PHP

Thierry Lecroq

Université de Rouen
FRANCE

Plan

UNIVERSITÉ DE ROUEN

Pages dynamiques

Exemple

Exemple statique

```
<html>
<head><title>Page statique</title></head>
<body>
Nous sommes le 2/10/2008
</body>
</html>
```

Problème

Afficher une page différente en fonction de l'utilisateur, de l'environnement, ...

Solution

Utiliser un langage de programmation évolué, par exemple PHP

Présentation

- Langage récent (crée en 1994)
- Versions utilisées :
  - 4.3 (plus répandue)
  - 5.0 (avec une couche objet)
- Langage de script
- Langage interprété
- Présence d'un interpréteur côté serveur
- Intégré au code HTML
- Syntaxe proche du C et du Java
- Interface simple avec beaucoup de SGBD

UNIVERSITÉ DE ROUEN

Modèle d'exécution

1. le client demande une page PHP
2. le serveur web exécute le code de la page
3. lancement de l'interpréteur
4. exécution du code
5. le serveur web renvoie le résultat de l'exécution
6. le client affiche le résultat

Pour le client, il est impossible de voir le code PHP
Seul le résultat de l'exécution est récupéré par le client

UNIVERSITÉ DE ROUEN

Premier exemple

Code côté serveur

```
<html>
<head><title>Page dynamique</title></head>
<body>
<?php
echo ( "Nous sommes le " ) ;
echo ( date ( "j/m/Y" ) );
?>
</body>
</html>
```

Premier exemple

Résultat côté client

```
<html>
<head><title>Page statique</title></head>
<body>
Nous sommes le 23/10/2008
</body>
</html>
```

Mélange HTML/PHP

- PHP s'intègre dans l'HTML entre `<?php` et `?>`
- Les instructions se finissent par ;
- Les commentaires sont soit entre `/*` et `*/`, soit après `//` ou `#`
- Manuel complet en français : `http://www.php.net/manual/fr`

UNIVERSITÉ DE ROUEN

Plan

UNIVERSITÉ DE ROUEN

Les variables

- Les variables sont préfixées par $
- Leur nom suit les règles classiques
  Exemple : `$my_var_03`
- Les noms sont sensibles à la casse : `$var = $Var`
- Pas de déclaration, typage implicite
  Exemple :

```
$my_var_03 = 54 ; // Maintenant, c'est un entier
$my_var_03 = "pif" ; // Maintenant, c'est une chaine
```

- Attention aux fautes de frappes dans les noms de variables

UNIVERSITÉ DE ROUEN

Les types

- entiers : 54
- flottants : 54.3
- chaînes : "54" ou '54'
- booléens : `false` ou `true`
- tableaux

Fonctions de test

- `isset($var)` : renvoie `true` si `$var` existe
- `unset($var)` : détruit `$var`
- `is_integer($var)`, `is_string($var)`,... : renvoie true si `$var` est un entier, une chaîne,...

UNIVERSITÉ DE ROUEN

Les commentaires et les constantes

Les commentaires

- après `//` : sur une ligne
- après `#` : sur une ligne
- entre `/*` et `*/` : sur plusieurs lignes

Constantes

On les définit à l'aide de la commande `define`
Exemple : `define("PI", 3.14)`
On les utilise directement (sans $) : `echo(PI)`
Test d'existence : `defined("PI")` renvoie 1, `defined("Pi")` renvoie 0

UNIVERSITÉ DE ROUEN

Les entrées/sorties

Les entrées

À l'aide de formulaires

Les sorties

On peut afficher avec la commande `echo` (avec ou sans parenthèses)
`print` est équivalente à `echo`
On peut faire un affichage comme en C avec printf

Opérateurs

- Arithmétiques : `+ - * / % ++ --`
- Affectation : `= .= += -= *= /= %=`
- Comparaison : `== < != > === <= !== >=`
- Logiques : `and && or || xor !`
- Conditionnel : `... ? ... : ...`

Instructions conditionnelles

if

```
if ( cond ) {
...
}
elseif ( cond ) {
...
}
else {
...
}
```

Instructions conditionnelles

switch

```
switch ( expr ) {
case VALEUR1 :
...
break ;
case VALEUR2 :
...
break ;
default :
...
break ;
}
```

