

PHP : Les boucles while et do...while

Dans ce cours, nous allons explorer en détail deux types de boucles en PHP : la boucle `while` et la boucle `do...while`. Tu vas découvrir comment elles fonctionnent, quand les utiliser, et voir des exemples concrets pour mieux comprendre leur utilité dans la programmation.

Objectifs de la leçon

- Comprendre le fonctionnement de la boucle `while`.
- Découvrir les spécificités de la boucle `do...while`.
- Savoir choisir entre ces deux types de boucles selon tes besoins.
- Mettre en pratique les concepts avec des exemples concrets.

La Boucle `while`

La boucle `while` exécute un bloc d'instructions tant que la condition spécifiée est vraie. Elle teste la condition **avant** chaque exécution du bloc, ce qui signifie que si la condition est fautive dès le départ, le code à l'intérieur de la boucle ne sera jamais exécuté.

Exemple concret

Imaginons que tu veuilles afficher les nombres de 0 à 4. Voici comment tu peux le faire avec une boucle `while` :

```
<?php
1  $i = 0;
2  while ($i < 5) {
3      echo "Le nombre est : $i<br>";
4      $i++; // Incrémente $i pour éviter une boucle infinie
5  }
?>
```

Explications :

- **Initialisation** : La variable `$i` est initialisée à 0.
- **Condition** : La boucle continue tant que `$i` est inférieur à 5.
- **Incrémentation** : À chaque passage dans la boucle, `$i` est augmenté de 1 pour atteindre la condition d'arrêt.

La Boucle `do...while`

La boucle `do...while` fonctionne de façon similaire à la boucle `while`, mais avec une différence clé : le bloc d'instructions est exécuté **au moins une fois** avant que la condition soit vérifiée. Cela peut être utile lorsque tu veux t'assurer que le code s'exécute au moins une fois, même si la condition est fautive dès le début.

Exemple concret

Supposons que tu souhaites afficher le nombre 0, puis vérifier la condition. Voici comment tu utiliserais la boucle `do...while` :

```
<?php
1  $i = 0;
2  do {
3      echo "Le nombre est : $i<br>";
4      $i++; // Incrémente $i
5  } while ($i < 5);
?>
```

Explications :

- **Exécution initiale** : Le code à l'intérieur du `do` s'exécute une première fois, même si `$i` ne remplit pas la condition.
- **Vérification après l'exécution** : La condition `$i < 5` est vérifiée **après** l'exécution du bloc, assurant que le code s'exécute toujours au moins une fois.

Comparaison et Bonnes Pratiques

- **Test de la condition** :
 - Avec `while`, la condition est vérifiée **avant** l'exécution.
 - Avec `do...while`, la condition est vérifiée **après** l'exécution, ce qui garantit un passage obligatoire dans la boucle.
- **Cas d'utilisation** :
 - Utilise `while` lorsque tu n'es pas sûr qu'un passage dans la boucle soit nécessaire dès le départ.
 - Utilise `do...while` quand tu veux forcer l'exécution du bloc au moins une fois (par exemple, pour afficher un menu ou demander une saisie utilisateur).
- **Évite les boucles infinies** : Assure-toi que la condition évolue vers un état faux, par exemple en incrémentant une variable à chaque itération.

Ce qu'il faut retenir

1. **Initialisation** : Toujours initialiser la variable qui contrôle la boucle.
2. **Condition** : La condition de la boucle doit être claire et atteignable.
3. **Incrémentation/Décrémentation** : Modifie la variable de contrôle pour sortir de la boucle.
4. **Différence essentielle** : La boucle `while` teste la condition avant, tandis que `do...while` le fait après l'exécution.
5. **Utilisation adaptée** : Choisis la boucle qui correspond à la logique de ton programme.