0.1%
5. Google Servers	0.9%	

percentages of sites

Source : <https://w3techs.com/>

Comparaison

Solution	PHP	Python	ASP	JSP
	<ul style="list-style-type: none"> ✓ Gratuit ✓ Plusieurs projets disponibles ✓ Code optimisé ✓ Hébergement largement supporté ✓ Documentation disponible 	<ul style="list-style-type: none"> ✓ Gratuit ✓ Plusieurs failles de sécurité ✓ Stabilité du code dans le temps ✓ Langage littéraire simple à comprendre 	<ul style="list-style-type: none"> ✓ Compilé ✓ Soutenu par Microsoft ✓ Documentation disponible 	<ul style="list-style-type: none"> ✓ Pseudo-compilé ✓ Plus Portable ✓ Code stable dans le temps ✓ Soutenu par Oracle
	<ul style="list-style-type: none"> ✓ Pas de structure pré définit ✓ Langage volatil 	<ul style="list-style-type: none"> ✓ Nécessite des frameworks pour le Web (Django, Grok, Pylons, Turbo Gears, web2py,...). 	<ul style="list-style-type: none"> ✓ Propriétaire, Coût élevé ✓ Hébergement exclusive Windows 	<ul style="list-style-type: none"> ✓ Gourmant en ressources ✓ Hébergement moins disponible

Support de PHP

- Multi plate-forme : UNIX, Linux, Windows, Mac OS X, ...
- Multi-Serveurs : Apache, Tomcat, IIS, Nginx, ...
- Multi-SGBD : MySQL, SQLServer, Oracle, MariaDB, PostgreSQL, SQLite, Access, MongoDB, Casandra, mSQL, DB2, Informix, Sybase, Adabas, Ingres, dBase, InterBase, Ovrimos, Empress, FrontBase, FilePro, Solid Hyperwave, Direct MS-SQL, Velocis, Informix, ODBC, Unix dbm, ...
- Supports ODBC, the Open Database Connection standard,
- Supporte plusieurs protocoles du web : HTTP, HTTPS, FTP, LDAP, SNMP, POP3, IMAP, NNTP, COM, ...
- Support de XML.

Premier Script

- PHP peut être utilisé de deux manières différentes :
 - soit comme un langage de programmation traditionnel (en ligne de commande)
 - soit intégré dans une page HTML.
- Les fichiers PHP ont une extension de fichier **.php** ou **.phtml**
- Un code PHP peut être placé n'importe où dans le document HTML.
- Un Code PHP entre **<?php** et **?>**, **<?language="php">** et **</script>**
- Chaque instruction doit se terminer par **;**.
- Double instructions pour l'affichage **echo**, **print** et **printf**.
- **var_dump** est plus verbeuse;

Premier Script

```
<html>
<head><title>Premier Script PHP</title></head>
<body>
<?php
    echo "Bonjour, je suis un script PHP!";
?
</body>
</html>
```

Ou

```
<?php
echo "<html> <head><title>Premier Script
PHP </title></head><body> Bonjour, je suis
un script PHP! </body> </html> ";
?
"
```

Commentaires

```
<?php
// sur une seule ligne.
# jusqu'à la fin de la ligne
/* et */ sur plusieurs lignes.
?>
```

Script Shell

- Permet d'exécuter des commandes Shell à travers un script PHP.
- ` est similaire à la fonction `shell_exec()`
 - Méthode 1 :
``ls -al`;`
 - Méthode 2 :
`shell_exec("ls -al");`
- Des fonctions de gestion de fichiers et de répertoires :
Copie, renommer, supprimer, consulter, ...

Script Shell

Fonction	Description
<code>chmod()</code>	modifier les droits d'accès
<code>chown()</code>	modifier le propriétaire
<code>chgrp()</code>	modifier le groupe
<code>mkdir()</code>	créer un répertoire
<code>rmdir()</code>	détruire un répertoire
<code>copy()</code>	copier un fichier
<code>basename()</code>	extrait le nom du fichier dans le chemin
<code>dirname()</code>	extrait le chemin du fichier sans le nom
<code>link()</code>	créer un lien physique
<code>symlink()</code>	créer un lien symbolique
<code>unlink()</code>	détruire un lien
<code>touch()</code>	changer la date de modification

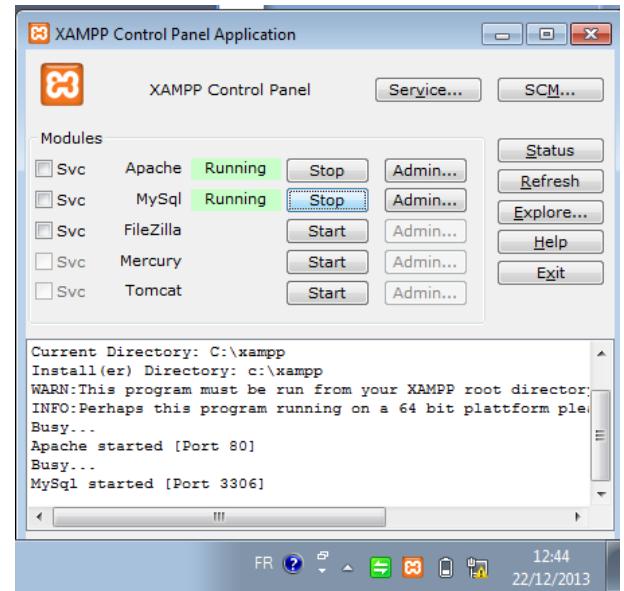
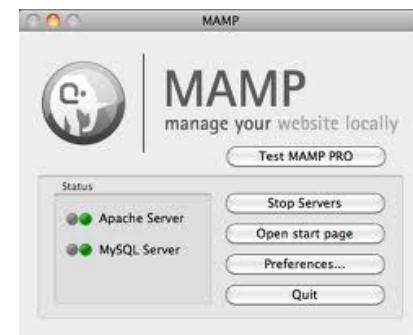
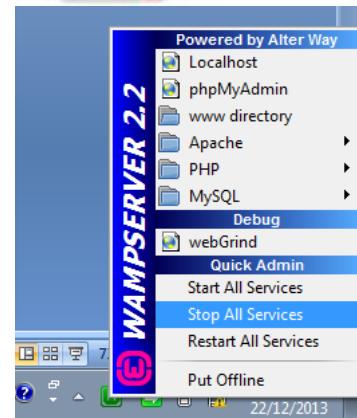
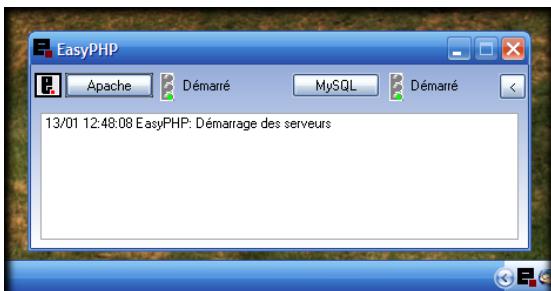
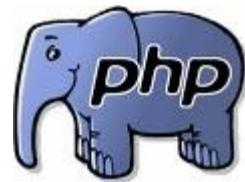
Script Shell

■ Autres fonctions :

Fonction	Fonction	Fonction	Fonction
<code>file_exists</code>	<code>is_dir</code>	<code>is_file</code>	<code>is_link</code>
<code>filetype</code>	<code>is_readable</code>	<code>is_executable</code>	<code>is_writable</code>
<code>disk_free_space</code>	<code>disk_total_space</code>	<code>filesize</code>	<code>fileatime</code>
<code>filectime</code>	<code>filemtime</code>	<code>fileowner</code>	<code>filegroup</code>
<code>fileperms</code>	<code>readfile</code>	<code>file_get_contents</code>	<code>file</code>
<code>fopen</code>	<code>fread</code>	<code>fwrite</code>	<code>fseek</code>
<code>ftell</code>	<code>fgetc</code>	<code>fgets</code>	<code>fclose</code>

Installation

- EasypHP, wampp, xampp, Lampp, bitNami, ... PHP



Les Variables

- Toutes les variables commencent en PHP par \$.
- PHP est un langage faiblement typé.
- Déclaration : **\$var = value ;**
- Le nom d'une variable doit être un nom valide.
- Types d'Affectations :
 - Affectation par valeur : **\$b=\$a ;**
 - Affectation par référence : **\$c = &\$a ;**
- Types de variables :
 - Variable locale : **\$var = value ;**
 - Variable globale : **global \$var = value ;**

Les Variables PHP

- Les différentes types :
 - Entier: 12, Réel: 1.54
 - Chaîne de caractères : "Hello" ou 'Bonjour'
 - Booléen: true, false
 - Tableau: \$tab[0]=12
 - Objet (PHP4, PHP5)
 - Ressource
 - NULL
- Bibliothèque riche de fonctions sur les variables :
 - Doubleval(), empty(), gettype(), settype(), intval(), is_array(), is_bool(), is_double(), is_float(), is_int(), is_integer(), is_long(), is_object(), is_real(), is_numeric(), is_string(), iset(), unset(), strval()

Les Constantes

- Constantes :
 - `define("NAME", valeur);`
 - `echo NAME;`
- Exemple :
 - `define("Ecole", "ENSIAS");`
 - `define("YELLOW", "#FFFF00");`
 - `define("PI", 3.14);`
 - `define("RL", "
");`
- Affichage :
 - `print("Notre école est : " . Ecole. RL);`
- Par convention, les constantes doivent être en majuscule.