Instructions itératives

for

```
for ( init ; cond ; modif ) {
...
}
```

while

```
while ( cond ) {
...
}
```

do while

```
do {
...
} while ( cond ) ;
```

Les tableaux

- Chaque élément du tableau a une clé et une valeur
- Pas de déclaration du tableau
- Les valeurs des éléments ne sont pas forcément toutes du même type

Exemple

remplissage à la volée :

```
$tab [ 0 ] = 54 ;
$tab [ 1 ] = "pif" ;
$tab [ "paf" ] = false ;
```

Exemple

remplissage direct :

```
$tab = array (54 , "pif" ) ;
$tab = array ( "paf" => false ) ;
```

Parcours de tableaux

Parcours « classique » avec `for`
Parcours spécifique :

```
foreach ( $tab as $value ) {
...
}
foreach ( $tab as $key => $value ) {
...
}
```

Les tableaux

Fonctions prédéfinies

- `count($tab)` : compte le nombre d'éléments initialisés
- `current($tab)` : retourne la valeur de l'élément en cours
- `key($tab)` : retourne l'indice de l'élément en cours
- `reset($tab)` : déplace le pointeur vers le premier élément
- `list($indice,$valeur)` avec `each($tab)` : permettent de parcourir les couples (indice, valeur) même si les indices ne sont pas consécutifs
- `next($tab)` : déplace le pointeur vers l'élément suivant
- `prev($tab)` : déplace le pointeur vers l'élément précédent

UNIVERSITÉ DE ROUEN

Les tableaux

Fonctions prédéfinies

- `sort($tab)` : trie les valeurs et réaffecte les indices
- `asort($tab)` : trie les valeurs et ne réaffecte pas les indices
- `rsort($tab)` : id `sort` mais dans l'ordre inverse
- `arsort($tab)` : id `asort` mais dans l'ordre inverse
- `ksort($tab)` : trie les indices
- `krsort($tab)` : id `ksort` mais dans l'ordre inverse
- `usort($tab,$critere)`, `uasort($tab,$critere)`, `uksort($tab,$critere)` : trie selon un citère

UNIVERSITÉ DE ROUEN

Les chaînes de caractères

- Délimitées par ’ : contenu non interprété
- Délimitées par " : contenu interprété
- Les unes peuvent contenir les autres
- Concaténation avec .

Exemple

```
$pif = "toto" ; // Contient "toto"
$paf = "comme $pif" ; // Contient "comme toto"
$pouf = ’pas comme $pif’ ; // Contient "pas comme $pif"
$bim = $pif.$paf ; // Contient "toto comme toto"
```

UNIVERSITÉ DE ROUEN

Les chaînes de caractères

- Accès à un caractère : `$bim[0]`
- `strlen ( $str )` : longueur de `$str`
- Comparaison avec ==, === ou `strcmp`

Les chaînes de caractères

- `str_repeat(ch, nb)` : répétition
- `strtolower(ch)` : minuscules
- `strtoupper(ch)` : majuscules
- `ucwords(ch)` : initiales en majuscules
- `ucfirst(ch)` : 1re lettre en majuscule
- `ltrim(ch, liste)` : suppression de caractères au début
- `rtrim(ch, liste)` : suppression de caractères à la fin
- `trim(ch, liste)` : suppression de caractères au début et à la fin

UNIVERSITÉ DE ROUEN

Les chaînes de caractères

- `strstr(ch, ch2)` : recherche sensible à la casse (retourne tous les caractères de `ch` depuis la 1re occurence de `ch` jusqu'à la finn)
- `stristr(ch, ch2)` : recherche insensible à la casse
- `substr(ch, indice, N)` : extraction de chaîne de caractères
- `substr_count(ch, ssch)` : décompte du nombre d'occurences d'une sous-chaîne
- `str_replace(oldssch, newssch, ch)` : remplacement
- `strpos(ch, ssch)` : position

UNIVERSITÉ DE ROUEN

Les fonctions

```
function ma_fonc ( $param1 , $param2 , ... ) {
...
return ... ;
}
```

