

PHP: Boucles et fonctions utiles



Sommaire



- Les différentes boucles.
- Contrôle des boucles.
- Parcours d'un tableau.
- Le manuel PHP.
- Les fonctions de test.
- Les fonctions de date et d'heure.

Les boucles: for



```
<?php
    for ($i=0; $i<=10; $i++)
    {
        echo $i."<br/>";
    }
?>
```

Les boucles: while



```
<?php
    $i=0;
    while ($i<=10)
    {
        echo $i++."<br/>";
    }
?>
```

Les boucles: do...while



```
<?php
    $u = 0;
    do
    {
        echo $u++."<br/>";
    }
    while ($u<=10) ;
?>
```

Exercice pratique: boucles



Parcours d'un tableau



```
<?php
    $tMarques = array('Peugeot', 'Renault', 'Dacia', 'Citroën', 'Seat',
                      'Lada', 'Skoda', 'VW', 'Audi', 'Land Rover');

    echo count($tMarques). '<br/>';
    echo "<ul>";
    foreach($tMarques as $marque)
    {
        echo "<li>".$marque."</li>";
    }
    echo "</ul>";
?>
```

Parcours d'un tableau



- PHP permet d'utiliser des tableaux associatifs, c'est-à-dire des tableaux dont les indices ont un nom et non juste un numéro.

```
<?php
    $tUser = array('nom'=>'TAVERNIER', 'prenom'=>'Bruno');
    echo $tUser['nom'];
    echo "<ul>";
    foreach($tUser as $index=>$marque)
    {
        echo "<li><strong>".$index."</strong>: ".$marque."</li>";
    }
    echo "</ul>";
?>
```


Parcours d'un tableau



```
<?php
    $tNoms = array('TAVERNIER', 'DUPONT',
                   'DURAND', 'MARTIN');

    //affiche le contenu du tableau...
    echo '<pre>'.print_r($tNoms, true).'
```

Exercice pratique: Tableaux et boucles



Contrôle d'un tableau



➤ La fonction `continue` permet de sauter un tour dans une boucle.

```
if ($a%2=1) continue;
```

➤ La fonction `break` permet de quitter une boucle.

```
if ($a==10) break;
```

Le manuel PHP



`http://php.net/manual/fr/`

The screenshot shows the PHP manual page for the `foreach` loop in French. The page is part of the 'Les structures de contrôle' (Control Structures) section. It includes a navigation bar at the top with links for Downloads, Documentation, Get Involved, and Help, along with a search bar. The main content area is titled 'foreach' and includes a 'Change language' dropdown set to 'French'. The text explains that `foreach` is used for iterating over arrays and objects, and provides two syntaxes: one for values and one for key-value pairs. A 'Note' section at the bottom states that the internal pointer is automatically reset to the first element after the loop ends. A right-hand sidebar lists other control structures: Introduction, if, else, elseif/else if, Syntaxe alternative, while, do-while, for, foreach (highlighted), break, continue, switch, declare, return, require, include, require_once, include_once, and goto.

Manuel PHP > Référence du langage > Les structures de contrôle

« for break »

Change language: French Edit Report a Bug

foreach

Ouvrir dans un nouvel onglet

(PHP 4, PHP 5)

La structure de langage *foreach* fournit une façon simple de parcourir des tableaux. *foreach* ne fonctionne que pour les tableaux et les objets, et émettra une erreur si vous tentez de l'utiliser sur une variable de type différent ou une variable non initialisée. Il existe deux syntaxes :

```
foreach (array_expression as $value) {  
    //commandes  
}  
  
foreach (array_expression as $key => $value) {  
    //commandes  
}
```

La première forme passe en revue le tableau *array_expression*. À chaque itération, la valeur de l'élément courant est assignée à *\$value* et le pointeur interne de tableau est avancé d'un élément (ce qui fait qu'à la prochaine itération, on accèdera à l'élément suivant).

La seconde forme assignera en plus la clé de l'élément courant à la variable *\$key* à chaque itération.

Il est possible de personnaliser [l'itération sur des objets](#).

Note:
Lorsque *foreach* démarre, le pointeur interne du tableau est automatiquement ramené au premier élément du tableau. Cela signifie que vous n'aurez pas à faire appel à [reset\(\)](#) avant *foreach*.

Les structures de contrôle

- Introduction
- if
- else
- elseif/else if
- Syntaxe alternative
- while
- do-while
- for
- » **foreach**
- break
- continue
- switch
- declare
- return
- require
- include
- require_once
- include_once
- goto

Les fonctions de test



- `empty()` permet de vérifier qu'une variable ou un élément d'un tableau existe ou contient une valeur non nulle.
- `isset()` permet de vérifier si une variable ou un élément d'un tableau est définie.
- `unset()` permet de supprimer une variable ou un élément d'un tableau.
- `print_r()` permet d'afficher un tableau (avec la balise `<pre>` pour plus de lisibilité).
- `var_dump()` permet d'afficher un tableau avec le détail des types.

Exercice pratique: Fonctions de test



Dates et heures



En PHP, les dates sont calculées à partir d'un timestamp Unix, c'est-à-dire le nombre de secondes écoulées depuis le 01 Janvier 1970 minuit.

```
<?php
echo date('d/m/Y H:i:s'); //affiche la date

echo date('d/m/Y H:i:s', '123456789'); //affiche la
date correspondant au timestamp fourni

echo time(); //affiche le timestamp
?>
```

Dates et heures



La fonction `strtotime()` retourne un timestamp à partir d'une date exprimée en anglais usuel.

```
<?php
    echo date('l', strtotime('2 October 2015'));
?>
```


Dates et heures



La fonction `microtime()` retourne un timestamp avec les microsecondes. Elle est pratique pour mesurer le temps d'exécution d'un script.

```
<?php
    $debut = microtime(true);
    ...
    ...
    $fin = microtime(true);
    echo ($fin-$debut);
?>
```

Dates et heures



La fonction `mktime()` retourne le timestamp d'une date. Les paramètres en sont pas obligés d'être valides.

```
<?php
    $timestamp = mktime($heure, $minute, $seconde,
                        $mois, $jour, $annee);
    echo date('l', $timestamp);
?>
```

Exercice pratique

Avec les fonctions dates et les tableaux, afficher la date en français.

