

Développement PHP

Partie 2 : Bases de PHP

DENIS LOEUILLET – IFA - 2017

Partie 2 : Bases de PHP

- Variables, types de variables
- Les instructions conditionnelles
- Les boucles
- les types de données composés : tableaux et objets
- Les fonctions
- Les constantes

Partie 2 : Bases de PHP

Les boucles

A quoi servent les boucles ?

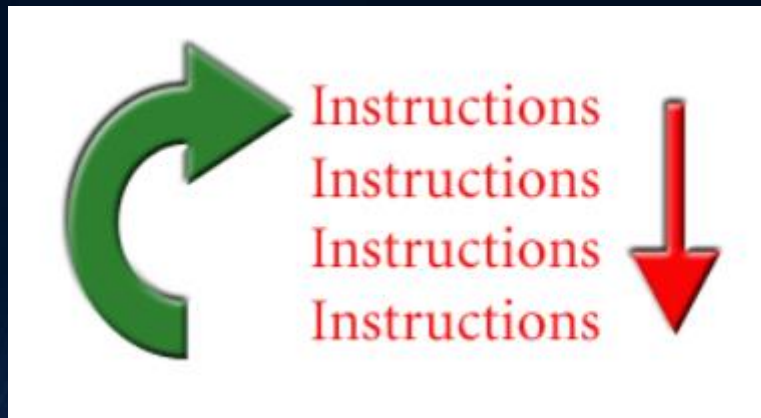
Les boucles sont des structures de contrôle permettant d'effectuer un certain nombre de fois les mêmes opérations. Concrètement, il s'agit d'une répétition d'instructions.

Par exemple afficher une suite de nombres que l'on incrémentera à chaque tour de boucle en fonction d'une ou plusieurs conditions à vérifier.

Partie 2 : Bases de PHP

Les boucles

Principe des boucles :



1. Comme d'habitude, les instructions sont exécutées dans l'ordre (de haut en bas)
2. À la fin des instructions, on retourne à la première ligne
3. On recommence à lire les instructions dans l'ordre
4. Et on retourne à la première
5. etc

Problème!! Comment arrêter l'exécution des instructions ?

Quel que soit le type de boucle, il faut prévoir des conditions d'exécution de la boucle.

Tant que les conditions sont remplies, les instructions sont exécutées.
Dès que les conditions ne sont plus remplies, on sort de la boucle.

Partie 2 : Bases de PHP

Les boucles

1. Une boucle simple : while
2. Une boucle plus complexe : for
3. La boucle do ... while
4. La boucle foreach
5. Les instructions d'arrêt et de continuité

Partie 2 : Bases de PHP

Les boucles

1. Une boucle simple : while

<http://php.net/manual/fr/control-structures.while.php>

- « while » peut se traduire par : tant que
- On dit à PHP : tant que CONDITION est vraie tu exécutes les instructions
- Les instructions à répéter se trouvent entre {}

Exemple :

```
1 <?php
2 $nombre_de_lignes = 1;
3
4 while ($nombre_de_lignes <= 100)
5 {
6     echo 'Je ne dois pas regarder les mouches voler quand j\'apprends le PHP.<br />';
7     $nombre_de_lignes++; // $nombre_de_lignes = $nombre_de_lignes + 1
8 }
9 ?>
```

Que va afficher ce code ?

Quand va-t-il s'arrêter?

Partie 2 : Bases de PHP

Les boucles : while

- La boucle pose la condition : TANT QUE `$nombre_de_lignes` est inférieur ou égal à 100.
- Dans cette boucle, il y a deux instructions :
 - Le `echo`, qui permet d'afficher du texte en PHP.
 - ensuite, une instruction bizarre : `$nombre_de_lignes++`;

Quésaco ? Regardez mon commentaire : c'est exactement la même chose. En fait, c'est une façon plus courte d'ajouter 1 à la variable.

On appelle cela l'incrémentation (ce nom barbare signifie tout simplement que l'on a ajouté 1 à la variable).

- Chaque fois qu'on fait une boucle, la valeur de la variable augmente : 1, 2, 3, 4... 99, 100...
- Dès que la variable atteint 101, on arrête la boucle

Partie 2 : Bases de PHP

Les boucles : while

Remarque :

Il faut TOUJOURS s'assurer que la condition sera fausse au moins une fois. Si elle ne l'est jamais, alors la boucle s'exécutera à l'infini !

PHP refuse normalement de travailler plus d'une quinzaine de secondes. Il s'arrêtera tout seul s'il voit que son travail dure trop longtemps et affichera un message d'erreur.

Partie 2 : Bases de PHP

Les boucles : while

Autre exemple :

```
1 <?php
2 $nombre_de_lignes = 1;
3
4 while ($nombre_de_lignes <= 100)
5 {
6     echo 'Ceci est la ligne n°' . $nombre_de_lignes . '<br />';
7     $nombre_de_lignes++;
8 }
9 ?>
```

Que va-t-il s'afficher ?

Modifiez l'instruction echo en introduisant du code HTML.

