

# Formation PHP/MySQL

Pierre PATTARD

Julien BENOIT

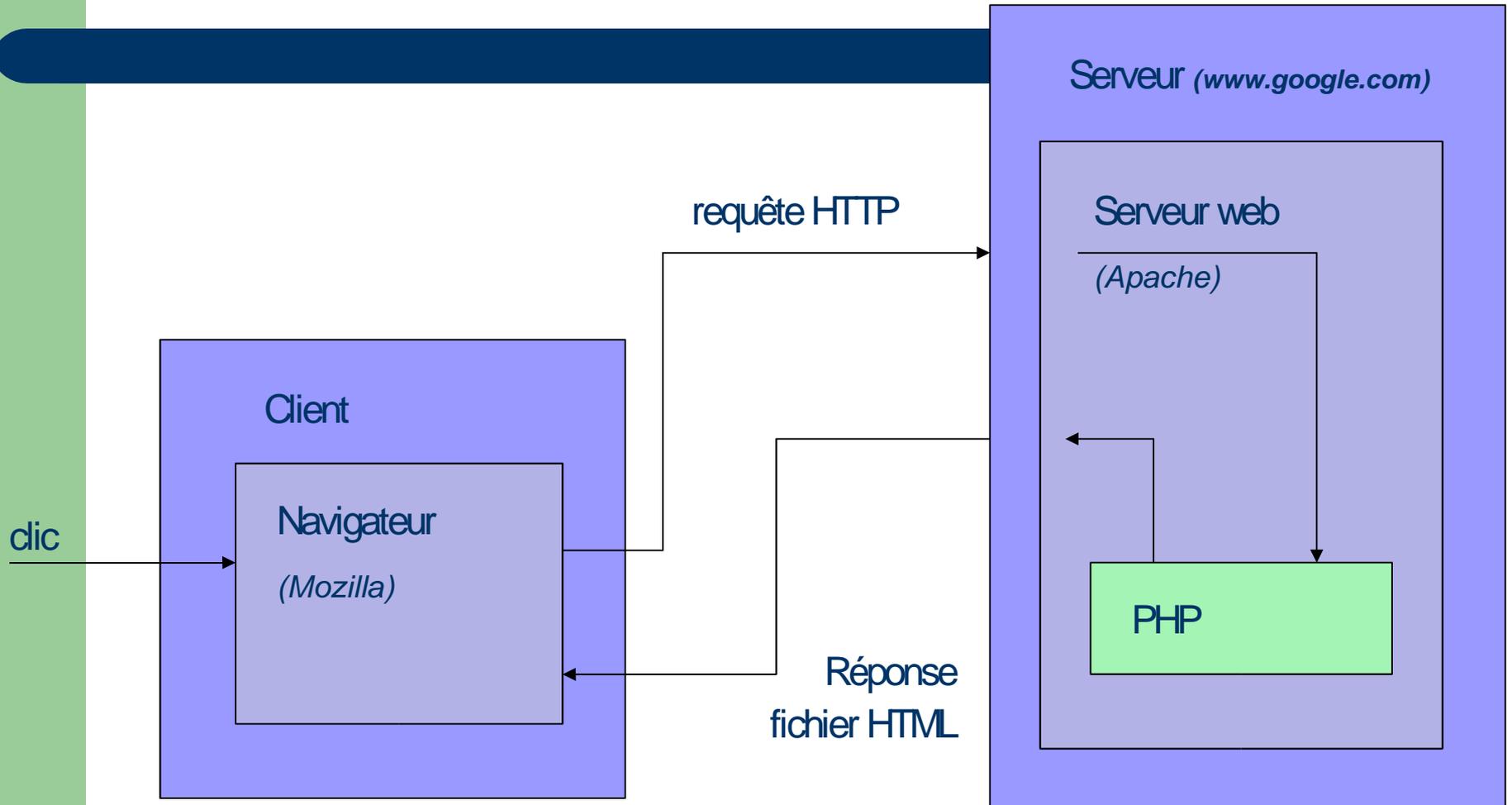


Avril 2005

# Le PHP

- Langage interprété, côté serveur
  - pages interprétées par le serveur web  
pas de compilation, code éditable avec un bloc notes.
  - aucune différences pour le visiteur (client)  
le client ne voit que du HTML

# Le PHP



# Le PHP: Avantages

- *Avantages:*
  - pas besoin de différencier les navigateurs du marché (le code fonctionne sur tous dès qu'il fonctionne sur un)
  - maîtrise du fonctionnement et du code
  - le code source n'est pas visible
  - nombreuses interactions avec le serveur (bases de données, fonctions avancées d'images, de génération de pdf, d'exécution de scripts)

# Le PHP: inconvénients

- Nécessite un serveur qui accepte le PHP
- Nombreuses failles de sécurité (revers de la médaille de sa puissance)

# La syntaxe de base

- Instructions insérées par les balises `<?php` et `?>` dans un fichier qui a la bonne extension (.php, .php3)
- Chaque instruction se termine par un `;"`  
ex: `print "Hello World!";`
- Les variables ne sont ni déclarées ni typées et leur nom commence par un `$`  
ex: `$foo="toto";`
- Commentaires: `//` et `/* ... */` pour plusieurs lignes