- pas de type pour les paramètres ou la valeur de retour
- nombre fixé de paramètres
- le nom ne commence pas par $
- le nom est insensible à la casse
- le résultat est renvoyé avec la commande `return`
- une seule valeur de retour
- passage des paramètres par valeur (par défaut)
- passage par référence : `&$param`

Les fonctions

Example

```
function double($n) {
    $n *= 2;
    return $n;
}

$x = 12
echo "double = " . double($x) . " valeur = " . $x
echo "double = " . double(&$x) . " valeur = " . $x
```

Les fonctions

Les variables utilisées à l'intérieur d'une fonctions sont détruites à la fin, sauf :

- si on les définit avec `static`
- si on les définit avec `global`

UNIVERSITÉ DE ROUEN

Les fonctions

Exemple

```
function ma_fonc ( ) {
static $appels = 0 ;
$appel s++;
echo ( "J'ai ete appellee $appels fois" ) ;
}

function ma_fonc2 ( ) {
global $var ;
$var = 54 ;
}

$var = 0 ;
ma_fonc2 ( ) ;
echo ( $var ) ;
```

Inclusion de fichiers

- `require (" fichier ")`
- `include (" fichier ")`
- `require_once(" fichier ")`
- `include_once(" fichier ")`

Les variantes `include` provoquent des warnings au lieu d'erreurs en cas de problème
Les variantes `_once` n'incluent le fichier que si celui n'a pas déjà été inclu

Plan

UNIVERSITÉ DE ROUEN

Les formulaires ou les entrées

L'élément `form`

```
<form>...</form>
```

Les formulaires

Les attributs de l'élément `form`

- `action` : permet de préciser comment doivent être traitées les données du formulaire côté serveur
  ex

  ```
  <form action="traitement.php">
  <form action="http//www.site.com/script/traitement.php">
  <form action="mailto:Thierry.Lecroq@univ-rouen.fr">
  ```

  traitement par le fichier lui-même

  ```
  <form action="<?=$_SERVER['PHP_SELF']?>">
  ```

- `method` : méthode d'envoi des données vers le serveur
  - `get` : utilisée par défaut, données transmises visibles par l'utilisateur
    ex

    ```
    http//www.site.com/script/traitement.php?prenom=Thierry&nom
    ```

  - `post` : données non visibles mais pas de navigation avec Précédent/Suivant

Les éléments descendants de l'élément `form`

- `<fieldset>` : délimite des groupes de composants actifs, contient l'élément `<legend>`
- Bouton d'envoi

```
<input type="submit" value="Envoyer"
       name="soumission1"
       tabindex="5" accesskey="E"
       title="Bouton d'envoi" />
<button type="submit"
        name="soumission2"
        tabindex="2" accesskey="B"
        title="Bouton d'envoi">
 Envoyer
</button>
```

Les éléments descendants de l'élément `form`

avec une image

```
<button type="submit"
        name="soumission2"
        tabindex="2" accesskey="B"
        title="Bouton d'envoi">
 Envoyer
</button>
```

Les éléments descendants de l'élément `form`

- bouton de remise à zéro

```
<input type="reset" value="Effacer" name="effacement" />
```

La saisie de texte

La saisie de texte uniligne

```
<label>Votre nom : </label>
<input type="text" name="nom"
       size="10" maxlength="25"
       value="Lecroq" onclick="this.value=’’" />
```

Autres attributs :

- `disabled="disabled"`
- `readonly="readonly"`
- `onfocus="script"`
- `onchange="script"`

UNIVERSITÉ DE ROUEN

Exemple.xhmtl

Exemple

```
<form method="post" action="exemple.php">
 <fieldset>
  <legend>Donn&eacute;es personnelles</legend>
  <label>Nom : </label>
  <input type="text" name="nom" maxlength="25" />
  </br></br>
  <label>Pr&eacute;nom : </label>
  <input type="text" name="prenom"
         value="Votre pr&eacute;nom"
         maxlength="25"
         onclick="this.value=''" />
  </br></br>
  <label>Adresse : </label>
  <input type="text" name="adresse" maxlength="60" />
  <label>Code postal : </label>
  <input type="text" name="codePostal" size="5" maxlength="5" />
  <label>Ville : </label>
  <input type="text" name="ville" maxlength="25" />
  <input type="submit" name="envoi" value="Envoyer" />
 </fieldset>
</form>
```

