



Les opérateurs et les structures

Le langage PHP de base

El hadji Mamadou NGUER Enseignant chercheur en Informatique à l'UVS



Chapitre 3 : Les opérateurs et les structures

Objectifs spécifiques : A la suite de cette séquence, l'étudiant doit être capable de décrire et mettre en œuvre :

1. les opérateurs du langage PHP,
2. ses structures de contrôle,
3. et ses structures de boucle.

Chapitre 3 : Les opérateurs et les structures

Plan de la séquence :

1. Introduction
2. Les opérateurs
3. Les structures de contrôle
4. Les structures de boucle
5. Conclusion

Introduction

On a souvent besoin dans un programme informatique :

- d'effectuer des calculs en utilisant des expressions pouvant être composées d'opérandes et d'opérateurs,
- de faire des tests de contrôle,
- et de répéter des instructions en utilisant les boucles.

Nous allons dans cette séquence décrire et mettre en œuvre:

- les différents types opérateurs du langage PHP
- les différentes structures de contrôle
- les différentes structures de boucle

Les opérateurs

Les opérateurs permettent de faire des calculs et des comparaisons. Ils sont regroupés en :

- opérateurs arithmétiques
- opérateurs d'affectation
- opérateurs de chaînes
- opérateurs d'incrémentation / décrémentation
- opérateurs de comparaison
- opérateurs logiques
- opérateurs sur les tableaux
- etc.

Les opérateurs

Les opérateurs arithmétiques

- Ils permettent de faire des calculs arithmétiques
- Le tableau ci-dessous décrit les plus usuels

Opérateur	Nom	Exemple	Résultat
+	Addition	$\$x + \y	Somme de $\$x$ et $\$y$
-	Soustraction	$\$x - \y	Difference de $\$x$ et $\$y$
*	Multiplication	$\$x * \y	Produit de $\$x$ et $\$y$
/	Division	$\$x / \y	Quotient de $\$x$ et $\$y$
%	Modulo	$\$x \% \y	Reste de $\$x$ divisé par $\$y$

Les opérateurs

Les opérateurs d'affectation

- Un opérateur d'affectation est utilisé pour écrire une valeur sur une variable.
- L'opérateur d'affectation de base en PHP est "=". Cela signifie que l'opérande gauche se voit affecter la valeur de l'expression de droite.

Opérateur	Equivalent à...	Description
<code>x = y</code>	<code>x = y</code>	L'opérande de gauche prend la valeur de l'opérande de droite
<code>x += y</code>	<code>x = x + y</code>	Addition
<code>x -= y</code>	<code>x = x - y</code>	Soustraction
<code>x *= y</code>	<code>x = x * y</code>	Multiplication
<code>x /= y</code>	<code>x = x / y</code>	Division
<code>x %= y</code>	<code>x = x % y</code>	Modulo

Les opérateurs

Les opérateurs de chaînes

Opérateur	Nom	Exemple	Résultat
.	Concaténation	\$txt1 = "Hello" \$txt2 = \$txt1 . " world!"	Maintenant \$txt2 contient "Hello world!"
.=	Concaténation affectation	\$txt1 = "Hello" \$txt1 .= " world!"	Maintenant \$txt1 contient "Hello world!"

Les opérateurs d'incrémentation / décrémentation

Opérateur	Nom	Description
++\$x	Pre-incrémentation	Incrémente \$x de 1, et retourne \$x
\$x++	Post-incrémentation	Retourne \$x, et incrémente \$x de 1
--\$x	Pre-décrémentation	Décrémente \$x de 1, et retourne \$x
\$x--	Post-décrémentation	Retourne \$x, et décrémente \$x de 1

Les opérateurs

Les opérateurs de comparaison

Ils sont utilisés pour comparer deux valeurs (nombre ou chaîne):

Operateur	Nom	Exemple	Résultat
==	Egale	<code>\$x == \$y</code>	Vrai si \$x est egal à \$y
===	Identique	<code>\$x === \$y</code>	Vrai si \$x est egal à \$y, et sont de même type
!=	Different	<code>\$x != \$y</code>	Vrai si \$x est different de \$y
<>	Different	<code>\$x <> \$y</code>	Vrai si \$x est different de \$y
!==	Non identique	<code>\$x !== \$y</code>	Vrai si \$x est different de \$y, ou sont de type different
>	Superieur strict à	<code>\$x > \$y</code>	Vrai si \$x est superieur strict à \$y
<	Inferieur strict à	<code>\$x < \$y</code>	Vrai si \$x est inferieur strict à \$y
>=	Superieur ou egal à	<code>\$x >= \$y</code>	Vrai si \$x est superieur ou egal à \$y
<=	Inferieur ou égal à	<code>\$x <= \$y</code>	Vrai si \$x est inferieur ou egal à \$y

Les opérateurs

Les opérateurs logiques

Ils sont utilisés pour comparer des expressions logiques.