# Les structures conditionnelles

- `If ( condition ) { instructions }`  
`elseif ( condition2 ) { instructions }`  
`else { instructions }`

ex: `if($i<2){ print " i est inférieur à 2"; }`  
`elseif((($i>=2) and ($i<4)) { print "i est compris entre 2 et 4"; }`  
`else { print "i est supérieur à 4"; }`

# Les structures itératives

- `for(initialisation;condition;incrémentation)`  
`{instructions}`

ex: `for($i=0;$i<=20;$i++) { print "i est égale à $i \n"; }`

rmq: `print "i est égale à $i \n"`  $\Leftrightarrow$  `print "i est égale à ".$i."\n"`

$\Leftrightarrow$  `print "i est égale à";`

`print $i;`

`print "\n";`

- `While ( condition ) { instructions }`

ex: `while($i<20) { ...`

`...`

`$i++;`

`}`

# Les tableaux

- `$tableau=array("0","1", ...);`  
`$tableau[0] → premier élément du tableau`
- Tableaux associatifs  
`$tableau=array(nom=>"dupont",prenom=>roger" ...),`  
`print $tableau->nom; // retourne "dupont"`

# Parcourir un tableau

- Tableau "normal":

```
for($i=0;$i<=length($tableau);$i++)  
    { print $tableau[$i]; }
```

- Tableau associatif:

```
foreach($tableau as $cle=>$valeur)  
    { print "La clé $cle a pour valeur $valeur";}
```

# Les variables spéciales:

- `$_POST` et `$_GET` permettent de récupérer la sortie d'un formulaire

ex: `$login=$_POST["login"];`  
`$motdepasse=$_POST["password"];`

- `$_SERVER` et `$_ENV` qui donnent accès à des variables d'environnement et du serveur

ex: `$_SERVER["host"]` → hôte (ou virtualhost) utilisé  
`$_SERVER[" "]`

# Se connecter à MySQL

- `mysql_connect("localhost","dupont","pass");`  
`mysql_select_db("mabase");`

```
$requete="SELECT nom FROM users";
```

```
$resultat=mysql_query($requete);
```

```
while( $user=mysql_fetch_object($result) )  
    { print $user->nom ; }
```

```
mysql_close();
```

# Les fonctions

- `Function nom(arguments) { instructions }`

ex: `function afficher($texte)`

```
{ print $texte; // affiche le contenu de $texte  
  return true; } // optionnel, valeur de retour
```

En PHP comme dans d'autres langages les variables des fonctions sont locales. Pour utiliser une variable globale il faut l'importer avec la commande `global $variable;`

- Fonctions (très) utiles: `include` et `require`.  
elles permettent d'inclure d'autres pages pdt l'interprétation qui seront immédiatement interprétées:

ex: `include("connect.inc.php");`

# La requête SELECT

- Permet d'afficher le contenu d'une table
  - `SELECT champs FROM nom_table WHERE condition  
ORDER BY champ ASC/DEC LIMIT inf,sup`
- Retourne sous forme d'un tableau les valeurs des champs des tables indiquées qui vérifient la condition fournie
- Ex: `SELECT nom,prenom FROM users WHERE age<20 ORDER BY nom ASC LIMIT 0,30`

# Les requêtes INSERT et DELETE

- Permettent de remplir et vider une table
- INSERT INTO table (champs) VALUES (valeurs)

→ ex: INSERT INTO users (`nom`, `prenom`, `age`)  
VALUES ('dupont', 'roger', '42')

- DELETE FROM table WHERE condition

→ ex: DELETE FROM users WHERE nom='dupont'

# La requête UPDATE

- Permet de modifier les données d'une table
- UPDATE nom\_table SET affectation WHERE condition

→ ex: UPDATE users SET age=40 WHERE nom='dupont'

```
UPDATE users SET prenom='maurice' AND age=50 WHERE  
nom='dupont'
```

# Les requêtes CREATE et DROP

- Permettent de créer et supprimer des tables  
CREATE TABLE nom\_table ( structure ) OPTIONS

→ ex: CREATE TABLE `users` (  
    `id` smallint(4) NOT NULL auto\_increment,  
    `nom` varchar(30) NOT NULL default "",  
    `prenom` varchar(30) NOT NULL default "",  
    `age` smallint(4) NOT NULL  
    PRIMARY KEY (`id`)  
    ) TYPE=MYISAM AUTO\_INCREMENT=1 ;

→ ex: DROP TABLE `users` IF EXISTS

# Les requêtes TRUNCATE et ALTER

- TRUNCATE permet de vider une table (mais conserve la table et sa structure)

TRUNCATE TABLE table

- ALTER permet de modifier une table

ex: ALTER TABLE nom\_table ADD champ type AFTER champ

ex: ALTER TABLE nom\_table DROP champ

# Les autres instructions

- **INNER JOIN**: produit cartésien pour joindre deux tables
- **HAVING**: permet de faire des sous-groupes
- Fonctions: count, sum, average, password

# Liens utiles

---

- [www.php.net](http://www.php.net): le manuel officiel du PHP
- [www.mysql.net](http://www.mysql.net): le manuel officiel de MySQL

**A vous de jouer !**

