



PHP

Les boucles



Dans ce module

- ❑ Boucle While
- ❑ Boucle For
- ❑ Boucle Foreach
- ❑ Scope des variables des boucles For
- ❑ Contrôle de boucle : Break et Continue
- ❑ Imbrications
- ❑ Pratique

Boucle While

- Une boucle sert à répéter un nombre limité ou infini de fois un code. Chaque répétition de code s'appelle une « **itération** ».
- « **while** » et « **do ... while** » servent à répéter infiniment un code tant que la **CONDITION** de la boucle est vraie.
- Les données de chaque itération ne sont pas perdues tant que la boucle tourne. Chaque itération génère des données nouvelles. Le problème de ce type de boucle est donc de gérer la **CONDITION** car tant que celle-ci est vraie la boucle ne s'arrête pas, engendrant des données croissantes et impliquant une limite software ou hardware provoquant l'arrêt forcé du programme ou un blocage de la machine.
- Rarement utilisé en PHP car trop risqué pour un serveur. Généralement utilisé lorsque la limite ne peut pas être connue.

```
while(CONDITION) {  
    // Repeated CODE if CONDITION true  
}
```

Code répété indéfiniment tant que la **CONDITION** est vraie

```
do {  
    // CODE at least once  
    // Repeated CODE if CONDITION true  
} while(CONDITION);
```

Exécuté au moins une fois puis répété indéfiniment tant que la **CONDITION** est vraie

Boucle While - Exemple

```
// Display a multiplication table

$tableChosen = 5;
$count = 0;

while($count <= 10) {
    $result = $tableChosen * $count;
    echo $tableChosen.' x '.$count.' = '.$result.'  
';
    $count++;
}
```

Résultat navigateur :

5 x 0 = 0
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50

Boucle For

- Une boucle « for » est limité par un nombre d'itération précis.
- On défini une variable (ex: « \$i ») qui définira le nombre d'itération de façon statique ou dynamique comme suit :
 1. *START: On défini une valeur de départ à « \$i » (généralement numérique).*
 2. *LIMIT: On défini une condition limite que « \$i » doit atteindre (ex: « \$i <= 10 »).*
 3. *VARIATOR: On défini la façon dont « \$i » par atteindre la LIMIT (ex: \$i++).*
- Très utilisé en PHP.

```
for (START; LIMIT; VARIATOR) {  
    // CODE as long as LIMIT is not reached  
}
```

Boucle For - Exemple

```
// Display a multiplication table

$tableChosen = 5;

for ($i = 0; $i <= 10; $i++) {
    $result = $tableChosen * $i;
    echo $tableChosen.' x '.$i.' = '.$result.'  
';
}
```

Résultat navigateur :

5 x 0 = 0
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50

Boucle Foreach

- Une boucle « foreach » sert à boucler sur une variable de type énumérable comme les Tableaux/Listes et certaines Classes.
 - *Une variable de type énumérable est une variable qui contient aucune ou autant de valeur (élément) de l'on veut. Exemple du contenu d'une variable « tableau » : [0, 1, 2, 3, 4]. En PHP, chaque élément d'un type énumérable peut être de n'importe quelle type. Exemple du contenu d'un « tableau » de string : ['pomme', 'fraise', 'poire', 'orange']*
- Dans le cas d'un serveur web, il s'agira souvent de données de la base de donnée que l'on va contenir dans un type énumérable comme un Tableau (array) ou une Classe dite « arrayable ».
- Chaque itération de la boucle sera une valeur de la variable énumérateur du premier au dernier éléments.
- Le plus utilisé en PHP.

```
foreach (ENUMERATOR as $RECEIVER) {  
    // CODE as long as ENUMERATOR has elements  
}
```

Boucle Foreach - Exemple

```
// Display fruit names

$fruits = ['pomme', 'fraise', 'poire', 'orange'];

foreach ($fruits as $fruit) {
    echo $fruit.'<br>';
}
```

Résultat navigateur :

pomme
fraise
poire
orange

Scope des variables des boucles For

- Dans le cas d'une boucle « for » ou « foreach », la variable d'itération (ex: « \$i », « \$fruit », ...) créer dans les parenthèses remplace (modifie) toute variable déjà existante qui aurait été créé avant la boucle et qui porterai exactement le même nom.
- La variable d'itération existera dans tous les cas même après la boucle et aura la valeur de la dernière itération (après VARIATOR dans le cas d'une boucle « for »).

