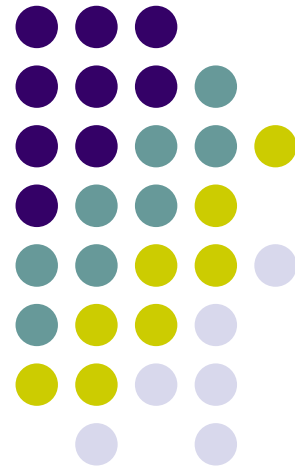
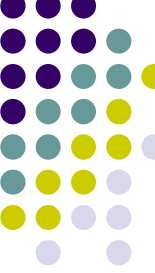


# Introduction au langage PHP

Licence Professionnelle  
Université Lyon 1, Bourg-en-Bresse  
Nicolas Zlatoff



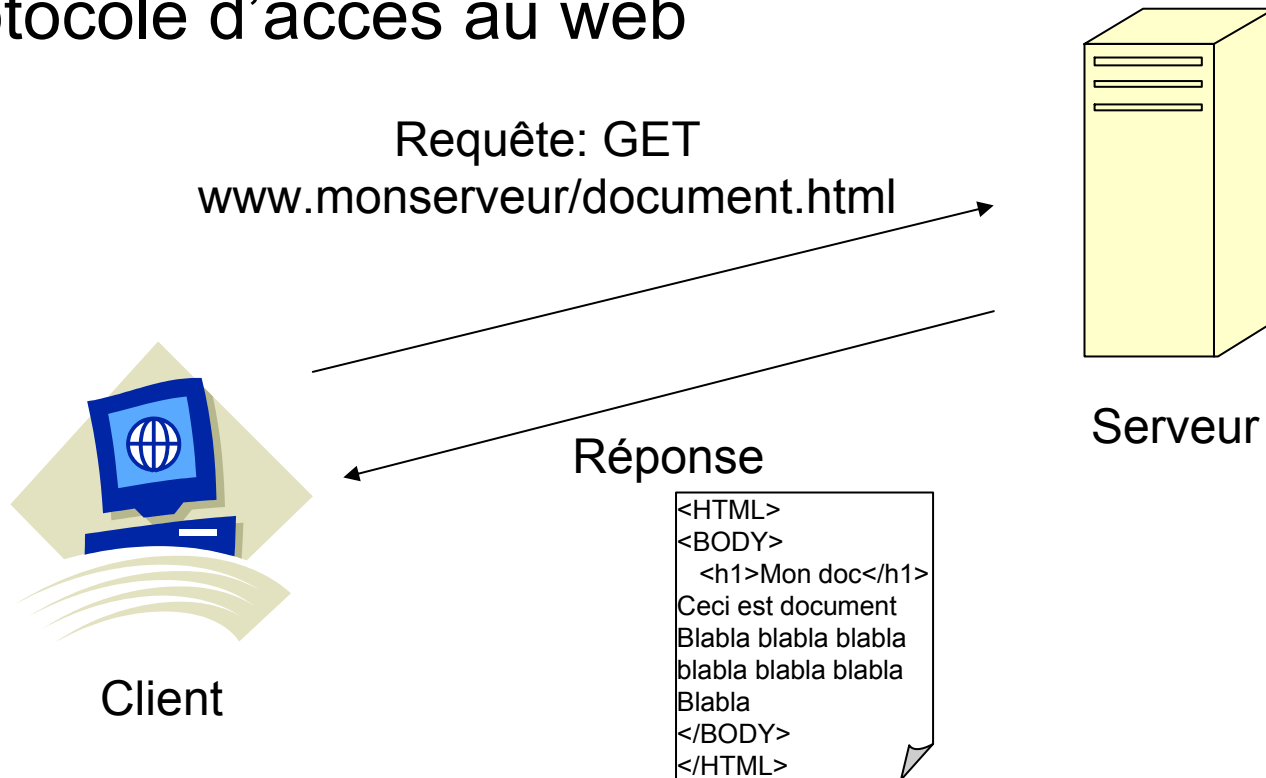


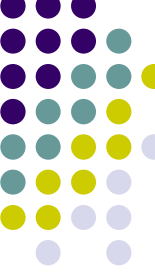
# Plan

- Pourquoi PHP?
- Présentation générale
- Premiers pas
  - Environnement
  - Bases du langage
- Activités pratiques

