Le langage PHP

Walid Belkhir
Université de Provence
belkhir@cmi.univ-mrs.fr
http://www.lif.univ-mrs.fr/~belkhir/

Plan

Qu'est ce que PHP ?

- langage interprété indépendant de la plate-forme d'éxécution
- s'exécute sur le serveur
- les instructions sont intégrées au code source du document HTML
- permet de générer des pages HTML

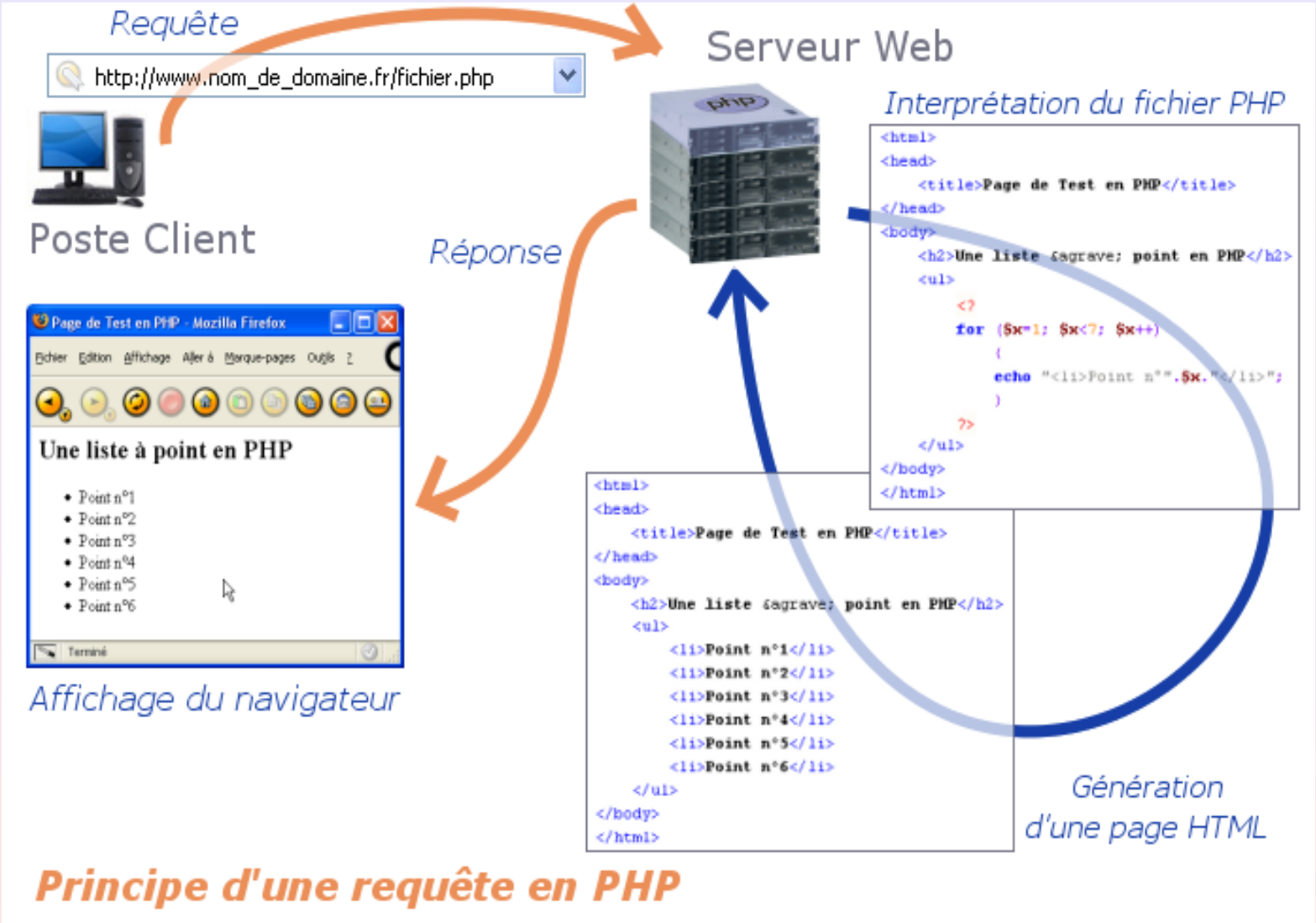
Un exemple : Ex1.php

```
<html> <head>
</head>
<body>
<H1> Voici l'email de l'administrateur du serveur </H1>
<br>
<?php
    echo $_SERVER['SERVER_ADMIN']
?>
</body>
</html>
```