Les Constantes Prédéfinies

■ Quelques constantes : get_defined_constants

Constante	Description
NULL	affiche la variable NULL
TRUE	affiche la variable TRUE
FALSE	affiche la variable FALSE
__FILE__	le fichier contenant le script
__LINE__	la ligne contenant le script
PHP_VERSION	la version du PHP
PHP_OS	Le système d'exploitation de PHP
E_ERROR	affiche l'erreur, 1 sinon.

Les Constantes Prédéfinies

■ Les constantes d'environnement

Constante	Description
INFO_CONFIGURATION	les informations de configuration.
INFO_CREDITS	les informations sur les auteurs du module PHP
INFO_ENVIRONMENT	les variables d'environnement.
INFO_GENERAL	les informations sur la version de PHP.
INFO_LICENSE	la licence GNU Public
INFO_MODULES	les informations sur les modules associés à PHP
INFO_VARIABLES	les variables PHP prédéfinies.

- Affichage des variables d'environnement : `phpinfo();`
- `echo getenv("HTTP_USER_AGENT");`

Les Variables Dynamiques

- Définition :

- \$var = "nom";
 - \$\$var= "valeur";

- Affichage :

- echo \$var ;
 - echo \$\$var ;

- Exemple :

```
echo "Bonjour, je suis un script PHP!";
define("Ecole", "ENSIAS");
$var = "département";
$$var= "IWIM-BI";
echo $$var; //donne IWIM-BI
$$var=" spécialisé du web et Business";
echo $$var.' est un '.$var.$$var.' de '.Ecole; //donne IWIM-BI est un
département spécialisé du web et Business de ENSIAS
```

Les Variables Prédéfinies

■ Les variables d'environnement dépendant du client

Variable	Description
<code>\$_SERVER["HTTP_HOST"]</code>	Nom d'hôte de la machine du client
<code>\$_SERVER["HTTP_REFERER"]</code>	URL de la page qui a appelé le script PHP
<code>\$_SERVER["HTTP_ACCEPT_LANGUAGE"]</code>	Langue utilisée par le serveur (par défaut en-us)
<code>\$_SERVER["HTTP_ACCEPT"]</code>	Types MIME reconnus par le serveur (séparés par des virgules)
<code>\$_SERVER["CONTENT_TYPE"]</code>	Type de données contenu présent dans le corps de la requête. Il s'agit du type MIME des données
<code>\$_SERVER["REMOTE_ADDR"]</code>	L'adresse IP du client appelant le script
<code>\$_SERVER["PHP_SELF"]</code>	Nom du script PHP

Les Variables Prédéfinies

■ Les variables d'environnement dépendant du serveur

Variable	Description
<code>\$_SERVER["SERVER_NAME"]</code>	Le nom du serveur
<code>\$_SERVER["HTTP_HOST"]</code>	Nom de domaine du serveur
<code>\$_SERVER["SERVER_ADDR"]</code>	Adresse IP du serveur
<code>\$_SERVER["SERVER_PROTOCOL"]</code>	Nom et version du protocole utilisé pour envoyer la requête au script PHP
<code>\$_SERVER["DATE_GMT"]</code>	Date actuelle au format GMT
<code>\$_SERVER["DATE_LOCAL"]</code>	Date actuelle au format local
<code>\$_SERVER["DOCUMENT_ROOT"]</code>	Racine des documents Web sur le serveur

Quelques fonctions de String

Fonction	Description
strlen (chaîne);	retourne la longueur d'une chaîne.
strcmp(chaîne1, chaîne2);	compare deux chaines de caractères
strstr(chaîne, sous-chaîne);	cherche une sous chaine dans une chaine
strchr(chaîne, char);	cherche un char dans une chaine
str_replace (chaîne, sous-chaîne);	remplace une sous chaine dans une chaine
strpos(chaîne,sois-chaîne)	cherche une sous-chaîne dans une chaîne. retournera la position de la première occurrence. Sinon FALSE.
explode(délimiteur, chaîne);	scinde une chaîne en fragments à l'aide d'un délimiteur et retourne un tableau.
addCSlashes(chaîne, liste_caractères);	ajoute des slashes dans une chaîne
chop(chaîne);	supprime les espaces blancs en fin de chaîne.
chr(nombre);	retourne un caractère en mode ASCII
crypt(chaîne [, chaîne_code])	code une chaîne avec une base de codage.

Autres : <http://php.net/manual/fr/indexes.functions.php>

Les Expressions Régulières

- Exemples :
 - [a-z], [aeiou], [a-zA-Z0-9]
 - [!a-z]
 - (exp)*, (exp)+, (exp)?,
 - (exp){n,m}, (exp){n}, (exp){n,}, (exp){,m}
 - ^(exp), (exp)\$
 - [[:alnum:]], [[:alpha:]], [[:digit:]]
 - [[:space:]], [[:lower:]], [[:upper:]],
- Utilisez les fonctions : **ereg** et **preg_match**

Les Opérateurs

- Arithmétiques : +, -, *, /, %,
- De Pas : ++, --.
- D'assignation : =, +=, -=, *=, /=, .=, %=, |= (Ou logique), ^=(Ou exclusif), &= (Et logique),
- De comparaison : ==, !=, <>, >, <, >=, <=, ===, !==, <=>
- Logiques : &&, ||, !, and, or, not, xor
- De concaténation : .
- Bit-à-bit : & (et), | (ou), ^ (ou exclusif), ~ (non)
- De rotation de bit : <<(rotation à gauche), >> (rotation à droite).
- Autres : & (référencement variable), -> (propriétés d'un objet)
- Ternaire : \$var=condition?vrai:faux;

\$var=conditionTrue?? conditionFalse;

Les Tableaux : Trois types

1. Tableau à indice numérique

- Méthode 1 :

```
$Villes[0] = "Rabat";  
$Villes[1] = "Casa";  
$Villes[2] = "Tanger";
```

- Le premier élément du tableau se trouve dans l'indice 0.
- Méthode 2 : `$Villes = array ("Rabat", "Casa", "Tanger");`
- Méthode 3: `$Villes = array(0=>"Rabat", 1=>"Casa", 2=>"Tanger");`
- Méthode 4 : `$Villes = array(0=>"Rabat","Casa","Tanger");`
- Méthode 5 : `array_push($Villes,"Rabat"); array_push($Villes,"Casa"); array_push($Villes," Tanger ");`
- Affichages :

```
echo $Villes[0]. " et ". $Villes[1]. " sont des villes marocaines."
```

Les Tableaux : Trois types

- Les éléments du tableaux ne sont pas forcement d'indices consécutifs :

```
$stab[0] = 12;  
$stab[1] = "Ahmed";  
$stab[2] = 3.14;  
$stab[5] = "C";
```

- Affichage : méthode 1

```
foreach($stab as $valeur)  
{    echo "Val: $valeur <br>"; }
```

- Affichage : méthode 2

```
foreach($stab as $cle => $valeur)  
{    echo "Valeur ($cle): $valeur"; }
```

Les Tableaux : Trois types

2. Tableau associatif : Les indices ne sont pas des entiers positifs

- Méthode 1 :

```
$taux ['nom'] = "tva";
```

```
$taux['C'] = 30;
```

```
$taux[3] = 34;
```

```
$taux[3.14] = 'PI';
```