# Un rappel: HTTP & HTML

- HTTP (HyperText Transfert Protocol): protocole d'accès au web





# Un rappel: HTTP & HTML

- HTML (HyperText Markup Language): langage *statique* de description de pages web
- Les pages sont déposés sur le serveur...
- ...et appelées à la demande

```
<HTML>
<BODY>
  <h1>Un titre</h1>
  <p>Blabla bla Blabla blaBlabla blaBlabla
blaBlabla blaBlabla blaBlabla blaBlabla
blaBlabla bla
  <h2>Un deuxième titre</h2>
<p>Blabla bla Blabla blaBlabla blaBlabla
blaBlabla blaBlabla blaBlabla blaBlabla
blaBlabla bla blaBlabla blaBlabla

</BODY>
</HTML>
```

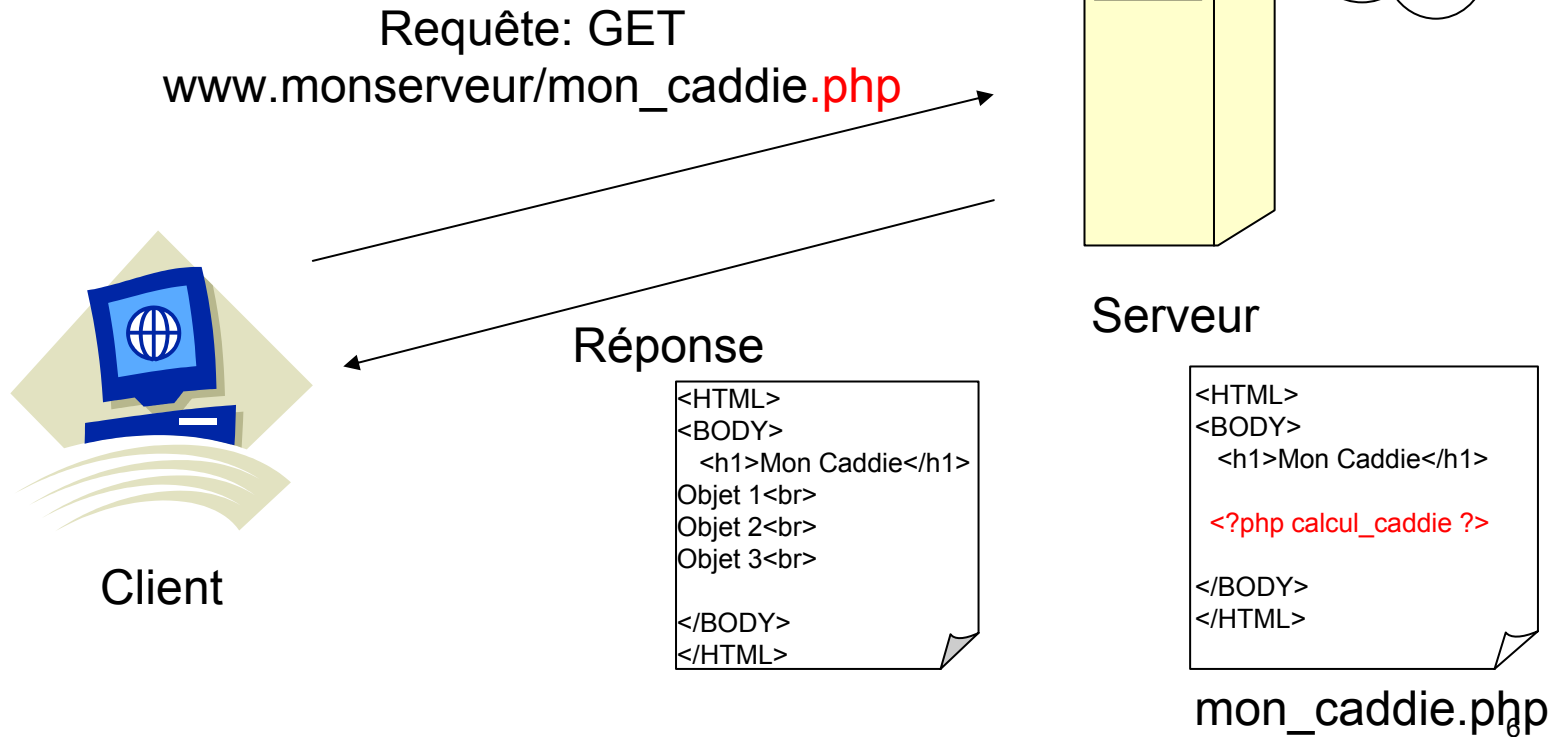
# Oui mais...