Principes de PHP

1. L'interpréteur du code PHP se trouve au serveur HTTP
2. Le serveur lit les instructions PHP intégrés à la page HTML (entre <?php ... ?>), les interprète et les remplace par le résultat de leur exécution
3. La page HTML générée par le serveur est envoyée au client
4. Le navigateur au niveau client affiche la page HTML

Principes de PHP



Inconvénients

- Vitesse d'exécution :
  - langage interprété par le serveur avec plusieurs requêtes simultanées $\Longrightarrow$ pas très rapide
- Pas d'interactivité au niveau du client : JavaScript est donc nécessaire

Caractéristiques de PHP

- Très portable : fonctionne sous Windows/Unix/ ....
- Syntaxe similaire à celle du C
- Extensible par de nombreuses bibliothèques :
  - calcul mathématique, connexion sécurisée, accès à la plupart des SGBD
  - Logiciel Open Source (donc facilement extensible et gratuit)
  - Conçu pour fonctionner efficacement avec le serveur Apache
- Un fichier PHP (.php) peut contenir
  1. du code HTML
  2. du code PHP
  3. du code JavaScript

Intégrer du PHP dans une page HTML

- La façon la plus simple :

```
<?php echo "Salut" ; ?>
```

- Autre méthode :

```
<script language="php"> echo "Salut" ; </script>
```

- On peut trouver aussi :

```
<? echo "Salut" ; ?>
```

  - mais pose un problème de compatibilité avec XML

Intégration de fichiers PHP externe

- Un exemple
  < ?php echo "Bonjour" ; include "Fichier_Externe.php" ; ?>
- Le fichier Fichier_externe.php contient :
  < ?php echo "re-bonjour" ; ?>

Intégration des fichiers PHP externe

- Inclure des fichiers se fait avec la fonction include ou require
- L'instruction require fonctionne comme include sauf, si une erreur s'est produite dans le fichier externe alors
  - avec include une alerte (warning) s'affichera, et on continue l'exécution du script
  - avec require une erreur fatale se génère, et on interrompe l'exécution du script

Plan

Commentaires et casse

- On peut utiliser les commentaires C, C++, et Shell :

```
<?php
    /* commentaire type C */
    // commentaire type C++
    # commentaire type Shell
?>
```

- La casse n'intervient pas dans les noms de fonction :
  `echo "salut" ;` et
  `eChO "salut" ;` sont équivalente
- mais elle intervient dans les noms de variables :
  `echo $nom ;` et
  `echo $NOM ;` concernent deux variables différentes

Types de données et variables

- Les noms de variable sont précédés d'un $:
  ```
  $x=3 ;
  echo "$x" ;
  ```
- pas de déclaration : l'affectation détermine le type de la variable
- PHP supporte les types de données suivants :
  - entiers
  - nombre à virgule flottante
  - booléens
  - chaînes de caractères
  - tableaux
  - objets

Booléens, entiers et flottants

- Entiers :
  - $y=-6 ; // base 10
  - $y=034 ; // base 8
  - $y=0x34 ; // base 16
  - pas de division entière mais transtypage :
    $i= (int)(5/2) ; // $i vaut 2
- Booléens :
  le booléen FALSE, l'entier 0, le flottant 0.0, un tableau vide, un objet vide et la constante NULL sont considéré comme faux.

Chaines de caractères (1)

- $ch1='Bonjour' ; $ch2="$ch1 Monsieur" ;
- $ch1='Bonjour ' ; $ch2="${ch1}Monsieur" ;
- echo "\n" ; provoque un saut de ligne dans le code HTML généré.
- echo "Bonjour", "$ch2" ; affiche "Bonjour Monsieur"
- Accès aux caractères d'un chaîne :
  $ch='Salut' ; echo $ch{3} ; // affiche "u"
- printf (comme en C)

Chaînes de caractères (2)

Substitution de chaînes :

- addslashes(ch) : ajoute un \ devant tous les caractères spéciaux de la chaîne ch (utile pour les requête de bases de données)
- stripslashes(ch) : fait l'opération inverse (supprime les \)
- str_replace(ch1,ch2,ch) remplace dans ch toutes les occurrences de ch1 par ch2

Chaînes de caractères (3)

Découpage de chaînes :

- explode(sep,ch) : retourne un tableau de chaînes découpant ch à l'aide du séparateur sep
- implode(sep,tab) : retourne une chaîne fabriquée par la concaténation des éléments du tableau tab et du séparateur sep entre chaque élément

Comparaison de longueur

- strcmp(ch1,ch2) : comme en C
- strlen(ch) : longueur de ch

Chaînes de caractères (3)