- Méthode 2

```
$Age = array ("Khalid" => 32, "Aicha" => 30, "Driss" => 34);
```

- Affichage

```
echo " Khalid a ". $Age['Khalid']. "ans."
```

Les Tableaux : Trois types

3. Tableau multidimensionnel : tableau de tableaux

```
$tab1= array( ... );  
$tab2= array( ... );  
$tableau = array($tab1, $tab2);
```

- Exemple :

```
$mois = array("Janvier", "Février", "Mars", "Avril", "Mai", "Juin",  
    "Juillet", "Août", "Septembre", "Octobre", "Novembre", "Décembre");  
$jour = array("Dimanche", "Lundi", "Mardi", "Mercredi", "Jeudi",  
    "Vendredi", "Samedi");  
$element_date = array($mois, $jour);
```

- Appellation :

```
echo $element_date[1][4] . " 10 " . $element_date[0][0] . "2020";
```

Affichage Tableaux

- Méthode 1 : loop

```
for ($i=0; $i < count($stab); $i++)  
    echo " tab[". ($i+1)."]= ".$stab[$i]." <br>";
```

- Méthode 2 : une boucle while

```
$i=0;  
while ($stab[$i])  
{echo $stab[$i++], "<br /> "; }
```

- Méthode 3 : each

```
while ($item = each( $stab ))  
print $item["key"]." est : ". $item["value"].'<br>';
```

- Méthode 4 : each and list

```
while ( list($key,$value) = each( $stab ))  
print "$value à la position $key<br>";
```

- Méthode 5 : print_r

```
print_r($stab)
```

Tableaux : Quelques fonctions

Fonction	Description
sort	effectue un tri selon un ordre alphanumérique ASCII , ne préserve pas les clés pour les tableaux associatifs .
rsort	effectue le tri selon l'ordre alphanumérique inverse.
asort	effectue le tri selon l'ordre alphanumérique qui préserve les clés pour les tableaux associatifs
ksort	effectue le tri des clés selon le critère des codes ASCII, et préserve les associations clé / valeur
krsort	effectue le tri des clés selon le critère des codes ASCII, et préserve les associations clé / valeur
natcasesort	effectue un tri dans l'ordre alphabétique non ASCII (« a » est avant « z » et « 10 » est après « 9 »)
array_combine	Crée un tableau par l'usage d'un premier tableau pour les clés et un autre pour les valeurs
array_walk_recursive	Execute une fonction récursive à chaque élément du tableau
array_count_values	retourne un tableau comptant le nombre d'occurrences des valeurs d'un tableau.

Tableaux : Quelques fonctions

■ Autres fonctions :

Fonction	Fonction	Fonction	Fonction
array_diff	array_intersect	array_flip	array_keys
array_map	array_merge	array_merge_recursive	array_multisort
array_pad	merge	push	slice
pop	keys	splice	count
array_rand	array_unique	array_walk	array_search
array_product	array_replace	array_combine	array_diff_uassoc
array_filter	compact	in_array	key_exists

Les Tableaux : Etude de Cas

- Simulation d'une base de données : library.txt

php manual	ZELLOU	Eyrolles	330
perl manual	KASSOU	Privat	540
C manual	RACHAD	Milan	480

- Traitement

```
$books = file("library.txt"); // cherche la base de données "library.txt"  
for ($i=0; $i<.....; $i++)  
$books_array[$i] = explode( "\t", $books[$i]); // extraire les éléments par lignes  
for ($i=0; $i<.....; $i++)  
print "Le livre : ..... de l'auteur : .....  
" édité par ". .... " avec un prix : ". .... "<BR>";
```

- Résultat

```
Le livre : php manual de l'auteur : Auteur1, édité par Eidteur1 avec un prix : 330  
Le livre : perl manual de l'auteur : Auteur2, édité par Eidteur2 avec un prix : 540  
Le livre : C manual de l'auteur : Auteur3, édité par Eidteur3 avec un prix : 480
```

Les Tableaux : Etude de Cas

- Simulation d'une base de données : library.txt

php manual	ZELLOU	Eyrolles	330
perl manual	KASSOU	Privat	540
C manual	RACHAD	Milan	480

- Traitement

```
$books = file("library.txt"); // cherche la base de données "library.txt"  
for ($i=0; $i<count($books); $i++)  
{$books_array[$i] = explode( "\t", $books[$i]); // extraire les éléments par lignes  
for ($i=0; $i<count($books_array); $i++)  
print "Le livre : ".$books_array[$i][0]." de l'auteur : ".$books_array[$i][1].  
" édité par ".$books_array[$i][2]. " avec un prix : ".$books_array[$i][3]."<BR>";
```

- Résultat

```
Le livre : php manual de l'auteur : Auteur1, édité par Eidteur1 avec un prix : 330  
Le livre : perl manual de l'auteur : Auteur2, édité par Eidteur2 avec un prix : 540  
Le livre : C manual de l'auteur : Auteur3, édité par Eidteur3 avec un prix : 480
```

Tests, boucles et fonctions

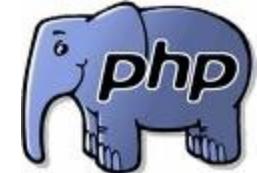
- Les instructions conditionnelles
 - if, else, elseif, switch et ternaire.
- Les boucles : for, while, do ... While et foreach.
- Les fonctions :
 - Sans paramètres : function nomFonction() { code ; }
 - Avec paramètres
 - Avec paramètres par défaut
 - Avec type de retour en PHP7
 - Anonymes
 - Passage par références
 - Récursives

Les Fonctions Prédéfinies

- La véritable puissance de PHP vient de ses fonctions.
 - String functions
 - Array functions
 - Calendar functions
 - Date functions
 - Directory functions
 - Error functions
 - Filesystem functions
 - Filter functions
 - FTP functions
 - HTTP functions

Les Fonctions Prédéfinies

- La véritable puissance de PHP vient de ses fonctions.
 - LibXML functions
 - Mail functions
 - Math functions
 - DB functions
 - SimpleXML functions
 - Zip functions
 - Divers autres fonctions.



Développement des Applications Web PHP, Programmation Serveur

Ahmed ZELLOU

IWIM, eMBI, ENSIAS, 2019-2020.

Plan PHP

- SII Server Side Includes
- Gestion de fichiers
- File Upload
- Gestion des erreurs
- Cookies
- Mail
- Sessions
- Ateliers

Les Formulaires des utilisateurs

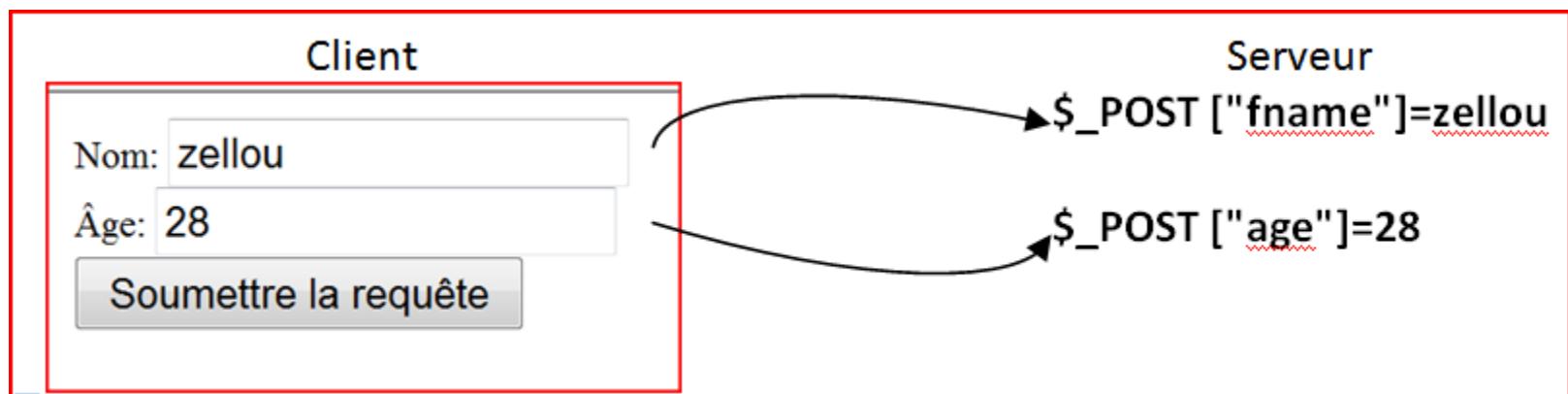
- \$_GET et \$_POST sont des tableaux superglobaux.
- Utilisées pour récupérer des informations à partir des formulaires.
- Accessibles en lecture et en écriture.
- Exemple :
- L'exemple ci-dessous contient un formulaire HTML avec deux champs de saisie et un bouton de soumission.

```
<html>
<body>
<form action="welcome.php" method="post">
Nom: <input type="text" name="fname"/> <br/>
Âge: <input type="text" name="age"/> <br/>
<input type="submit" value="Soumettre la requête"/>
</form>
</body>
</html>
```

The image shows a simple HTML form within a red-bordered box. The form consists of two text input fields: one for 'Nom:' and one for 'Âge:'. Below these fields is a single button labeled 'Soumettre la requête'.