Scope des variables des boucles For - Exemple

```
$fruits = ['pomme', 'fraise', 'poire', 'orange'];  
$fruit = 'mangue';  
  
echo 'Valeur de $fruit avant : '.$fruit.'<br>';  
  
foreach ($fruits as $fruit) {  
    echo '- '.$fruit.'<br>';  
}  
  
echo 'Valeur de $fruit après : '.$fruit;
```

Résultat navigateur :

Valeur de \$fruit avant : mangue

- pomme
- fraise
- poire
- orange

Valeur de \$fruit après : orange

```
$tableChosen = 5;  
$i = 50;  
  
echo 'Valeur de $i avant : '.$i.'<br>';  
  
for ($i = 0; $i <= 10; $i++) {  
    $result = $tableChosen * $i;  
    echo $tableChosen.' x '.$i.' = '.$result.'<br>';  
}  
  
echo 'Valeur de $i après : '.$i;
```

Résultat navigateur :

Valeur de \$i avant : 50

5 x 0 = 0

5 x 1 = 5

5 x 2 = 10

...

5 x 9 = 45

5 x 10 = 50

Valeur de \$i après : 11

Contrôle de boucle : Break et Continue

- Il existe 2 actions particulière utilisables n'importe où dans le code d'une boucle pour:
 - *Arrêter complètement la boucle à un endroit/moment précis*
 - *Passer à l'itération suivante*
- « break; » permet l'arrêt total de la boucle
- « continue; » permet de passer à l'itération suivante en ignorant le reste de l'itération en cours

Contrôle de boucle : Break et Continue - Exemple

```
$tableChosen = 5;

for ($i = 0; $i <= 10; $i++) {
    $result = $tableChosen * $i;

    if($result % 2 == 0) {
        continue;
    }

    echo $tableChosen.' x '.$i.' = '.$result.'<br>';
}
```

Résultat navigateur :

5 x 1 = 5
5 x 3 = 15
5 x 5 = 25
5 x 7 = 35
5 x 9 = 45

```
$fruits = ['pomme', 'fraise', 'poire', 'orange'];
$searched = 'poire';

foreach ($fruits as $fruit) {
    if($fruit == $searched) {
        echo "Le fruit 'poire' existe dans le tableau.";
        break;
    }
}
```

Résultat navigateur :

Le fruit 'poire' existe dans le tableau.

Imbrications

- Il est possible d'avoir autant de boucles dans des boucles que l'on veut.

```
foreach (ENUMERATOR as $RECEIVER) {  
    // CODE as long as ENUMERATOR has elements  
    for (START_1; LIMIT_1; VARIATOR_1) {  
        // CODE as long as LIMIT_1 is not reached  
        for (START_2; LIMIT_2; VARIATOR_2) {  
            // CODE as long as LIMIT_2 is not reached  
        }  
    }  
}
```

Pratique

1. Réalisez l’affichage n°1. Créez une variable « \$star » contenant un nombre d’étoile et avec une boucle « for » réalisez l’affichage. Vous devez prendre en compte que le nombre d’étoile varie selon la valeur de « \$star ». S « \$star = 8 », il doit y avoir 8 étoiles au début et 1 étoile à la fin. Vous aurez besoin de « str_repeat() ».
2. Réalisez l’affichage n°2 avec une boucle « for ». Espace HTML :
3. En utilisant un « foreach », générez une liste de liens HTML (<a ...>) à partir du tableau suivant : `['google.com', 'php.net', 'wordpress.com']` (recopiez-le). Aidez-vous des exemples du cours.
4. Réalisez un programme générant automatiquement un tableau HTML avec un nombre de colonne et de ligne défini par une variable. (Astuce: Vous aurez besoin d’imbriquer une boucle « for » dans une autre boucle « for »).

N°1

```
$star = 5;  
Résultat :  
*****  
****  
***  
**  
*
```

N°2

```
$star = 5;  
Résultat :  
 *  
  *  
   *  
  *  
 *  
*
```



PHP

Les boucles

Fin du module

