

Développement web côté serveur avec PHP

(index.html)

les boucles

PHP: while - Manual (<http://php.net/manual/fr/control-structures.while.php>).

PHP: do-while - Manual (<http://php.net/manual/fr/control-structures.do.while.php>).

PHP: for - Manual (<http://php.net/manual/fr/control-structures.for.php>).

PHP: foreach - Manual (<http://php.net/manual/fr/control-structures.foreach.php>).

PHP: Les structures de contrôle - Manual (<http://php.net/manual/fr/language.control-structures.php>).

Note : le but premier des boucles est de pouvoir répéter des instructions. Elles permettent aussi de parcourir un tableau et de répéter des instructions sur chacun des éléments.

- while
- do while
- for
- foreach

[code/while1.php \(code/while1.php\)](#)

```
1  <?php
2  // initialisation d'une variable qui servira d'index / compteur
3  $i = 0;
4
5  // itération d'une boucle de type `while` de 0 à 10 (non inclus)
6  while ($i < 10) {
7      // affichage de la variable
8      echo $i;
9      echo PHP_EOL;
10
11     // incrémentation de la variable
12     $i++;
13 }
```

[code/while2.php \(code/while2.php\)](#)

```
1 <?php
2 // initialisation d'une variable qui servira d'index / compteur
3 $i = 0;
4
5 // itération d'une boucle de type `while` tant que la variable `$boucler == true`
6 while (true) {
7     // affichage de la variable `$i`
8     echo $i;
9     echo PHP_EOL;
10
11     // incrémentation de la variable `$i`
12     $i++;
13
14     // vérification de la valeur de la variable `$i`
15     if ($i == 10) {
16         // on arrête la boucle avec le mot clé `break`
17         break;
18     }
19 }
```

[code/while3.php \(code/while3.php\)](#)

```
1 <?php
2 // initialisation d'une variable qui servira d'index / compteur
3 $i = 0;
4 // initialisation d'une variable qui servira à contrôler la boucle
5 $boucler = true;
6
7 // itération d'une boucle de type `while` tant que la variable `$boucler == true`
8 while ($boucler) {
9     // affichage de la variable `$i`
10    echo $i;
11    echo PHP_EOL;
12
13    // incrémentation de la variable `$i`
14    $i++;
15
16    // vérification de la valeur de la variable `$i`
17    if ($i == 10) {
18        // en affectant la valeur `false` à la variable `$boucle`, on arrête la boucle
19        $boucler = false;
20    }
21 }
```

[code/while4.php \(code/while4.php\)](#)

```
1  <?php
2  // initialisation d'une variable qui servira d'index / compteur
3  $i = 0;
4
5  // la boucle ne sera jamais exécutée
6  while (false) {
7      echo $i;
8      echo PHP_EOL;
9      $i++;
10 }
```

[code/while5.php \(code/while5.php\)](#)

```
1  <?php
2  // exemple d'utilisation de la boucle `while` pour récupérer l'entrée d'un utilisateur
3
4  require 'lib-user-input.php';
5
6  $loop = true;
7
8  while ($loop) {
9      $age = user_input_get_answer('quel âge avez-vous ?');
10
11      if (is_numeric($age)) {
12          echo "vous avez $age ans";
13          echo PHP_EOL;
14          $loop = false;
15      } else {
16          echo "vous n'avez pas répondu à la question";
17          echo PHP_EOL;
18      }
19 }
```

[code/do_while1.php \(code/do_while1.php\)](#)

```
1  <?php
2  // initialisation d'une variable qui servira d'index / compteur
3  $i = 0;
4
5  // début de la boucle
6  do {
7      // instructions répétées dans la boucle
8      // exemple : affichage du compteur
9      echo $i;
10     echo PHP_EOL;
11
12     // incrémentation du compteur
13     $i++;
14
15 } while ($i < 10); // vérification de la condition pour continuer ou non la boucle
```

[code/do_while2.php \(code/do_while2.php\)](#)

```
1  <?php
2  // initialisation d'une variable qui servira d'index / compteur
3  $i = 0;
4
5  // initialisation d'une variable qui servira de condition pour continuer ou non la bouc
6  $boucler = true;
7
8  // La boucle sera exécutée au moins une fois
9  // début de la boucle
10 do {
11     // instructions répétées dans la boucle
12     // exemple : affichage du compteur
13     echo $i;
14     echo PHP_EOL;
15
16     // incrémentation du compteur
17     $i++;
18
19     // vérification de la valeur de l'index / compteur
20     if ($i == 10) {
21         // si l'index / compteur est à 10, la boucle s'arrête
22         $boucler = false;
23     }
24
25 } while ($boucler); // vérification de la condition pour continuer ou non la boucle
```

[code/do_while3.php \(code/do_while3.php\)](#)

```
1  <?php
2  // initialisation d'une variable qui servira d'index / compteur
3  $i = 0;
4
5  // initialisation d'une variable qui servira de condition pour continuer ou non la bouc
6  $boucler = false;
7
8  // La boucle sera exécutée au moins une fois
9  // début de la boucle
10 do {
11     // instructions répétées dans la boucle
12     // exemple : affichage du compteur
13     echo $i;
14     echo PHP_EOL;
15
16     // incrémentation du compteur
17     $i++;
18
19 } while ($boucler); // vérification de la condition pour continuer ou non la boucle
```

