



# Éléments de langage

## Les balises PHP

<http://php.net/manual/fr>

//balise de début de code

<?php

...

//balise de fin de code

...

?>

A l'exécution le moteur PHP exécute le code à l'intérieur des balises et ignore le reste du code



# Éléments de langage

## Les balises PHP

Exemple d'exécution de code PHP par le moteur PHP:

```
<?php
$message = 'bienvenue';
$nom = 'MyName';
?>
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Test PHP</title>
  </head>
  <body>
    <?php
    echo $message . ' ' . $nom;
    ?>
  </body>
</html>
```



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Test PHP</title>
  </head>
  <body>
    bienvenue MyName
  </body>
</html>
```

Page html produite et  
envoyée au navigateur

Page PHP sur le serveur



# Éléments de langage

## Les balises PHP

Lorsque le fichier ne contient que du code PHP il n'est pas nécessaire de mettre la balise fermante «?>»

```
<?php  
var_dump($_GET);  
  
$texte = $_GET['texte'];  
$texte = trim($texte);  
if (empty($texte)) {  
    echo "$texte est non défini";  
}
```



# Eléments de langage

## Les Commentaires

Une ligne

// ceci est un commentaire

# ceci est un autre commentaire

Plusieurs lignes

/\*

Ceci est un commentaire  
de plusieurs lignes

\*/



# Éléments de langage

**Le code PHP** insensible à la casse echo = Echo

## Les Variables

\$nom\_variable / sensible à la casse \$a ≠ \$A

Typage: dynamique

integer

double (=float)

string

boolean

array (tableau)

object



# Eléments de langage

## Les Variables

Les variables **ne sont pas typées explicitement**, le type est déterminé automatiquement en fonction du contenu.

```
$age = 18;  
$poids = 72.5;  
$nom = 'contador';  
$prenom = 'alberto';  
$majeur = true;
```

Le type de variable peut être récupéré par la fonction: **getType()**

Ex: `echo 'le type de $age est: '.getType($age); //integer`

Pour des informations complètes sur la variable: **var\_dump()**

Ex: `var_dump($age); //integer(18)`



# Éléments de langage

## Les Variables

- Le typage d'une variable n'est pas définitif mais peut changer en cours d'exécution de programme.

```
$age = 18;
```

```
echo 'le type de $age est: ' . getType($age); //integer
```

```
$age = 'dix-huit';
```

```
echo 'le type de $age est: ' . getType($age); //string
```

- Le typage d'une variable peut être modifié explicitement par:
  - o La méthode **settype()** `settype($age, "string")`
  - o Un transtypage (**cast**) `$age = (string)$age;`



# Éléments de langage

## Les Variables de type string

Elles permettent de stocker des chaînes de caractères ou des caractères

```
$nom = 'Bob'; (ou "Bob")
```

```
$initiale = 'B'; (ou "B")
```

La différence entre guillemets doubles ou simples se fait à l'interprétation)

```
echo "Affichage $prenom  
$nom"; => Affichage Sébastien  
ROIZOT echo 'Affichage $prenom  
$nom'; => Affichage $prenom  
$nom
```





# Éléments de langage

Accès à un caractère d'une chaîne:

```
$initiale = $nom[0]; (ou $nom{0})
```

9

## Les Variables de type

### string Concaténation:

Réalisée par l'opérateur « . »

```
$nom = 'Bob';
```

```
$texte = 'Mon nom est ' . $nom . ' . ';
```

```
//Mon nom est Bob.
```

Il existe de nombreuses fonctions de manipulation de chaîne

```
Ex :substr(), trim(), strcmp(),...
```



# Éléments de langage

Voir : <http://php.net/manual/fr/ref.strings.php>

## Les Variables de type tableau (array)

La taille d'un tableau n'est pas figée, elle évolue dynamiquement

Création :

```
$tab = array();
```

ou

```
$tabNote = array(15,16,17);
```

```
$tab[] = 58; ajout valeur à la fin tableau
```

```
$tabNote = [15,16,17];
```

```
$tab[] = 70;
```



# Éléments de langage

Le contenu du tableau peut être vérifié

(débogage) par les fonctions `var_dump()` ou

`print_r()`

```
var_dump($tab) //array(2) { [0]=> int(58)
```

```
[1]=> int(70) }
```

## Les Variables de type tableau (array)

L'accès à un élément se fait par un indice  
numérique.

```
$ta
```

```
b =
```



# Éléments de langage

arr

ay(

15,

16,

17)

;

ec

ho

\$ta

b[1

]

//1

6



# Éléments de langage

Remplacement d'un élément d'un

tableau : `$tab[1]`

Insertion d'éléments n'importe où

dans un tableau:

```
array_splice( &array: $tab, offset: 1, length: 1, [40,41]);
```

en fin de tableau `$tab[ ]= 5` ou `array_push( &array: $tab, values: 12);`

Il existe de nombreuses fonctions de manipulation de tableau

Ex : `count()`, `sort()`,...



# Éléments de langage

## Les tableaux associatifs

Dans ces tableaux l'indice est remplacé par une clé représentée par une chaîne de caractères.

```
$tab = array();  
$tab ["nom"] = 'contador';  
$tab["prenom"] = 'alberto';  
$tab["age"] = 35;
```

```
//affichage  
echo($tab["prenom"]); //alberto
```

```
$coordonnees = [  
    'prenom' => 'François',  
    'nom' => 'Dupont',  
    'adresse' => '3 Rue du Paradis',  
    'ville' => 'Marseille'];
```



# Éléments de langage

## Parcours d'un tableau

*Tableau à indice numérique:*

```
$tabNotes = array(16,17,18);  
    for($i = 0 ; $i < count($tabNotes) ; $i++){  
        echo $tabNotes[$i].'<br/>';  
    }
```

Ou

```
foreach ($tabNotes as $index=> $element) {  
    echo ('index \' . $index . '\' -> valeur: ' . $element . '<br>');  
}
```



# Éléments de langage

## Parcours d'un tableau

### *Tableau associatif :*

```
$tab = array();  
$tab["nom"] = 'contador';  
$tab["prenom"] = 'alberto';  
$tab["age"] = 35;  
  
foreach($tab as $cle => $value){  
    echo $cle . ' = ' . $value . '<br/>';  
}
```

Affichage :

```
nom = contador  
prenom = alberto  
age = 35
```





# Éléments de langage

## Structures de contrôles

`if(...){.....}`

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

`else if`

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

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

`for ($i=0 ; $i<10; $i++){...}`

`foreach(...)`

`switch(..){...}`



# Eléments de langage

## Les fonctions:

*Pas besoin de typer le retour d'une fonction*

**Sans paramètre** : sans retour de valeur

```
function writeMsg() {  
    echo 'Hello world!';  
}  
writeMsg();
```

**Avec paramètres** : avec retour de valeur

```
function addNumbers($a, $b) {  
    return $a + $b;  
    echo addNumbers(5, 5);  
}
```