

# PHP:Dynamiser un site HTML



# Sommaire



- Les fonctions d'inclusion.
- Mutualiser le code HTML.
- Affichage de balises HTML avec des boucles.
- Fonctions utilisateurs.
- Portée des variables et variables globales.
- Fonctions internes sur les tableaux.

# Include\_path

- La directive de configuration `include_path` spécifie une liste de répertoires dans lesquels PHP ira chercher les fichiers pour les fonctions `include()`, `require()`, `fopen()`, `file()`, `readfile()`, et `file_get_contents()`.
- PHP vérifiera dans le premier chemin, puis le suivant si le fichier n'est pas trouvé, etc. Si à la fin de la liste aucun fichier n'est trouvé, une erreur de type `WARNING` ou `FATAL` sera retournée.

# Les fonctions d'inclusion



- `include()` permet d'ajouter et d'exécuter un fichier spécifié en argument. Le script est recherché en fonction du chemin fourni, et aussi dans l'`include_path`. Si le fichier n'est pas trouvé, une erreur de type `WARNING` sera retournée.
- `require()` est identique à `include()`, sauf que l'erreur en cas d'absence du fichier est de type `FATAL`.
- `include_once()` et `require_once()` ajoutent et évaluent le code durant l'exécution, afin d'éviter d'inclure plusieurs fois le même code.

# Les fonctions d'inclusion



Fichier 1:

```
<?php
    $nombre1 = 3;
    $nombre2 = 5;
?>
```

Fichier 2:

```
<?php
    include "fichier1.php";
    $resultat = $a + $b;
    echo $resultat;
?>
```

# Mutualiser le code HTML



## Fichier 1:

```
<html>
  <head>
    <title><?php echo $titre; ?></title>
    <link ...>
  </head>
  <body>
```

## Fichier 2:

```
<?php
  $titre = "Bienvenue dans ma page web";
  include "fichier1.php";
  $resultat = $a + $b;
  echo $resultat;
?>
```

# Exercice pratique:

Mettre en place une entête et un pied de page fonctionnels pour la suite de la formation



# Affichage de tableaux multidimensionnels



```
<?php
$tableau = array(array('langage'=>'PHP', 'age' => 20),
                  array('langage'=>'JAVA', 'age' => 20),
                  array('langage'=>'C#', 'age' => 20),
                  array('langage'=>'COBOL', 'age' => 20),
                  array('langage'=>'BASIC', 'age' => 20),
                  array('langage'=>'C', 'age' => 20),
                  array('langage'=>'C++', 'age' => 20)
                );

echo '<table><tr><th>Langage</th><th>Age</th></tr>';
foreach($tableau as $langage)
    echo '<tr><td>'.$langage[0].</td><td>'.$langage[1].</td></tr>';
echo '</table>';
?>
```



# Exercice pratique:

Afficher le tableau multidimensionnel  
avec deux boucles en récupérant les entêtes  
directement dans le tableau



# Les fonctions utilisateurs



- Une fonction comporte
  - Un prototype ou une déclaration
  - Un corps
  - Une valeur de retour
- Le prototype:
  - nom, propriétés, arguments
- Le corps:
  - L'ensemble des instructions qui font partie de la fonction
- La valeur de retour:
  - Elle est renvoyée par le mot clé 'return'

# Les fonctions utilisateurs



```
<?php
/* Une fonction qui renvoi le double d'une valeur
*/
Le prototype est : « function double($number) »
Le corps est tout ce qui est entre { et }
/* @param integer $number
@return integer
*/
function double($number) {
return $number * 2;}
?>
```

# Les fonctions utilisateurs



```
<?php
include 'function_double.php';
// Appel de la fonction double
// Affiche la valeur 68
$result = double(34);
echo $result;
// On peut également combiner les appels de fonctions
// Affiche 16
echo double(double(4));
?>
```

# Les fonctions utilisateurs



```
<?php
function double($number) {
    // Cette variable est visible uniquement dans la
    // fonction
    $multiplier = 2;
    return $number * $multiplier;
}
// N'affiche rien, car $multiplier est 'local' à
// la fonction double(...)
echo $multiplier;
?>
```

# Exercice pratique

Créer une fonction qui retourne un tableau  
HTML en fonction d'un tableau PHP  
à deux dimensions



# Les fonctions sur les tableaux



On peut appliquer une fonction sur un tableau.

- `array_map($callback, $tab1)` retourne un tableau contenant tous les éléments du tableau `$tab1` après leur avoir appliqué la fonction `callback`.
- `array_filter($tab1, $callback)` évalue tous les éléments du tableau `$tab1` en les passant à la fonction `callback`. Si la fonction retourne `true`, l'élément est retourné dans le tableau. Les clés sont conservées.

# Les fonctions sur les tableaux



```
<?php
function odd($var)
{
    return($var & 1); // retourne lorsque l'entrée est impaire
}

function even($var)
{
    return(!($var & 1)); // retourne lorsque l'entrée est paire
}

$array1 = array("a"=>1, "b"=>2, "c"=>3, "d"=>4, "e"=>5);
$array2 = array(6, 7, 8, 9, 10, 11, 12);
echo "Impair :";
print_r(array_filter($array1, "odd"));
echo "Pair :\n";
print_r(array_filter($array2, "even"));

?>
```



# Les fonctions sur les tableaux



```
<?php
function cube($n)
{
    return($n * $n * $n);
}

$a = array(1, 2, 3, 4, 5);
$b = array_map("cube", $a);
print_r($b);
?>
```

# Exercices pratiques