Expression régulières

- . : n'importe quel caractère
- [A-Z] : caractère appartenant à l'ensemble des majuscules
- [^A-Z] : caractère n'appartenant pas à l'ensemble des majuscules
- $r*$: 0, ou plusieurs occurrences successive de l'expression régulière r
- $r+$: 1, ou plusieurs occurrences successive de r
- r$\{m,n\}$: entre m et n occurrences successives de r
- r$\{m\}$: exactement m occurrences successives de r
- r$\{m,\ \}$: au moins m occurrences de r
- \ : caractère d'échappement (m*n=m*n)
- (r1 | r2 | r3) : r1 ou r2 ou r3

Chaînes de caractères (4)

Expression régulières

- ereg(profil, ch, $tab) : recherche profil dans ch, retourne true ou false, et remplit $tab avec les occurrences trouvées (sensible à la casse)
- eregi(profil, ch, $tab) : pareil mais insensible à la casse
- ereg_replace(profil, ch2, ch) : recherche profil dans ch et le remplace par ch2 et retourne ch modifiée (sensible à la casse)
- ereg_replace(profil, ch2, ch) : pareil mais insensible à la casse
- split(profil, ch) : retourne un tableau de sous-chaîne de ch délimitées par profil
- spliti(profil, ch) : pareil mais insensible à la casse

Date et heure

- Afficher la date et l'heure actuelle :
  echo "On est le" . date("j m Y"). "et il est" .date("H \h i"). " mn" ;
- Les fonction date() et mktime()
  - date("format", $timestamp) : retourne une chaîne de caractères qui contient la date $timestamp au format indiqué
  - Un timestamp est la date Unix (nombre de secondes depuis 1er Janvier 1970)
  - si $timestamp est omis, il s'agit de la date actuelle
  - si date() : retourne le timestamp
  - mktime(h,m,s,M,J,A) retourne le timestamp associé à la date spécifié en paramètres

Tableaux (1)

- Les tableaux en PHP sont associtifs
  - l'index dans le tableau est appelé clé
  - la valeur associée à la clé est appelé valeur
  - un tableau est un ensemble d'associations clé/valeur
  - la clé peut être un entier ou une chaîne de caractères
- Création d'un tableau
  - soit directement en affectant des valeurs au tableau
  - soit en utilisant la fonction array()

```
$tab[0]=1 ;  // clé entière, valeur entière
$tab[1]="toto" // clé entière, valeur de type chaîne de caractères
$tab["Oxygen"]="O2" // clé et valeur de type chaîne de caractères
```

Tableaux (2)

- Quelque exemples
  $tab["toto"][1]=1 ; // tableau à 2 dimensions
  $tab2=array(1,"toto") ; // 0=> 1 et 1=>"toto"
  $tab3=array("Oxygen"=>"O2","Hydrogene"=>"H2") ;
- Nombre d'éléments d'un tableau : sizeof($tab)

Tableaux(3)

- Suppression d'un élément
  unset($tab["Oxygen"]) ; // marche aussi pour une variable
- Tris de tableaux
  - le tri peut se faire sur les clés et/ou les valeurs
  - asort()/arsort() : trie de tableau par ordre croissant/décroissant de valeurs
  - ksort()/krsort() : trie le tableau par ordre croissant/décroissant de clés
  - sort() : trie le tableau par ordre croissant de valeurs et réassigne des clés (0,1,...) ; l'association clé/valeur est perdu
  - uasort(), uksort(), usort() : identiques à leur homologue mais on doit fournir la fonction de comparaison

Tableaux (4)

Le pointeur de tableau

- à chaque tableau correspond un pointeur interne qui est une référence sur l'élément courant
- current($tab) : donne l'élément courant
- next($tab) : déplace le pointeur vers l'élément suivant
- prev($tab) : déplace le pointeur vers l'élément précédent
- end($tab) : déplace le pointeur vers le dernier élément
- reset($tab) : déplace le pointeur sur le premier élément

```
$tab=array("a"=>1, "b"=>5, "c"=>4);
$val=current($tab); echo "$val<br>" ; //affiche "1<br>"
$val=next($tab); echo "$val<br>" ; //affiche "5<br>"
```

Tableaux (5)

Extraction d'éléments d'un tableau

- list() : extraire des valeurs d'un tableau
  $tab=array(3,2,9) ; sort($tab) ; list($x,$y)=$tab ;
  echo"$x, $y" ; affiche "2 3"
- key($tab) ; : extraire la clé de l'élément pointé par le pointeur du tableau
  $tab=array("a"=>"3","b"=>"2","c"=>"9") ; next($tab) ;
  $cle=key($tab) ; $x=$tab[$cle] ;
  echo "$cle : $x" ; //affiche "b :2"
- extract($tab) : permet d'extraire d'un tableau toutes les valeurs, chaque valeur est recopiée dans une variable dont le nom est la valeur de la clé :
  $tab=array("x"=>3, "y"=>"9") ; extract($tab) ;
  echo"$x $y" ; //affiche "3 9"

