



Institut Supérieur des Technologies de l'Informations et de la Communication de Borj Cédria

Unité enseignement	Technologies et programmation web, Techniques Multimédia	Code U.E	520080
Elément d'enseignement	Programmation Web Dynamique	Code E.E	520101
Niveau	License Fondamentale en Sciences de l'Informatique	Classe	LFSI-3

CHAPITRE 2 : LE LANGAGE PHP

ELABORE PAR : AZZABI MOHAMED SEIFEDDINE



Année Universitaire 2017-2018

TABLE DES MATIERES

Chapitre 2 : Le Langage PHP	1
Elaboré par : Azzabi Mohamed Seifeddine	1
1 La syntaxe de base	3
1.1 Balises PHP	3
1.2 PHP et HTML.....	3
1.2.1 Echappement avancé en utilisant des conditions	3
1.3 Commentaires	3
2 Les types PHP	3
2.1 Exemples.....	4
2.2 Les chaines de caractères.....	4
2.2.1 Entourée de guillemets simples	4
2.2.2 Entourée de guillemets doubles.....	5
2.2.3 Analyse des variables	5
2.3 Les tableaux.....	5
2.3.1 Déclaration	5
2.3.2 Utilisation	5

1 LA SYNTAXE DE BASE

1.1 BALISES PHP

Lorsque PHP traite un fichier, il cherche les balises d'ouverture et de fermeture (`<?php` et `?>`) qui délimitent le code qu'il doit interpréter. De cette manière, cela permet à PHP d'être intégré dans toutes sortes de documents, car tout ce qui se trouve en dehors des balises ouvrantes / fermantes de PHP est ignoré.

```
<?php
echo "Bonjour le monde !";

// ... encore du code
?>
```

1.2 PHP ET HTML

Tout ce qui se trouve en dehors d'une paire de balises ouvrantes/fermantes est ignoré par l'analyseur PHP, ce qui permet d'avoir des fichiers PHP mixant les contenus. Ceci permet à PHP d'être contenu dans des documents HTML, pour créer par exemple des templates.

```
<p>Ceci sera ignoré par PHP et affiché au navigateur.</p>
<?php echo 'Alors que ceci sera analysé par PHP.'; ?>
<p>Ceci sera aussi ignoré par PHP et affiché au navigateur.</p>
```

1.2.1 Echappement avancé en utilisant des conditions

```
<?php if ($expression == true){ ?>
    Ceci sera affiché si l'expression est vrai.
<?php } else {?>
    Sinon, ceci sera affiché.
<?php } ?>
```

1.3 COMMENTAIRES

```
<?php
echo 'Ceci est un test'; // Ceci est un commentaire sur une seule ligne
, style c++
/* Ceci est un commentaire sur
   plusieurs lignes */
echo 'Ceci est un autre test';
echo 'Et un test final'; # Ceci est un commentaire style shell sur une
seule ligne
?>
```

2 LES TYPES PHP

PHP supporte 10 types basiques.

4 types scalaires :

- [boolean](#)
- [integer](#)
- [float](#) (nombre à virgule flottante, i.e. [double](#))
- [string](#)

4 types composés :

- [array](#)
- [object](#)
- [callable](#)
- iterable

Et finalement, 2 types spéciaux :

- [resource](#)
- [NULL](#)

2.1 EXEMPLES

```
<?php
$a_bool = TRUE; // un booléen
$a_str = "foo"; // une chaîne de caractères
$a_str2 = 'foo'; // une chaîne de caractères
$a_int = 12; // un entier

echo gettype($a_bool); // affiche : boolean
echo gettype($a_str); // affiche : string

// Si c'est un entier, incrément de 4
if (is_int($a_int)) {
    $a_int += 4;
}

// Si $a_bool est une chaîne de caractères, on l'affiche
if (is_string($a_bool)) {
    echo "String: $a_bool";
}
?>
```

2.2 LES CHAINES DE CARACTERES

2.2.1 Entourée de guillemets simples

La façon la plus simple de spécifier une [chaîne de caractères](#) est de l'entourer de guillemets simples (le caractère `'`).

```
<?php
$a="test" ;
echo 'ceci est une chaîne simple';

echo 'Vous pouvez également ajouter des nouvelles lignes
dans vos chaînes
de cette façon';

// Affiche : Arnold $a dit : "I'll be back"
echo 'Arnold $a dit : "I\'ll be back"';

// Affiche : Voulez-vous supprimer C:\*.*?
echo 'Voulez-vous supprimer C:\\*.*?';

// Affiche : Voulez-vous supprimer C:\*.*?
echo 'Voulez-vous supprimer C:\*.*?';

// Affiche : Ceci n'affichera pas \n de nouvelle ligne
echo 'Ceci n\'affichera pas \n de nouvelle ligne';
```

```
// Affiche : Les variables ne seront pas $traitees $ici
echo 'Les variables ne seront pas $traitees $ici';
?>
```

2.2.2 Entourée de guillemets doubles

Si la [chaîne de caractères](#) est entourée de guillemets doubles ("), PHP interprétera les séquences d'échappement suivantes pour les caractères spéciaux

Caractères échappés	
Séquence	Signification
<code>\n</code>	Fin de ligne (LF ou 0x0A (10) en ASCII)
<code>\r</code>	Retour à la ligne (CR ou 0x0D (13) en ASCII)
<code>\t</code>	Tabulation horizontale (HT or 0x09 (9) en ASCII)
<code>\v</code>	Tabulation verticale (VT ou 0x0B (11) en ASCII) (depuis PHP 5.2.5)
<code>\e</code>	Échappement (ESC or 0x1B (27) en ASCII) (depuis PHP 5.4.4)
<code>\f</code>	Saut de page (FF ou 0x0C (12) en ASCII) (depuis PHP 5.2.5)
<code>\\</code>	Antislash
<code>\\$</code>	Signe dollar
<code>\"</code>	Guillemet double
<code>\[0-7]{1,3}</code>	La séquence de caractères correspondant à cette expression rationnelle est un caractère, en notation octale, qui débordera silencieusement pour s'adapter à un octet (e.g. "\400" === "\000")
<code>\x[0-9A-Fa-f]{1,2}</code>	La séquence de caractères correspondant à cette expression rationnelle est un caractère, en notation hexadécimale
<code>\u{[0-9A-Fa-f]+}</code>	la séquence de caractères correspondant à l'expression régulière est un codepoint Unicode, qui sera la sortie de la chaîne de caractères représentant le codepoint UTF-8 (ajouté dans PHP 7.0.0)

2.2.3 Analyse des variables

Lorsqu'une [chaîne de caractères](#) est spécifiée entre guillemets doubles ou en Heredoc, les [variables](#) qu'elle contient sont interprétées.

2.3 LES TABLEAUX

2.3.1 Déclaration

```
<?php

// Déclaration d'un tableau associatifs
$array = [
    "foo" => "bar",
    "bar" => "foo",
];
?>
```

2.3.2 Utilisation

```
<?php

echo $array["foo"]; // Affichera bar

?>
```