Exemple.php

Exemple

```
<table>
 <?php
  foreach ($_POST as $cle => $valeur) {
   echo "<tr><td>".$cle."</td><td>".$valeur."</td></tr>";
  }
 ?>
</table>
```

Saisies

Saisie de mot de passe

`type="password"`

Saisie de texte long

élément `<textarea>`

attributs

- `cols="N"` : largeur de la zone
- `rows="N"` : hauteur visible de la zone

Exemple

```
<textarea name="commentaires" cols="70" rows="10"
          onclick="this.value=''">
Tapez vos commentaires ici...
</textarea>
```

Les boutons radio et les cases à cocher

```
<input type="radio" name="nom1" value="valeur1"
       checked="checked" />
<input type="radio" name="nom2" value="valeur2" />
<input type="checkbox" name="nom3" value="valeur3"
       checked="checked" />
<input type="checkbox" name="nom4" value="valeur4" />
```

UNIVERSITÉ DE ROUEN

Les listes de sélection

```
<select name="nom">
 <option value="valeur1">Option 1</option>
 <option value="valeur2">Option 2</option>
 .
 .
 .
 <option value="valeurN">Option N</option>
</select>
```

les attributs

- `size="N"` : nombre de lignes visibles lors de l'affichage (1 par défaut)
- `multiple="multiple"` : plusieurs choix possibles dans la liste (maintenir la touche `Ctrl` enfoncée)

UNIVERSITÉ DE ROUEN

Groupes d'option

Exemple

```
<select name="nom">
 <optgroup label="Groupe 1">
  <option value="valeur1">Valeur 1</option>
  <option value="valeur2">Valeur 2</option>
 </optgroup>
 <optgroup label="Groupe 2">
  <option value="valeur3">Valeur 3</option>
  <option value="valeur4">Valeur 4</option>
  <option value="valeur5">Valeur 5</option>
 </optgroup>
</select>
```

UNIVERSITÉ DE ROUEN

Le transfert de fichiers

```
<form action="fichier.php" method="post"
      enctype="multipart/form-data">
<input type="file" name="fichier" accept="type MIME">
```

côté serveur en php : $_FILES (tableau)

Récapitulatif

Les différents types pour les balises `input` :

- `text` : une zone de texte sur une seule ligne
- `password` : idem, mais avec affichage d'étoiles
- `file` : permet la sélection d'un fichier
- `checkbox` : une case à cocher
- `button` : un bouton simple (pas d'action sans javascript)
- `hidden` : un champ texte caché
- `radio` : un bouton d'option
- `reset` : un bouton de remise à zéro
- `submit` : un bouton de soumission

UNIVERSITÉ DE ROUEN

Plan

UNIVERSITÉ DE ROUEN

Les fichiers

Ouverture

```
$fd=fopen(chemin, mode)
```

mode

- `r` : lecture (read)
- `w` : écriture (write)
- `a` : ajout (append)
- `+` : lecture/écriture

Lecture caractère par caractère

```
$car=fgetc($fd)
```

UNIVERSITÉ DE ROUEN

Les fichiers

Lecture de lignes

```
$ligne=fgets($fd, [longueur])
```

Lecture d'octets

```
$v=fread($fd, nboctets)
```

taille d'un fichier

```
filesize(chemin)
```

Les fichiers

Écriture

```
fwrite($fd, message, [longueur-maximale])
fputs
```

Fin de fichier

```
feof($fd)
```

Fermeture

```
fclose($fd)
```

Existence d'un fichier

```
file-exists(chemin)
```

UNIVERSITÉ DE ROUEN

Plan

UNIVERSITÉ DE ROUEN

Les cookies

- conçus par Netscape
- fichiers texte courts stockés par le navigateur (côté client)
- analogie : carte d'identité
- cookie sans durée d'expiration : mémoire vive
- cookie avec durée d'expiration : mémoire secondaire

UNIVERSITÉ DE ROUEN

Exemple

```
<?php
   setcookie('truc', 'machin');
?>
<html>
   <head>
      <title>Titre</title>
   </head>
   <body>
      <p>Un cookie a &eacute;t&eacute; envoy&eacute;</p>
   </body>
</html>
```