Les Formulaires des utilisateurs

- Lorsqu'un utilisateur remplit le formulaire et clique sur le bouton d'envoi, les données du formulaire seront envoyées au fichier "welcome.php":



- "welcome.php" ressemble à ceci:
- ```
<html>
<body>
Bienvenue <?php echo $_POST ["fname"];?>!

Vous avez <?php echo $_POST ["age"];?> Ans.
</ body>
</ html>
```

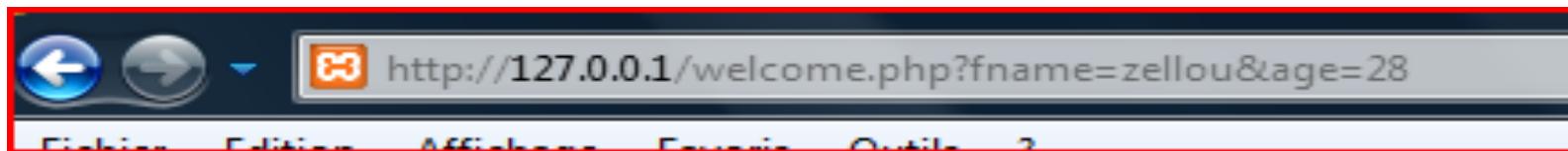
Ce qui affiche :  
Bienvenue Ahmed !  
Vous avez 28 ans.

# La Fonction \$\_GET

- \$\_GET est utilisé pour récupérer les valeurs des formulaire.
- Méthode limitée : ?????
- Exemple

```
<form action="welcome.php" method="get">
Nom: <input type="text" name="fname"/>
Âge: <input type="text" name="age"/>
<input type="submit"/>
</form>
```

- Après validation:



- Les noms des champs du formulaire seront utilisés comme des clés dans le tableau associatif \$\_GET.

```
Bienvenue <?php echo $_GET ["fname"];?>.

Vous avez <?php echo $_GET ["age"];?> Ans
```

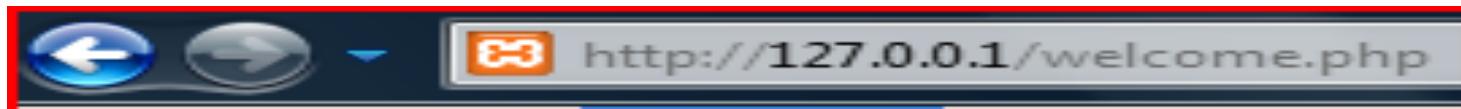
# La Fonction \$ \_POST

---

- \$ \_POST est utilisée pour récupérer les valeurs d'un formulaire qui utilise la méthode = "post".
- Les données sont invisibles.
- Aucune limite sur la quantité d'informations à envoyer.
- Taille maximale de 8 Mo, par défaut (variable post\_max\_size du fichier php.ini).
- Exemple

```
<form action="welcome.php" method="post">
Nom: <input type="text" name="fname" />
Âge: <input type="text" name="age" />
<input type="submit" />
</ form>
```

- Après validation:



# La Fonction \$\_POST

---

- Le fichier welcome.php peut utiliser \$\_POST pour récupérer les données de formulaire.
- Les noms des champs seront utilisés comme des clés dans le tableau \$\_POST.

Bienvenue <?php echo \$\_POST ["fname"];?>! <br/>  
Vous êtes <?php echo \$\_POST ["age"];?> Ans.

- PHP intègre aussi d'autres méthodes : \$\_PUT et \$\_DELETE.

# SII Server Side Includes

---

- Pour insérer le contenu d'un fichier PHP dans un autre : **include()** et **require()**.
- Les deux fonctions sont identiques sauf dans la gestion des erreurs:
  - include() génère un avertissement, mais le script continue son exécution
  - require() génère une erreur fatale, et le script s'arrête.
- Utilisez aussi **include\_once()** et **require\_once()**. Exemple : ?
- Exemple

```
<?php include("Entete.php");
require("Fichier_erreurs.php");
?>
```

# Gestion de fichiers

- La fonction **fopen()** est utilisé pour ouvrir les fichiers en PHP.
- Paramètres : nom\_du\_fichier et mode.

```
<? php
$file = fopen("file.txt", "r");
?>
```

- Si la fonction fopen() est incapable d'ouvrir le fichier, elle retourne 0 (faux).

Mode	Description
r	Lecture seule. Commence au début du fichier
r+	Read/Write. Commence au début du fichier
w	Ecriture seulement. Ouvre et efface le contenu du fichier, ou crée un nouveau fichier s'il n'existe pas
w+	Read/Write. Ouvre et efface le contenu du fichier, ou crée un nouveau fichier s'il n'existe pas

# Gestion de fichiers

Mode	Description
a	Ouvre et écrit à la fin du fichier ou crée un nouveau fichier s'il n'existe pas
a+	Préserve le contenu des fichiers en écrivant à la fin du fichier
x	Ecriture seulement. Crée un nouveau fichier. Retourne FALSE et une erreur si le fichier existe déjà
x+	Read/Write. Crée un nouveau fichier. Retourne FALSE et une erreur si le fichier existe déjà

- Exemple
- L'exemple suivant génère un message si la fonction fopen() est incapable d'ouvrir le fichier spécifié:

```
<?php
$file = fopen("file.txt", "r") or exit ("Impossible d'ouvrir le fichier!");
?>
```

# Gestion de fichiers

---

- Lecture d'un fichier ligne par ligne
  - `fgets()` est utilisée pour lire une seule ligne à partir d'un fichier.
  - Après un appel à cette fonction le pointeur se déplace à la ligne suivante.

```
<?php
$file = fopen("file.txt", "r") or exit ("Impossible d'ouvrir le fichier!");
echo fgets($file). "
";
?>
```

- Lecture d'un fichier caractère par caractère
  - `fgetc()` est utilisé pour lire un seul caractère d'un fichier.
  - Après un appel à cette fonction le pointeur se déplace au caractère suivant.

```
<?php
$file = fopen("file.txt", "r") or exit ("Impossible d'ouvrir le fichier!");
echo fgetc($file);
?>
```

# Gestion de fichiers

---

## ■ Fermeture d'un fichier

- `fclose()` est utilisée pour fermer un fichier ouvert:

```
<?php
$file = fopen("file.txt", "r");
// code à exécuter
fclose ($file);
?>
```

## ■ Tester la fin d'un fichier

- `feof()` test si (EOF) a été atteint, utile pour boucler sur des données de longueur inconnue.
- Exemple : `if (feof($file)) echo "Fin de fichier";`
- Exemple complet

```
<?php
$file = fopen("file.txt", "r") or exit ("Impossible d'ouvrir le fichier!");
while (!feof($file)) echo fgets($file). "
";
fclose ($file);
?>
```

# Gestion de fichiers

---

## ■ La Fonction fwrite()

- Permet d'écrire dans un fichier ouvert.
- Retourne le nombre d'octets écrits ou FALSE en cas d'échec.
- **Syntaxe :**
- **fwrite (fichier, chaîne, longueur)**
  - fichier : obligatoire, indique le fichier
  - chaîne : obligatoire, spécifie la chaîne à écrire dans le fichier
  - Longueur : optionnelle, indique le nombre maximal d'octets à écrire

## ■ Exemple

```
<?php
$file = fopen("file.txt", "w");
echo fwrite($file, "Bonjour le monde.");
fclose($file);
?>
```

Sortie:

21

# Gestion des erreurs

- Dans certains cas, il n'est ni possible ni utile de poursuivre l'exécution du code PHP (variables non définies, valeurs erronées, échec de connexion, ...).