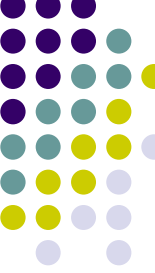


- Site de vente en ligne
  - Un utilisateur sélectionne un produit
  - Le met dans un caddie
  - Et souhaite afficher le contenu du caddie
- ... le résultat est *dynamique* (« variable »)
- ... il faut donc être capable de construire des pages *à la volée*

# Retour sur HTTP

- Construction à la volée d'une page HTML





# Plan

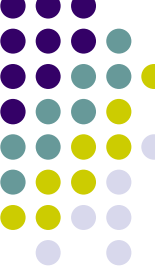
- Pourquoi PHP?
- **Présentation générale**
- Premiers pas
  - Environnement
  - Bases du langage
- Activités pratiques



# PHP: caractéristiques

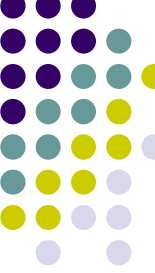
- « Personal Home Pages » puis « PHP Hypertext Preprocessor »
- Langage de scripts proche du C, Perl, Java (mais plus simple!)
  - Qui s'intègre dans une page HTML
  - Traité par le serveur
  - *spécifiquement dédié* à la création dynamique de pages HTML
- Créé en 1994 par Rasmus Lerdof
  - Aujourd'hui: PHP version 4
  - Licence GPL
  - Développé en C





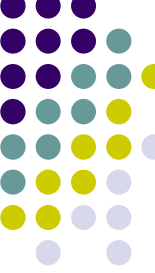
# PHP: atouts majeurs

- Rapidité, facilité de prise en main et de développement
  - ...car directement intégré dans les pages HTML
- Gestion d'outils avancés: sessions, cookies
- S'interface avec tous les SGBD
- Transparent vis-à-vis des OS et serveurs
  - Windows, Unix, Linux, MacOS
  - Apache, Microsoft IIS
- Programmation procédurale ou objet
- Grande communauté de développeurs



# Plan

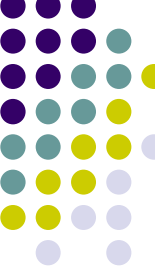
- Pourquoi PHP?
- Présentation générale
- Premiers pas
  - Environnement
  - Bases du langage
- Activités pratiques



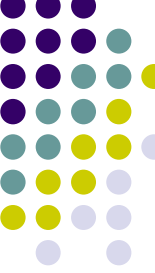
# Environnement

- Un OS
- Un serveur avec un interpréteur PHP
  - Apache: [www.apache.org](http://www.apache.org)
  - Configurable avec conf/httpd.conf
  - Voir aussi php.ini pour l'interpréteur PHP
- Un éditeur de texte
  
- Éventuellement une base de donnée
- Voir EasyPHP sur [www.easyphp.org](http://www.easyphp.org)

# Environnement



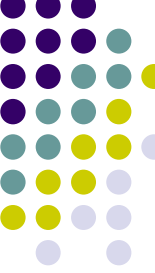
- Deux possibilités
  - Écrire des pages PHP, uploader sur serveur, et tester
  - Installer en local un serveur avec interpréteur PHP
- Le plus simple?



# Environnement

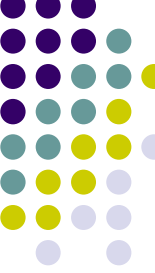
- Ecriture de index.php en texte brut
  - Attention à NotePad et autres!
- Sauvegarde de index.php dans Easyphp\www\mon\_repertoire
  - Path paramétrable dans httpd.conf
- Ouvrir un navigateur
- Exécuter le script

[http://localhost/mon\\_repertoire/index.php](http://localhost/mon_repertoire/index.php)



# Plan

- Pourquoi PHP?
- Présentation générale
- Premiers pas
  - Environnement
  - Bases du langage
- Activités pratiques



# Bases du langage

- Intégrer PHP dans HTML...

```
<html>
  <p>voici du texte HTML
  avec <?php print "un peu" ?> de
  code PHP.<br>
</html>
```

- ... ou HTML dans PHP, suivant les cas

```
<?php
  print "<html><p>";
  print "voici du texte HTML avec
  un peu de code PHP.<br></html>";
?>
```



# Bases du langage

- Fin de ligne avec ;

