PHP : Les boucles for

Les boucles `for` sont des outils essentiels en programmation qui te permettent d'exécuter un bloc de code un nombre déterminé de fois. Elles sont particulièrement utiles lorsque tu connais à l'avance le nombre d'itérations nécessaires. Dans cet article, nous allons voir comment utiliser la boucle `for` en PHP et explorer **5 exemples concrets de la vie courante** pour mieux comprendre leur utilité.

Objectifs de la leçon

- Comprendre la structure et le fonctionnement de la boucle `for` en PHP.
- Apprendre à appliquer cette boucle à des situations concrètes.
- S'exercer à écrire et modifier des boucles pour résoudre des problèmes pratiques.
- Identifier les bonnes pratiques pour éviter les erreurs (par exemple, les boucles infinies).

Structure de la Boucle `for`

La structure de la boucle `for` en PHP se compose de trois parties placées dans les parenthèses :

- **Initialisation** : Permet de définir la variable de contrôle.
- **Condition** : Tant que cette condition est vraie, le bloc de code s'exécute.
- **Incrémentation/Décrémentation** : Modifie la variable de contrôle à chaque itération.

Syntaxe générale :

```
for (initialisation; condition; incrémentation) {
    // Code à exécuter
}
```

Exemples Concrets

Exemple 1 : Afficher une Liste de Numéros

Imaginons que tu veuilles afficher les numéros de 1 à 10, par exemple pour numéroter les pages d'un rapport.

```
<?php
for ($i = 1; $i <= 10; $i++) {
    echo "Page numéro : $i<br>";
}
?>
```

Explication :
La boucle commence à 1 et s'arrête quand `$i` atteint 10. À chaque itération, le numéro de page est affiché.

Exemple 2 : Calculer le Total d'une Commande

Supposons que tu as une liste de prix de produits et que tu souhaites calculer le total de la commande.

```php
<?php
$prixProduits = [15.99, 23.50, 9.75, 12.30];
$total = 0;

for ($i = 0; $i < count($prixProduits); $i++) {
    $total += $prixProduits[$i];
}

echo "Le total de la commande est : $total €";
?>
```

Explication :
La boucle parcourt le tableau des prix et ajoute chaque valeur à la variable `$total`. À la fin, le montant total est affiché.

Exemple 3 : Afficher les Jours d'une Semaine

Si tu veux afficher les jours de la semaine dans un planning, tu peux utiliser une boucle `for` associée à un tableau.

```php
<?php
$jours = ["Lundi", "Mardi", "Mercredi", "Jeudi", "Vendredi", "Samedi", "Dimanche"];

for ($i = 0; $i < count($jours); $i++) {
    echo "Jour " . ($i + 1) . " : " . $jours[$i] . "<br>";
}
?>
```

Explication :
La boucle itère sur chaque index du tableau `$jours`, et affiche le jour correspondant avec son numéro.

Exemple 4 : Simuler un Compte à Rebours pour un Événement

Pour créer un compte à rebours qui annonce un événement, tu peux utiliser une boucle `for` qui décrémente une valeur.

```php
<?php
$tempsRestant = 10; // Par exemple, 10 secondes avant l'événement

for ($i = $tempsRestant; $i > 0; $i--) {
    echo "L'événement démarre dans : $i secondes<br>";
}

echo "L'événement commence maintenant !";
?>
```

Explication :
La boucle commence à 10 et décrémente `$i` à chaque itération jusqu'à ce qu'il atteigne 1, créant ainsi un compte à rebours.

Exemple 5 : Générer une Table de Multiplication

Une autre application pratique consiste à générer la table de multiplication d'un nombre, par exemple celle du 5.

```php
<?php
$nombre = 5;

echo "Table de multiplication de $nombre :<br>";

for ($i = 1; $i <= 10; $i++) {
    $resultat = $nombre * $i;
    echo "$nombre x $i = $resultat<br>";
}
?>
```

Explication :
La boucle parcourt les nombres de 1 à 10 et calcule le résultat de la multiplication du nombre donné par l'itération. Cela permet d'afficher la table de multiplication de manière dynamique.

Bonnes Pratiques et Conseils

1. **Initialisation Correcte :** Assure-toi de bien initialiser la variable de contrôle pour éviter des erreurs ou des boucles infinies.
2. **Condition Clair :** La condition doit être formulée de manière à ce qu'elle devienne fausse à un moment donné.
3. **Incrémentation/Décrémentation :** Ne pas oublier de modifier la variable de contrôle afin de garantir que la boucle se termine.
4. **Utiliser `count()` pour les Tableaux :** Lors du parcours d'un tableau, utilise la fonction `count()` pour rendre ton code adaptable même si le tableau change de taille.
5. **Commentaires :** N'hésite pas à commenter ton code pour expliquer le rôle de chaque partie de la boucle, surtout dans un contexte pédagogique.

Ce qu'il faut retenir

- La boucle `for` est idéale lorsque tu connais le nombre exact d'itérations.
- Elle se compose de trois parties : initialisation, condition et incrémentation/décrémentation.
- Les exemples concrets montrent comment appliquer la boucle à divers cas pratiques, allant de l'affichage de numéros à la génération d'une table de multiplication.
- Les bonnes pratiques garantissent un code clair et évitent les erreurs comme les boucles infinies.