- Arrêt brutal de l'exécution du code:
  - die(message)
  - exit(message)

Envoie message au navigateur et termine l'exécution du script.

Constante
E_ERROR
E_WARNING
E_PARSE
E_NOTICE
E_CORE_ERROR
E_CORE_WARNING
E_COMPILE_ERROR
E_COMPILE_WARNING
E_USER_ERROR
E_USER_WARNING
E_USER_NOTICE
E_ALL
E_STRICT

# Gestion des erreurs

- Sur un serveur en production, toute erreur affichée donne des indices sur les scripts et rend le site vulnérable.

```
$v = file("dummy.txt") ;
```

**Warning:** file(dummy.txt): failed to open stream: No such file or directory in **dummy.php** on line **68**

- Plutôt :

```
$v = @file("dummy.txt") or die("Problème de lecture") ;
```

Problème de lecture

# Atelier

---

- Eriez un script "Compteur.php" qui compte le nombre des accès à une page via les fonctions de lecture-écriture.
- Comptez aussi le temps passé dans toutes les pages.

# File Upload

- Pour permettre aux utilisateurs de charger des fichiers à partir d'un formulaire:

```
<html>
<body>
<form action="upload_fichier.php" method="post"
enctype="multipart/form-data">
 <label for="fichier"> Nom du fichier:</label>
 <input type="file" name="fichier" />

 <input type="submit" value=" Valider " />
</form>
```

- L'attribut *enctype* spécifie le format du fichier à charger.  
"multipart/form-data" est utilisé pour les données binaires.
- *type = "file"* précise que l'entrée doit être traitée comme un fichier.
- Permettre aux utilisateurs de charger des fichiers est un gros risque de sécurité. Pourquoi : ?
- Autoriser seulement les utilisateurs de confiance.

# File Upload

---

## ■ Script d'upload

```
<?php
if ($_FILES["fichier"]["error"] > 0)
 echo "Error: " . $_FILES["fichier"]["error"] . "
";
else
{
 echo "Upload: " . $_FILES["fichier"]["name"] . "
";
 echo "Type: " . $_FILES["fichier"]["type"] . "
";
 echo "Taille: " . ($_FILES["fichier"]["size"] / 1024) . " Kb
";
 echo "Enregistré dans " . $_FILES["fichier"]["tmp_name"];
}
?>
```

## ■ En utilisant un tableau **\$\_FILES** :

- **\$\_FILES** [**"fichier"**] [**"name"**] - le nom du fichier
- **\$\_FILES** [**"fichier"**] [**"type"**] - le type du fichier
- **\$\_FILES** [**"fichier"**] [**"size"**] - la taille en octets du fichier

# File Upload

---

- En utilisant un tableau `$_FILES` :
  - `$_FILES["fichier"]["tmp_name"]` - le nom de la copie temporaire du fichier stocké sur le serveur
  - `$_FILES["fichier"]["error"]` - le code d'erreur lié à l'upload du fichier
- Pour des raisons de sécurité, vous devez ajouter des restrictions sur le fichier à télécharger.
- Dans ce script, on ajoute quelques restrictions sur le format (.gif, .jpeg ou .pjpeg) et la taille (inférieur à 20 ko):

```
<?php
if ((($_FILES["fichier"]["type"] == "image/gif")
|| ($_FILES["fichier"]["type"] == "image/jpeg")
|| ($_FILES["fichier"]["type"] == "image/pjpeg"))
&& ($_FILES["fichier"]["size"] < 20000))
{
 if ($_FILES["fichier"]["error"] > 0)
 echo "Error: " . $_FILES["fichier"]["error"] . "
";
```

# File Upload

---

```
else
{
 echo "Upload: " . $_FILES["fichier"]["name"] . "
";
 echo "Type: " . $_FILES["fichier"]["type"] . "
";
 echo "Taille : " . ($_FILES["fichier"]["size"] / 1024) . " Kb
";
 echo "Enregistré dans : " . $_FILES["fichier"]["tmp_name"];

 $tmp_name = $_FILES["fichier"]["tmp_name"];
 $name = $num.$_FILES["fichier"]["type"];
 move_uploaded_file($tmp_name, $name);
}
}

else
{
 echo " fichier non valide ";
}
?>
```

- Quel Risque : ?
- Note: IE reconnaît les fichiers jpg avec pjpeg, FireFox avec jpeg.

# Mail

- La fonction **mail()** est utilisée pour envoyer des courriels à partir d'un script.
- **mail(to, subject, message, headers, parameters)**

Paramètre	Description
to	Obligatoire. Spécifie le récepteur / récepteurs de l'email
subject	Obligatoire. Précise l'objet de l'email.
message	Obligatoire. Définit le message qui sera envoyé. Chaque ligne doit être séparée par un caractère LF (\n). Lignes ne devraient pas dépasser 70 caractères
headers	Facultatif. Indique des entêtes supplémentaires, comme From, Cc et Cci. Les en-têtes supplémentaires doivent être séparés par un CRLF (\r \n)
parameters	Facultatifs. Spécifie un paramètre supplémentaire au programme sendmail

# Mail

---

- Nécessite l'installation d'un système e-mail.
- Le programme à utiliser est défini par les paramètres de configuration dans le fichier php.ini.
- Simple E-Mail

```
$to = "personne@example.com";
$to .= ",personne2@example.com";
$subject = "Mail de test";
$message = "Bonjour, ceci est un email de rappel de notre RDV.";
$message .= "Je vous propose le Vendredi 12 Février à 16H00 \n";
$message .= "--\r\n";
$message .= " OK ? ";
$from = "envoyeur@example.com";
$headers = "From: $from";
mail($to,$subject,$message,$headers);
```

# Cookies

---

- Un cookie (biscuit) est utilisé pour identifier un utilisateur.
- C'est un petit fichier que le serveur donne à l'ordinateur de l'utilisateur.
- Chaque fois que l'ordinateur demande une page avec le navigateur, il enverra le cookie aussi.
- Comment créer un cookie?
  - La fonction **setcookie()** est utilisée pour définir un cookie.
  - Elle doit apparaître avant la balise <html>.  
**setcookie(name, value, expire, path, domain);**
  - Exemple 1 : L'exemple suivant, crée un cookie appelé "user" et assigne au cookie la valeur "ZELLOU".
  - Ce cookie doit expirer au bout d'une heure:  
**<?php  
setcookie ("user", "ZELLOU", time () +3600);  
?>  
<html> .....**

# Cookies

---

- Le tableau **`$_COOKIE`** est utilisé pour récupérer la valeur d'un cookie.
- L'exemple suivant, récupère la valeur du cookie nommé "user" et l'affiche sur une autre page:

```
<?php
echo $_COOKIE["user"];
print_r($_COOKIE);
?>
```

- **`isset()`** est utilisé pour vérifier l'existence du cookie.

```
<?php
if (isset($_COOKIE["user"]))
 echo "Welcome " . $_COOKIE["user"] . "!" .
;
else echo " Bienvenue invité!" .
;
?>
```

# Cookies

---

- Comment effacer un cookie?

- Pour supprimer un cookie, mettez une date d'expiration déjà passée.
  - Exemple:

```
<?php
setcookie("user", "",);
?>
```

# Atelier

---

- Ecrivez un programme dans lequel, vous affichez la liste des mots clés recherchés par un utilisateur.

# Atelier

---

```
if(isset($_GET['kw']))
{ if(!isset($_COOKIE['kws']))
 {$kws=array($_GET['kw']);
 $kws=serialize($kws); setcookie ("kws",$kws);}
 else { $kws=unserialize($_COOKIE["kws"]);
 array_push($kws, $_GET['kw']);
 $kws=serialize($kws);
 setcookie ("kws",$kws); }
}
if(isset($_COOKIE["kws"]))
{ $kws=unserialize($_COOKIE["kws"]);
foreach ($kws as $kw) echo $kw; }
?>
<form> key word<input name="kw"> <input type="submit" name="search"></form>
```

# Sessions

---

- Une variable de session est utilisé pour stocker des informations ou modifier les paramètres d'une session utilisateur.
- Elles détiennent l'information sur un seul utilisateur, et elles sont valables pour toutes les pages dans une seule application.
- Démarrer une session PHP avec la fonction `session_start()`
- Elle doit être antérieure à la balise `<html>`:  
`<?php session_start(); ?>`  
`<html>`
- Ce code va créer une session pour l'utilisateur (assigner un UID pour cette session utilisateur).