Partie 2 : Bases de PHP

Les boucles

<http://php.net/manual/fr/control-structures.for.php>

2. Une boucle plus complexe : for

- « for » est un autre type de boucle, plus condensée et plus commode à écrire, ce qui fait que « for » est assez fréquemment utilisé.
- « for » et « while » donnent le même résultat et servent à la même chose
 - Répéter des instructions en boucle
 - « for » ressemble beaucoup à « while »
 - Première ligne un peu particulière

Le même exemple :

```
1 <?php
2 for ($nombre_de_lignes = 1; $nombre_de_lignes <= 100; $nombre_de_lignes++)
3 {
4     echo 'Ceci est la ligne n°' . $nombre_de_lignes . '<br />';
5 }
6 ?>
```

Partie 2 : Bases de PHP

Les boucles : for

```
1 <?php
2 for ($nombre_de_lignes = 1; $nombre_de_lignes <= 100; $nombre_de_lignes++)
3 {
4     echo 'Ceci est la ligne n°' . $nombre_de_lignes . '<br />';
5 }
6 ?>
```

- Après le mot « for », il y a des parenthèses qui contiennent trois éléments, séparés par des points-virgules
 - Le premier sert à l'initialisation.
C'est la valeur que l'on donne au départ à la variable (ici, elle vaut 1).
 - Le deuxième, c'est la condition.
Comme pour le « while », tant que la condition est remplie, la boucle est réexécutée. Dès que la condition ne l'est plus, on en sort.
 - Enfin, le troisième c'est l'incrémentation, qui vous permet d'ajouter 1 à la variable à chaque tour de boucle.

Partie 2 : Bases de PHP

Les boucles : for

- Les deux codes donnent donc exactement le même résultat.
- Le « for » fait la même chose que le « while », mais rassemble sur une seule ligne tout ce qu'il faut savoir sur le fonctionnement de la boucle.
- La boucle « while » est plus simple et plus flexible : on peut faire tous les types de boucles avec mais on peut oublier de faire certaines étapes comme l'incrémentation de la variable.
- « for » est bien adapté quand on doit compter le nombre de fois que l'on répète les instructions et il permet de ne pas oublier de faire l'incrémentation pour augmenter la valeur de la variable !
- Si vous hésitez entre les deux, il suffit simplement de vous poser la question suivante : « Est-ce que je sais d'avance combien de fois je veux que mes instructions soient répétées ? ».
 - Si la réponse est oui, alors la boucle « for » est tout indiquée.
 - Sinon, alors il vaut mieux utiliser la boucle « while ».

Partie 2 : Bases de PHP

Les boucles

3. La boucle do ... while

- L'instruction « do » est associée au terme « while » (qui est parfois baptisé « until » dans d'autres langages que PHP).
- Répéter / faire ... tant que
- Similaire à boucle « while » sauf :
 - Les instructions sont exécutées au moins une fois
 - La condition n'est testée pour la première fois qu'après exécution des instructions

Syntaxe :

```
do {  
    Bloc d'instructions  
}  
while (condition);
```


Partie 2 : Bases de PHP

Les boucles : do ... while

<http://php.net/manual/fr/control-structures.do.while.php>

Exemple :

```
<?php
$i = 5;
do {
    echo $i;
    $i++;
} while ($i<9);
?>
```

- Que va afficher ce code ?
- Ce type de boucle est généralement peu utiliser

- Nous commençons par initialiser le compteur à la valeur 5.
- Puis nous définissons la liste d'instruction à exécuter.
- Nous terminons par le test de la condition permettant de sortir ou non de la boucle.

Remarque : si nous avons initialisé le compteur `$i` à une valeur inférieure ou égale à 9, alors le programme aurait calculé et affiché 9 puis serait sorti de la boucle car la condition est fausse dès le début.

Partie 2 : Bases de PHP

Les boucles

5. Les instructions d'arrêt et de continuité

- 2 instructions particulières des boucles
 - continue : force le passage à l'itération suivante
 - Break : force à quitter une structure conditionnelle telle que for, while, foreach, switch
- Ces deux instructions peuvent prendre un paramètre optionnel de type entier permettant de connaître le nombre de structures emboîtées qui ont été interrompues.

Partie 2 : Bases de PHP

Les boucles : les instructions d'arrêt et de continuité

Exemple :

Exemple d'utilisation de break

```
<?php
for($i=0; $i<=10; $i++)
{
    // On affiche la valeur du compteur
    echo 'Compteur = ';
    // On sort de la boucle si on atteint le chiffre 5
    if($i>5)
    {
        break;
    }
    // Affichage des nombres de 0 à 5
    echo $i , '<br/>';
}
?>
```

- On demande de sortir de la boucle dès que l'on a atteint le chiffre 5

Exemple :

Exemple d'utilisation de continue

```
<?php
for($i=0; $i<=10; $i++)
{
    // On n'affiche pas les 5 premiers nombres
    if($i<=5)
    {
        continue;
    }
    // Affichage des nombres de 6 à 10
    echo $i , '<br/>';
}
?>
```

- On commence à itérer depuis la valeur 0 mais nous ne voulons afficher que les valeurs comprises de 6 à 10.
- Grâce à la condition et au mot-clé continue, nous parvenons à ne pas afficher les 6 premières valeurs.
- nous aurions pu obtenir le même résultat sans utiliser le mot-clé continue en remplaçant la condition du if() par \$i > 5 et en plaçant l'instruction echo() à la place du continue.

Partie 2 : Bases de PHP

Les boucles : résumé

- Les boucles demandent à PHP de répéter des instructions plusieurs fois.
- Les deux principaux types de boucles sont :
 - while : à utiliser de préférence lorsqu'on ne sait pas par avance combien de fois la boucle doit être répétée ;
 - for : à utiliser lorsqu'on veut répéter des instructions un nombre précis de fois.
- L'incrémentación est une technique qui consiste à ajouter 1 à la valeur d'une variable. La décrémentation retire au contraire 1 à cette variable. On trouve souvent des incrémentations au sein de boucles for.
- La boucle foreach permet de parcourir les éléments d'un tableau