Opérateur	Nom	Exemple	Résultat
and	And	\$x and \$y	Vrai si \$x et \$y sont vrai
or	Or	\$x or \$y	Vrai si \$x est vrai ou \$y est vrai
xor	Xor	\$x xor \$y	Vrai si \$x est vrai ou \$y est vrai, pas les deux
&&	And	\$x && \$y	Vrai si \$x est vrai et \$y est vrai
	Or	\$x \$y	Vrai si \$x est vrai \$y est vrai
!	Not	!\$x	Vrai si \$x n'est pas vrai

Les opérateurs

Les opérateurs sur les tableaux

Ils sont utilisés pour comparer les tableaux:

Opérateur	Nom	Exemple	Résultat
+	Union	$\$x + \y	Union de $\$x$ et $\$y$ (mais les clés dupliquées ne sont pas réécrites)
==	Egalité	$\$x == \y	Vrai si $\$x$ et $\$y$ ont la même paire clef/valeur
===	Indentique	$\$x === \y	Vrai si $\$x$ et $\$y$ ont la même paire clef/valeur dans le même ordre et de même type
!=	Inégalité	$\$x != \y	Vrai si $\$x$ est différent de $\$y$
<>	Inégalité	$\$x <> \y	Vrai si $\$x$ est différent de $\$y$
!==	Non identique	$\$x !== \y	Vrai si $\$x$ n'est pas identique à $\$y$

Les structures de contrôle

- Les structures de contrôle permettent de faire des tests de contrôle dans un programme.
- Il en existe deux : La structure **if ... elseif ... else** et la structure **switch**.
- Syntaxes :

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

```
switch( ... ) {  
    case ... : { ... } break  
    case ... : { ... } break  
  
    ...  
    default : { ... }  
}
```

Les structures de contrôle

Exemple : (if)

```
<?php
    $t=date("H");
    if ($t<"10") {
        echo "Bonjour!";
    }elseif ($t<"20") {
        echo "Bonsoir!";
    }else {
        echo "bonne nuit!";
    }
?>
```

Les structures de contrôle

Exemple : (switch)

<?php

```
$choix="rouge ";
```

```
switch ($choix){
```

```
    case "rouge":
```

```
        echo "Votre couleur favorite est le rouge!";
```

```
        break;
```

```
    case "bleu":
```

```
        echo " Votre couleur favorite est le bleu!";
```

```
        break;
```

```
    case "vert":
```

```
        echo " Votre couleur favorite est le vert!";
```

```
        break;
```

```
    default:
```

```
        echo " Votre couleur favorite n'est ni le rouge, ni le bleu ni le vert!";
```

```
}
```

?>



Les structures de boucle

- Les structures de boucle permettent de répéter des instructions.
- Syntaxes :

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

Exemple : (while)

```
<?php  
    $x=1;  
    while($x<=5){  
        echo "Le nombre est: $x <br>";  
        $x++;  
    }  
?>
```

Les structures de boucle

Exemple : (do ... while)

```
<?php
$x=1;
do{
    echo "Le nombre est: $x <br>";
    $x++;
}
while ($x<=5)
?>
```

Exemple : (for)

```
<?php
for ($x=0; $x<=10; $x++){
    echo "Le nombre est: $x <br>";
}
?>
```


Les structures de boucle

La boucle foreach

- Elle fonctionne uniquement sur les tableaux et est utilisée pour parcourir chaque paire clé / valeur dans un tableau.
- **Syntaxe:**

```
foreach ($array as $value) {  
    code à executer  
}
```
- A chaque itération, la valeur de l'élément de tableau est affectée à \$value et le pointeur de tableau est déplacé d'un rang, jusqu'à ce qu'il atteigne le dernier élément du tableau.
- La boucle foreach sera étudiée dans la séquence consacrée aux tableaux.

Conclusion

- Au cours de cette séquence, nous avons vu :
 - les opérateurs du langage PHP
 - ses structures de contrôle,
 - et ses structures de boucle
- Pour consolider les connaissances acquises dans cette séquence, nous devez faire :
 - les tests de connaissance de la séquence
 - et les exercices de la fiche de TP