# Sessions

---

- Stockage d'une variable de session
- Pour stocker et récupérer les variables de session, utiliser la variable `$_SESSION`:

```
<?php
session_start();
// Stocker les données de session
$_SESSION['Vue']=1;
?
<html>
<body>
<?php
//Récupère les données de session
echo "Page Vue=". $_SESSION['Vue'];
?
</body>
</html>
```

Sortie:  
Page Vue=1

# Sessions

---

- L'exemple suivant, crée un compteur de pages visités.
  - isset() vérifie si la variable "vue" est déjà définie.
  - Si la variable "vue" est définie, on incrémente notre compteur. Sinon, on crée la variable et on l'initialise à 1.

```
<?php
session_start();
if(isset($_SESSION['vue']))
$_SESSION['vue']=$_SESSION['vue']+1;
else
$_SESSION['vue']=1;
echo "Vue=". $_SESSION['vue'];
?>
```

# Sessions

---

## ■ Détruire une session

- Utiliser la fonction **unset()** ou **session\_destroy()**.
- La fonction **unset()** est utilisée pour libérer une variable de session.

```
<?php
unset($_SESSION['vue']);
?>
```

- **session\_destroy()** permet de détruire carrément la session.
- Avec **session\_destroy()**, vous perdrez toutes les données stockées dans la session.

```
<?php
session_destroy();
?>
```

# Etude de Cas

## ■ Gestion de Panier de commandes :

- L'objectif est de réaliser un petit site de e-commerce qui permet à des utilisateurs de commander des articles (stockés dans plusieurs fichiers php).
- Nous supposons que tous les clients qui accèdent au site sont inscrits.

The diagram illustrates the flow of a purchase through four stages:

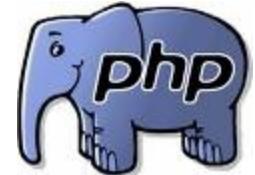
- Login Screen:** A green window with fields for "Login" and "Password", and a "Login" button.
- Book Preview:** An image of a book titled "A LA CONQUÊTE DES GRANDS FONDS".
- Product Information:** Text showing the book's price as "Prix public : 28,00 €", its price on EYROLLES.COM as "Prix EYROLLES.COM 26,60 €", and a red price tag icon indicating the discounted price of "26,60 €". Below this is a link "[> ajouter au panier](#)" (Add to cart).
- Shopping Cart:** A table titled "Mon panier" showing three items:
 

Article	Prix
A la conquête des grands fonds Techniques d'étude de la géologie marine	26,60 €
Droit de la consommation	38,00 €
Indignez vous !	2,85 €

 The cart also includes a note about free shipping for orders over 49€, a breakdown of totals (Total avant réduction: 71,00 €, Economie: 3,56 €, Total: 67,45 €), and links to update or clear the cart.

# Quelques Variables

Fonction	Description
\$argc	Nombre d'arguments passés au script
\$argv	Tableau d'arguments passés au script
\$PHP_SELF	Le nom du fichier contenant le script
\$HTTP_COOKIE	Tableau associatif des variables passées en cookies
\$HTTP_GET_VARS	Tableau associatif des variables
\$HTTP_GET	Tableau associatif des variables passées via la méthode GET
\$HTTP_POST	Tableau associatif des variables passées via la méthode POST
\$_FILES	Tableau associatif des variables des fichiers attachés
\$HTTP_ENV	Tableau associatif des variables d'environnement
\$_SERVER	Tableau associatif des variables serveurs
\$HTTP_SESSION_	Tableau associatif des variables de sessions
\$_REQUEST	Tableau associatif de toutes les variables : GET, POST et Cookie



---

# Développement des Applications Web

## Interaction MySQL

Ahmed ZELLOU

IWIM, eMBI, ENSIAS, 2019-2020.

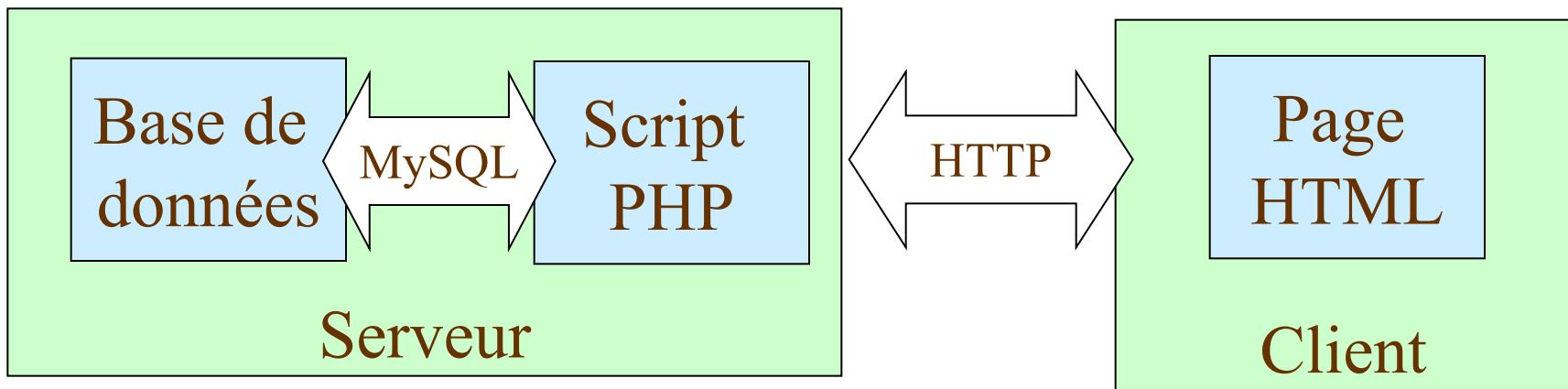
# Plan

---

- Connexion au serveur MySQL
- Sélection d'une base de données
- Insertion dans une base MySQL
- Sélection dans une base MySQL
- Afficher le résultat au format HTML
- Suppression dans une base MySQL
- Modification dans une base MySQL
- Ateliers

# Connexion au serveur MySQL

- MySQL dérive directement de SQL (Structured Query Language).
- Passage vers la version MariaDB.
- Le serveur MySQL est souvent utilisé avec le langage PHP.



# PhpMyAdmin

---

- Une interface Web intuitive pour l'administration de MySQL, écrite en PHP.
- Téléchargeable sur <http://phpmyadmin.sourceforge.net>
- Cet outil permet de :
  - créer de nouvelles bases
  - créer/modifier/supprimer des tables
  - afficher/ajouter/modifier/supprimer des tuples dans des tables
  - Recherches / Requêtes / Importation / exportation de tables / base / données / Optimisation de tables / base
  - gérer les privilèges des utilisateurs
  - Moteurs : MyISAM : plain text, très rapide, Moteur INNODB :transactionnel, ACID, moins rapide
  - ....

# PhpMyAdmin

Screenshot of the PhpMyAdmin interface running on a Windows operating system.

The browser address bar shows: localhost/phpmyadmin/index.php?db=mit&token=86002e39d0674a98e3b611ae00a34c04

The left sidebar lists databases:

- cdcol (1)
- information\_schema (37)
- issam (32)
- mit (1)
- mysql (24)
- performance\_schema (17)
- phpmyadmin (8)
- serie4 (3)
- test
- webauth (1)

A message at the bottom left says: "Choisissez une base de données" (Select a database).

The main content area includes:

- Actions**: MySQL localhost section with "Créer une base de données" button and dropdowns for collation and character set.
- Interface**: Language (Français - French), Theme (Original), Color choice (Réinitialiser), and Text size (82%).
- MySQL**: Server information (localhost via TCP/IP, version 5.5.8, protocol 10, user root@localhost), character set (UTF-8 Unicode (utf8)).
- Serveur web**: Apache/2.2.17 (Win32) mod\_ssl/2.2.17 OpenSSL/0.9.8o PHP/5.3.4 mod\_perl/2.0.4 Perl/v5.10.1, MySQL version 5.0.7-dev, PHP extension mysql.
- phpMyAdmin**: Version 3.3.9, Documentation, Wiki, Site officiel, ChangeLog, Git, Lists.
- System tray icons**: Windows, Firefox, Google Chrome, File Explorer, Control Panel, Paint, Task View, and Date/Time (14:45, 10/02/2016).