[code/for1.php \(code/for1.php\)](#)

```
1 <?php
2 // initialisation d'une variable nommée `$i` qui servira d'index / compteur
3 // la boucle for se terminera quand `$i` vaudra 10
4 // la variable `$i` est incrémentée de 1 à chaque tour
5 for ($i = 0; $i < 10; $i++) {
6     // à chaque tour on affiche la valeur de `$i`
7     echo $i;
8     echo PHP_EOL;
9 }
```

[code/for2.php \(code/for2.php\)](#).

```
1 <?php
2 // la variable nommée `$liste` contient un tableau simple de trois éléments
3 $liste = array(
4     'foo',
5     'bar',
6     'baz'
7 );
8
9 // initialisation d'une variable nommée `$i` qui servira d'index / compteur
10 // la boucle for se terminera quand `$i` vaudra `count($liste)`, c-à-d quand `$i` sera
11 // la variable `$i` est incrémentée de 1 à chaque tour
12 for ($i = 0; $i < count($liste); $i++) {
13     // tour à tour, on affiche chaque élément de la variable `$liste`
14     echo $liste[$i];
15     echo PHP_EOL;
16 }
```

[code/for3.php \(code/for3.php\)](#).

```
1 <?php
2 // La variable nommée `$tableau` contient plusieurs tableaux simples
3 // La variable nommée `$tableau` est un tableau à deux dimensions
4 $tableau = array(
5     array('1', 'php'),
6     array('2', 'html'),
7     array('3', 'css'),
8     array('4', 'javascript'),
9     array('5', 'sql'),
10 );
11
12 // initialisation d'une variable nommée `$i` qui servira d'index / compteur
13 // La boucle for se terminera quand `$i` vaudra `count($tableau)`, c-à-d quand `$i` ser
14 // La variable `$i` est incrémentée de 1 à chaque tour
15 for ($i = 0; $i < count($tableau); $i++) {
16     // tour à tour, on affecte chaque élément de la variable `$tableau` à la variable `
17     $ligne = $tableau[$i];
18
19     // initialisation d'une variable nommée `$j` qui servira d'index / compteur
20     // La boucle for se terminera quand `$j` vaudra `count($ligne)`, c-à-d quand `$j` s
21     // La variable `$j` est incrémentée de 1 à chaque tour
22     for ($j = 0; $j < count($ligne); $j++) {
23         // tour à tour, on affiche chaque élément de la variable `$ligne`
24         echo $ligne[$j];
25         echo PHP_EOL;
26     }
27 }
```

[code/foreach1.php](#) ([code/foreach1.php](#))

```
1 <?php
2 // La variable nommée `$liste` contient un tableau simple de trois éléments
3 $liste = array(
4     'foo',
5     'bar',
6     'baz'
7 );
8
9 // début de la boucle
10 // à chaque tour, php affecte la valeur de chaque élément de la variable `$liste` à la
11 foreach ($liste as $element) {
12     // affichage de la variable `$element`
13     echo $element;
14     echo PHP_EOL;
15 }
```

[code/foreach2.php](#) ([code/foreach2.php](#))

```

1  <?php
2  // la variable nommée ` $liste ` contient un tableau associatif de trois éléments
3  $liste = array(
4      'artichaut' => 3,
5      'broccoli' => 2,
6      'carotte' => 5,
7  );
8
9  // début de la boucle
10 // à chaque tour, php affecte :
11 // - la clé de chaque élément de la variable ` $liste ` à la variable ` $cle `
12 // - la valeur de chaque élément de la variable ` $liste ` à la variable ` $valeur `
13 foreach ($liste as $cle => $valeur) {
14     // affichage des variables ` $cle ` et ` $valeur `
15     echo "$cle : $valeur";
16     echo PHP_EOL;
17 }

```

[code/foreach3.php \(code/foreach3.php\)](#)

```

1  <?php
2  // la variable nommée ` $liste ` contient un tableau simple de trois éléments
3  $liste = array(
4      'foo',
5      'bar',
6      'baz'
7  );
8
9  // début de la boucle
10 // à chaque tour, php affecte :
11 // - la clé de chaque élément de la variable ` $liste ` à la variable ` $cle `
12 // - la valeur de chaque élément de la variable ` $liste ` à la variable ` $valeur `
13 foreach ($liste as $cle => $valeur) {
14     // affichage des variables ` $cle ` et ` $valeur `
15     // on remarque que php avait automatiquement affecté des nombres entiers en commençant à 0
16     echo "$cle : $valeur";
17     echo PHP_EOL;
18 }

```

[code/foreach4.php \(code/foreach4.php\)](#)

```
1 <?php
2 // La variable nommée `$tableau` contient plusieurs tableaux associatifs
3 // La variable nommée `$tableau` est un tableau à deux dimensions
4 $tableau = array(
5     array('id' => '1', 'nom' => 'php'),
6     array('id' => '2', 'nom' => 'html'),
7     array('id' => '3', 'nom' => 'css'),
8     array('id' => '4', 'nom' => 'javascript'),
9     array('id' => '5', 'nom' => 'sql'),
10 );
11
12 // début de la boucle
13 // à chaque tour, php affecte la valeur de chaque élément de la variable `$tableau` à l
14 foreach ($tableau as $ligne) {
15     // début de la boucle
16     // à chaque tour, php affecte :
17     // - la clé de chaque élément de la variable `$ligne` à la variable `$cle`
18     // - la valeur de chaque élément de la variable `$ligne` à la variable `$valeur`
19     foreach ($ligne as $cle => $valeur) {
20         // affichage des variables `$cle` et `$valeur`
21         echo "$cle : $valeur";
22         echo PHP_EOL;
23     }
24 }
```