- Extraction d'éléments d'un tableau
  - each($tab) retourne la paire clé/valeur courante du tableau et avance le pointeur ;
- Parcours du tableau
  - avec list(), et each()
    $tab=array("a"=>"2", "c"=>"5", "b"=>"9") ;
    while(list($k, $v)= each ($tab)) {
    echo "$k : $v
" ;}
  - Quand le pointeur dépasse la fin du tableau, each retourne false

Tableaux (5)

Parcours du tableau

- foreach() place le pointeur en tête du tableau et parcourt l'ensemble des éléments, foreach() travaille sur une copie du tableau original

```
$tab=array("a"=>3,"b"=>2, "c"=>5,"d"=>8);
//Pour parcourir des valeurs uniquement :$v=3,2,5,8
foreach($tab as $v) { echo"$v" ;}
//Pour parcourir des couples clé/valeur
foreach($tab as $k=>$v) { echo"$k : $v" ;}
```

Les constantes

- Constantes définies par le programmeur :
  DEFINE ("PI",3.14) ; echo PI ;
  DEFINE ("FAC","CMI") ; echo FAC ;
- Principale constantes définies par PHP
  - _FILE_ : chemin absolu du fichier en cours d'exécution
  - _LINE_ : numéro de la ligne en cours d'exécution
  - PHP_VERSION : version de PHP qui est utilisée
  - PHP_OS : Système d'exploitation de la machine
  - E_* : gérer les erreur (E_ALL = toutes les erreurs)
  - ...

Les erreurs

- 4 types d'erreurs/alertes en PHP
  - E_ERROR : erreur d'exécution
  - E_WARNING : alerte
  - E_PARSE : erreur d'analyse
  - E_NOTICE : notes
- err_log() : envoie un message d'erreur dans les logs du serveur web, dans un fichier, ...

la valeur NULL

- Constante particulière qui représente l'absence de valeur
-

```
$var=NULL ;
isset($var) —> retourne FALSE
is_null($var) —> retourne TRUE
$ch="" ;
isset($ch) —> retourne TRUE
is_null($ch) —> retourne FALSE
```

Les opérateurs

- Opérateurs arithmétiques : +,- ,* , % (modulo)
- Concaténation de chaîne de caractère (.) :
  echo $ch1.$ch2 ;
  echo "Bonjour " . $ch ;
- Opérateur binaires : && (et), || (ou), ! (not)
- affectation : =
- Opérateurs de comparaison : ==, <=, <, ...

Instructions

- if/then/else, switch comme en C
- while, for, do while comme en C
- instruction conditionnelle simplifié :

```
if (condition1) :
...
elseif (condistion2) :
...
elseif (condistion3) :
...
else : ...
endif ;
```

Les fonctions

- Déclaration de fonction : comme en C, JavaScript

```
function fact($n) {
if ($n==0) {return 1 ;} else { return $n*fact($n-1) ; }
```

- Les arguments de fonction en PHP supporte :
  1. passage par valeur
  2. passage par référence

Les variables prédéfinies

- $GLOBALS : variables globales de l'exécution en cours
- $_SERVER : variables fournies par le serveur WEB
- $_COOKIE : variables issues des cookies HTTP reçus
- $_FILES : variables fournies par HTTP suite à un téléchargement de fichiers
- $_ENV : variables d'environnement positionnées au démarrage du serveur WEB

Variables globales

- déclarer la variable comme global dans le bloc (cela crée une référence locale sur la variable globale)
- utiliser le tableau associatif $GLOBALS

```
<?php $a1=1; $a2=2; // variables globales

function affiche() {
global $a1; // $a1 est la variable globale
echo $a1." et " .$GLOBALS['a2'];
$a1++; }

affiche(); //-> affiche "1 et 2"
echo $a1; // --> affiche "2", $a1 a été modifiée
?>
```

Variables et type

- is_array(), is_bool, is_double, is_float(), is_int, is_object, gettype() ; settype()
- gettype($var) retourne le type de $var

```
$foo=true ;
settype($foo,"string") ;  // $foo vaut maintenant "1"
```

Le transtypage

- Les transtypages sont :
  - (int)
  - (bool)
  - (double), (float)
  - (string)
  - (array)
  - (object)
- Exemple
  $x= 1.7 ; $y= (int) ($x * 2) //–> $y vaut 3
- Attention : le transtypage n'a pas toujours une valeur prévisible