Two error messages are displayed at the bottom:

- "Certaines fonctionnalités ayant trait aux tables reliées sont désactivées. Pour une analyse du problème, cliquez ici." (Some features related to linked tables are disabled. Click here to analyze the problem.)
- "Votre fichier de configuration fait référence à l'utilisateur «root» sans mot de passe, ce qui correspond à la valeur par défaut de MySQL. Votre serveur MySQL est donc ouvert aux intrusions, et vous devriez corriger ce problème de sécurité." (Your configuration file refers to the 'root' user without a password, which corresponds to the default value of MySQL. Your MySQL server is therefore open to intrusions, and you should correct this security issue.)

# Connexion au serveur MySQL

- Connexion s'effectue avec la fonction **mysqli\_connect()**.
- Syntaxe : **mysqli\_connect(Serveur, Utilisateur, Password);**

Paramètre	Description
nom_du_serveur	facultatif. Spécifie le serveur où on va se connecter. La valeur par défaut est "localhost: 3306"
nom_utilisateur	facultatif. Spécifie le nom d'utilisateur qui va se loger. La valeur par défaut est le nom de l'administrateur de la base MySQL.
mot_de_passe	facultatif. Spécifie le mot de passe de connexion. Par défaut est ""

- Deux types de connexions :
  - non-persistantes **mysqli\_connect()**, se referme automatiquement à la fin du script.
  - persistantes **mysqli\_pconnect()**, ne se referme pas automatiquement à la fin du script
- Pour fermer la connexion : **mysqli\_close(\$desc\_Conexion)**<sup>81</sup>

# Connexion au # Serveurs

---

- Plusieurs types de connexions :
  - `mysql_connect("serveur","utilisateur","passe");`
  - `mssql_connect("serveur","utilisateur","passe");`
  - `ociLogon("utilisateur","passe", "base");`
  - `pg_connect("dbname=nom_base,host=nom_serveur port=num_port ";"user=utilisateur; password=passe");`
  - `sybase_connect("serveur","utilisateur","passe");`
- Pour fermer la connexion : `typeBase_close($desc_Conexion)`

# Informations de la connexion

---

- Plusieurs fonctions PHP à propos de la connexion en cours
  - `mysqli_get_client_info();`
  - `mysqli_get_host_info($id_connexion);`
  - `mysqli_get_proto_info($id_connexion);`
  - `mysqli_get_server_info($id_connexion);`
  - `pg_host($id_connexion);`
  - `pg_options($id_connexion);`
  - `pg_port($id_connexion);`
  - `pg_client_encoding($id_connexion);`

# Sélection d'une base de données

- **mysqli\_select\_db()** pour sélectionner la base de données.  
Retourne TRUE en cas de succès, ou FALSE en cas d'échec.
- Syntaxe : **mysqli\_select\_db (base\_de\_donnees, connexion)**

Paramètre	Description
base_de_donnees	Obligatoire. Spécifie la base de données à sélectionner.
connexion	Optionnel. Spécifie la connexion MySQL. S'il n'est pas spécifié, la dernière connexion ouverte avec la fonction mysqli_connect () sera utilisée.

- Pour les autres serveurs :
  - **mysqli\_select\_db, msql\_select\_db, sybase\_select\_db, pg\_connect, ...**

# Informations de la connexion

---

- Pour créer une bases de données :
  - `mysqli_create_db ($nom_base_donnee, $id_connexion);`
  - `msql_create_db($nom_base_donnee,$id_connexion);`
- Pour supprimer une bases de données :
  - `mysqli_drop_db ($nom_base_donnee, $id_connexion);`
  - `msql_drop_db($nom_base_donnee, $id_connexion);`
- Une bonne pratique :

```
function connexion($sgbdr, $hote, $port, $utilisateur,
$mot_passe,$param_sup)
{ avec des switch() }
```

# Informations de la connexion

---

- Liste des bases de données : `mysqli_list_dbs($id_connexion);`
- Liste de tables : `mysqli_list_tables($nom_base_donnee, $id_connexion);`
- Le nom d'un champ : `mysqli_field_name($id_resultat, $num_champ);`
- Le type d'un champ : `mysqli_field_type($id_resultat, $num_champ);`
- La longueur d'un champ : `mysqli_field_len($id_resultat, $num_champ);`
- Détails d'un champs (NOT NULL, PRIMARY, ...) : `mysqli_field_flags($id_resultat, $num_champ);`

# Informations de la connexion

---

- Libération des ressources : `mysqli_free_result($id_resultat);`
- Numéro et message d'erreur due à la dernière action :  
`mysqli_error([$id_connexion]);`
- Les fonctions de connexions :
  - `connection_aborted()` : vérifie si le client a abandonné la connexion.
  - `connection_status()` : retourne le statut de la connexion.
  - `connection_timeout()` : vérifie l'expiration du script en cours.

# Insertion dans une base MySQL

---

- L'instruction **INSERT INTO** permet d'insérer de nouveaux enregistrements dans une table selon deux formats.
  - Format 1 : **INSERT INTO nom\_table VALUES (valeur1, valeur2, valeur3 ,...)**
  - Format 2 : **INSERT INTO nom\_table (colonne1, colonne2, colonne3 ...) VALUES (valeur1, valeur2, valeur3 ,...)**
- Pour exécuter une requête : **mysqli\_query(\$req).**
- Cette fonction est utilisée pour envoyer une requête ou une commande pour une connexion MySQL.

# Insertion dans une base MySQL

---

## ■ Exemple :

```
<?php
define("HOST","host"); define("USER","user"); define("PASS", "pass");
define("BASE","labase");
$con = mysqli_connect(HOST, USER, PASS);
if (!$con) { echo "Connexion impossible: ".mysqli_error(); }

mysqli_select_db("BASE", $con);

$requete= "INSERT INTO Personnes (Nom, Prenom, Age)
VALUES ('ZELLOU', 'AHMED', '30')";

mysqli_query ($requete, $con);
mysqli_close($con);
?>
```

# Sélection dans une base MySQL

---

- L'instruction SELECT est utilisé pour sélectionner les données à partir d'une base de données.
- La fonction **mysqli\_query()** retourne, dans ce cas, le résultat d'exécution de cette requête.
- La fonction **mysqli\_fetch\_row()** permet d'organiser le résultat de la requête fourni par la fonction **mysqli\_query()** sous forme d'une matrice.
- Utilisez aussi **mysqli\_fetch\_array()**

# Sélection dans une base MySQL

---

- Chaque ligne de la matrice contienne un seul enregistrement.
- La fonction retourne FALSE en cas d'échec ou quand il ya pas d'enregistrements repenant à la requête.
- Syntaxe :
  - **mysqli\_fetch\_row(data)**
  - data spécifie le pointeur de données à utiliser.
- Une fois les données récupérées, cette fonction se déplace à la ligne suivante dans le jeu d'enregistrements après chaque lecture.

# Sélection dans une base MySQL

---

- Exemple

```
<?php
 require 'connexion.php';
 $condition="zellou";
 $requete = "SELECT * FROM personne WHERE 'Nom' = $condition ";
 $resultat = mysqli_query ($requete, $con);
 $donnees= mysqli_fetch_row ($result);
 echo"$donnees[0] "; echo"$donnees[1] "; echo"$donnees[2] ";
 mysqli_close($con);
?>
```

- La fonction `mysqli_fetch_array()` retourne la première ligne du jeu d'enregistrements dans un tableau associatif.