Exemple

```
<html>
   <head>
      <title>Titre</title>
   </head>
   <body>
      <?php
         if (isset($_COOKIE['truc'])) {
            echo '<p>Un cookie a &eacute;t&eacute; envoy&eacut
            echo '<p>Son contenu est : ';
            echo $_COOKIE['truc'];
            echo '</p>';
         }
         else {
            echo "<p>Aucun cookie du nom de truc n'a &eacute;t
         }
      ?>
   </body>
</html>
```

Les cookies

Suppression d'un cookie

`setcookie('nom')`

Modification de la valeur d'un cookie

`setcookie('nom', nouvelle valeur)`

Validité et date d'expiration

`setcookie('nom', valeur, timestamp)`

`timestamp` : nombre de secondes depuis le 1erjanvier 1970

`mktime(heures, minutes, secondes, mois, jour, an)`

UNIVERSITÉ DE ROUEN

Plan

UNIVERSITÉ DE ROUEN

Persistence des données

On veut parfois garder de l'information entre plusieurs pages :

- Login / Password
- Préférences de navigation
- Sélection de produits à acheter (panier, ...)

On utilise donc les sessions PHP.
Les cookies permettent de stocker des informations côté client.
Les sessions permettent de stocker des informations côté serveur.
Elles sont identifiées par un numéro qui reste valide tant que le visiteur reste connecté.
Le numéro est transmis au serveur soit dans l'URL, soit dans un cookie.
Les données se placent et se récupèrent dans `$_SESSION`, comme pour les formulaires.

UNIVERSITÉ DE ROUEN

Utilisation des sessions

La session existe dès qu'elle est crée et jusqu'à ce qu'elle soit détruite
Création (et réouverture) : `session_start ()`
Destruction : `session_destroy ()`
Note : les sessions s'autodétruisent après un certain temps (généralement 30 min)

Exemple d'utilisation

```
<html>
 <head>
  <title>Connexion au site</title>
 </head>
 <body>
  <form method="post" action="verifLogin.php">
   <label>Login</label>
   <input type="text" name="login">
   <br/>
   <label>Mot de passe</label>
   <input type="password" name="password">
   <br/>
   <input type="submit" name="submit" value="Login">
  </form>
 </body>
</html>
```

```php
<?php
// On démarre la session
session_start();

// On n'effectue les traitement qu'à la condition que
// les informations aient été effectivement postées
if ( isset($_POST) && (!empty($_POST['login'])) && (!empty($_P

  extract($_POST);  // création des variables $login, $mdp et

  // On va chercher le mot de passe afférent à ce login
  // Établir la connexion, sélectionner la base
  $connexion=mysql_connect("localhost","","");
  mysql_select_db("nom_de_la_base");
  $sql = "SELECT pseudo, age, sexe, ville, mdp FROM user WHERE
  $req = mysql_query($sql) or die('Erreur SQL : <br />'.$sql);

  // On vérifie que l'utilisateur existe bien
  if (mysql_num_rows($req) > 0) {
```

Utilisation

```
<?php
// On appelle la session
session_start();

// On affiche une phrase résumant les infos sur l'utilisateur
echo 'Pseudo : ',$_SESSION['pseudo'],'<br/>',
     'Age : ',$_SESSION['age'],'<br/>',
     'Sexe : ',$_SESSION['sexe'],'<br/>',
     'Ville : ',$_SESSION['ville'],'<br/>';
?>
```

UNIVERSITÉ DE ROUEN

Déconnexion

Code de la page logout.php

```
<?php
// On appelle la session
session_start();

// On écrase le tableau de session
$_SESSION = array();

// On détruit la session
session_destroy();
?>
```

Plan

UNIVERSITÉ DE ROUEN

XHTML 2.0

- `href` peut être attribut de tout élément
- `acronym`, `b`, `i`, `small`, `big`, `tt` disparaissent
- `hr` est remplacé par `separator`
- l'attribut `role` permet d'ajouter des informations sémantiques aux éléments (incorporation du RDF (Ressource Description Format))
- `<section><h>` plutôt que `<h`n`>`

en concurrence avec X/HTML 5