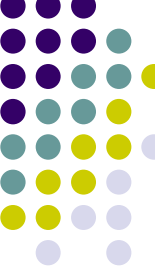
```
<?php
    print "Blablabla";
?>
```

- Commentaires: // ou /\* \*/

```
<?php
    // Commentaires
    print "Blabla";
    /* D'autres commentaires sur
    plusieurs lignes */
?>
```



# Bases du langage: types & variables

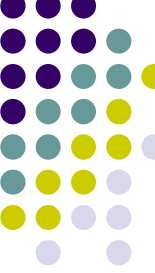


- PHP est un langage faiblement typé
- Variables identifiées avec \$, sans déclaration

```
<?php
    $my_integer = 12;
    $my_float = 1.03;
    $my_string = "coucou";
    $my_boolean = TRUE;
?>
```

- Transtypage

```
<?php
    $my_string = "az";
    $my_integer = 12;
    $my_string = $my_integer;
    print "value: $my_string";
?>
```



# Bases du langage: tableaux

- Tableaux « classiques »: la clé est un **integer**

```
<?php
    $my_array = array(1,45,65);
    print "élément no2:$my_array[1]";
    /* affiche 'élément no2:45' */
?>
```

- Tableaux associatifs: la clé est une **string**

```
<?php
    $my_array = array ('ain' => 1,
                       'aisne' => 2,
                       ...);
    print $my_array['aisne'];
    /* affiche '2' */
?>
```

# Bases du langage: portée des variables (1/2)



- Généralement, portée = script

```
<?php $my_integer = 12 ?>
<html><body>
  <h1>Une page HTML</h1>
  <?php print $my_integer ?>
  /* affiche 12 */
</body>
</html>
```

- Les variables sont locales aux fonctions

```
<?php
$a = 1; /* portée globale */
function test() {
  print $a; /* portée locale */
}
test();
/* n'affiche rien */
?>
```

# Bases du langage: portée des variables (2/2)



- Définition de variables globales

```
<?php
$a = 1;
$b = 2;
function somme() {
    global $a, $b;
    /* les variables $a et $b de la
    fonction sont déclarées globales
    et font référence à $a et $b ci
    dessus */
    $b = $a + $b;
}
somme();
/* affiche 3 */
print "\$b = $b";
```

# Bases du langage: structures de contrôle (1/2)



- Structures « classiques »
  - `if ... else` ou `if ... elseif`
  - `for`
  - `while`

```
?php
* exemple avec if */

if ($a > $b) {
    print "a + grand que b";
elseif ($a == $b) {
    print "a égal à b";
else {
    print "a + petit que b";
}
>
```

```
<?php
/* exemple avec for */

for ($i = 1; $i < 11; $i++)
{
    print "\$i = $i";
}
?>
```

```
<?php
/* exemple avec while */

$i = 1;
while ($i <= 10) {
    print $i++;
    /* La valeur affiché
    est $i avant
    l'incrémentatation
    (post-incrémentatation) */
}
>
```

# Bases du langage: structures de contrôle (2/2)



- Structures « spécifiques »
  - `foreach` pour le parcours de tableaux
  - `break`, `continue` à l'intérieur d'une boucle
  - `include()` lit et exécute le fichier en paramètre

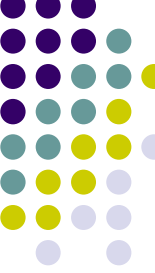
```
?php
/* exemple avec foreach */
$a = array (1,2,3,17);
foreach ($a as $v) {
    print "Current value: $v";
    print "<br>";
}
>
```

```
<?php
/* exemple avec foreach */
$a = array ("un" => 1,
           "deux" => 2,
           "trois" => 3,
           "dix-sept" => 17);

foreach($a as $k => $v) {
    print "\$a[$k] => $v";
}
?>
```

```
<?php
/* exemple avec include */
vars.php
<?php $couleur = 'verte';
      $fruit = 'pomme';?>

test.php
<?php
    print "Une $fruit $couleur";
    /* Une */
    include 'vars.php';
    print "Une $fruit $couleur";
    /* Une pomme verte */
```



# Et après?

- Documentation officielle
  - <http://www.php.net/docs.php>
- Voir aussi Apache
  - <http://www.apache.org>



# Plan

- Pourquoi PHP?
- Présentation générale
- Premiers pas
  - Environnement
  - Bases du langage
- **Activités pratiques**

<http://www710.univ-lyon1.fr/~nzlatoff/enseignements/PHP/TP1.html>