# Sélection dans une base MySQL

---

- La boucle while boucle sur tous les enregistrements dans le jeu d'enregistrements.
- Exemple

```
<?php
$con = mysqli_connect("localhost","user","motdepasse");
if (!$con)
{ echo "Connexion impossible: ".mysqli_error(); }
mysqli_select_db("mabase", $con);
$requete = "SELECT * FROM personne WHERE 1 ";
$resultat = mysqli_query ($requete, $con);
while($ligne = mysqli_fetch_array($resultat))
{ echo $ligne['Nom'] . " " . $ligne ['Prenom']; }
mysqli_close($con);
?>
```

# Afficher le résultat au format HTML

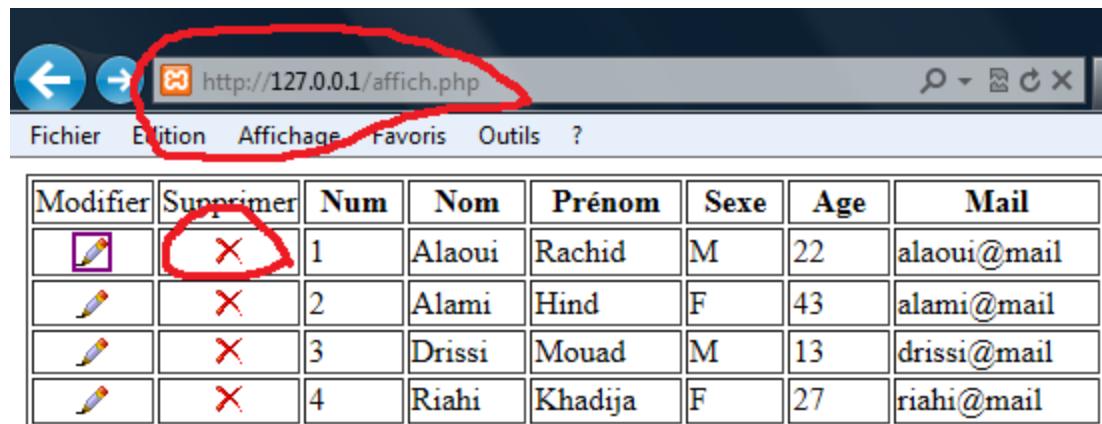
---

- ```
$con = mysqli_connect("localhost", "user", "motdepasse");
    { echo "Connexion impossible: ".mysqli_error(); }
mysqli_select_db("mabase", $con);
$requete = "SELECT * FROM personne WHERE 1 ";
$resultat = mysqli_query ($requete, $con);

echo "<table border='1'>
<tr><th> Nom </th> <th> Prénom </th> </Tr> ";
while($row = mysqli_fetch_array($result))
{
    echo "<tr>";
    echo "<td>" . $row['Nom'] . "</td>";
    echo "<td>" . $row['Prenom'] . "</td>";
    echo "</tr>";
}
echo "</table>";
mysqli_close ($con);
```

Suppression dans une base MySQL

- Lister tous les enregistrements sur "affich.php" sous forme d'un tableau.



Modifier	Supprimer	Num	Nom	Prénom	Sexe	Age	Mail
		1	Alaoui	Rachid	M	22	alaoui@mail
		2	Alami	Hind	F	43	alami@mail
		3	Drissi	Mouad	M	13	drissi@mail
		4	Riahi	Khadija	F	27	riahi@mail

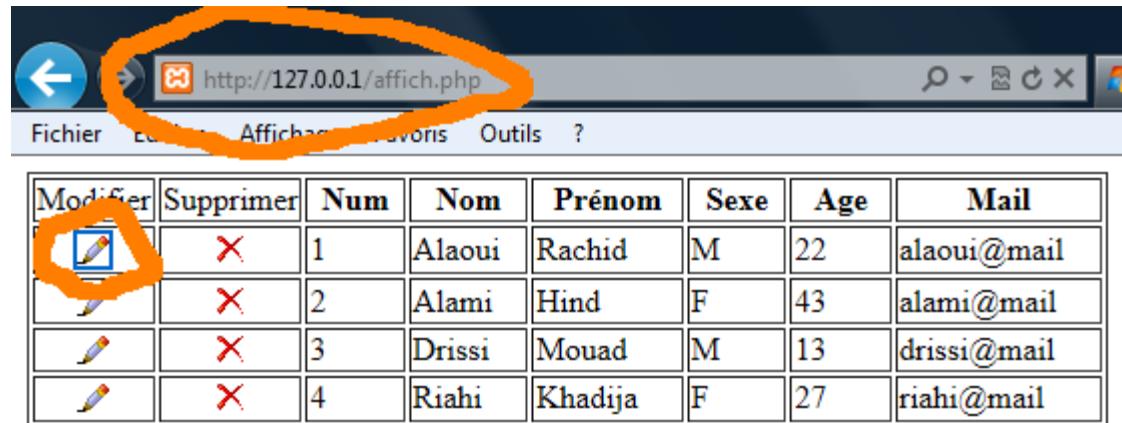
- Dans le fichier de suppression "delete.php"
- Récupération du numéro \$numero de l'enregistrement à modifier. \$numero=\$_GET['num']
- Connexion au serveur et sélection de la Base

Suppression dans une base MySQL

- Crédit de la requête.
 - \$requete="delete from table where clé=\$numero"
- Exécution de la requête avec
\$resultat=mysqli_query(\$requete)
- Retour vers la page d'affichage
header('Location: affich.php');

Modification dans une base MySQL

- Lister tous les enregistrements sur "affich.php" sous forme d'un tableau.



The screenshot shows a web browser window with the URL `http://127.0.0.1/affich.php` highlighted by a red oval. Below the address bar is a menu bar with French labels: Fichier, Ed., Affichage, Favoris, Outils, ?.

Modifier	Supprimer	Num	Nom	Prénom	Sexe	Age	Mail
	X	1	Alaoui	Rachid	M	22	alaoui@mail
	X	2	Alami	Hind	F	43	alami@mail
	X	3	Drissi	Mouad	M	13	drissi@mail
	X	4	Riahi	Khadija	F	27	riahi@mail

- Rajouter dans chaque ligne du tableau :

- `<td><a href="update.php?num=<?php echo $donnee[0];?>"></td>` un lien vers la page update.php avec image cliquable b_edit.png 
- `<td><a href="delete.php?num=<?php echo $donnee[0];?>"></td>` un lien vers la page delete.php avec image cliquable b_drop.png 

Modification dans une base MySQL

- Dans le fichier de modification "update.php"
- Récupération du numéro \$numero de l'enregistrement à modifier. \$numero=\$_GET['num']
- Connexion au serveur et sélection de la Base
- Récupération de la l'enregistrement à modifier :
\$requete="select * from table where clé=\$numero"
- Exécution de la requête avec
\$resultat=mysqli_query(\$requete)
- Récupération de la ligne avec
\$donnee=mysqli_fetch_row(\$resultat)

Modification dans une base MySQL

- Affichage du formulaire avec les données de l'enregistrement à modifier déjà saisie.

The screenshot shows a web browser window with the URL `http://127.0.0.1/update.php?num=1` in the address bar. A red oval highlights the address bar area. Below the address bar is a menu bar with French labels: Fichier, Edition, Affichage, Favoris, Outils, and ?. The main content area is titled "Formulaire de Modification". It contains a table with six rows, each representing a field and its value:

Formulaire de Modification	
Num	1
Nom	Alaoui
Prénom	Rachid
Sexe	M <input checked="" type="radio"/> F <input type="radio"/>
Age	22
Mail	alaoui@mail

At the bottom of the form are two buttons: "Modifier" (in blue) and "Annuler".

- `<input type="text" name="nom" value="<?php echo $donnee[1]?>">` Pour afficher le nom dans le champ de texte face au Nom.
- `<input type="text" name="num" value="<?php echo $donnee[0]?>" disabled>` Pour afficher le numéro dans le champ de texte face au Num.
- `disabled` pour que le champ ne soit pas modifiable.

Modification dans une base MySQL

- L'action du fichier "update.php" pointe sur "valideUpdate.php"
- Dans "valideUpdate.php"
- Récupération des données de l'enregistrement à modifier.
 - \$numero=\$_POST['num']
 - \$nom=\$_POST['nom']
 - \$prenom=\$_POST['prenom']
 - \$sexe=\$_POST['sexe']
 - \$age=\$_POST['age']
 - \$mail=\$_POST['mail']
- Connexion au serveur et sélection de la Base

Modification dans une base MySQL

- \$nom=\$_POST['nom']
- \$prenom=\$_POST['prenom']
- \$sexe=\$_POST['sexe']
- \$age=\$_POST['age']
- \$mail=\$_POST['mail']
- Crédit de la requête.
 - \$requete="update table set nom='\\$nom',
prenom='\\$prenom', sexe='\\$sexe', age=\\$age,
mail='\\$mail' where clé=\\$numero"
- Exécution de la requête avec
\$resultat=mysqli_query(\$requete)
- Retour vers la page d'affichage
header('Location: affich.php');

Merci