
Les structures de contrôle en PHP



3 types de structures de contrôle

- **Conditions** (if, else, elseif, ...)
- **Boucles** (for, foreach)
- **Instructions d'arrêt** (Break, Continue)

Structures de Contrôle

Conditions : if

- Permet l'exécution conditionnelle d'une partie de code. (SI)
- IF

Exemple :

```
$age = 34;  
  
if ($age >= 18) {  
    echo "Tu es majeur";  
}
```

Structures de Contrôle

Conditions : else

- Permet l'exécution conditionnelle d'une partie de code. (ALORS)
- ELSE (toujours associé à un IF)

Exemple :

```
$a = 10;  
  
if ($a > 5) {  
    echo "La variable a est supérieure à 5.";  
} else {  
    echo "La variable a n'est pas supérieure à 5.";  
}
```

Structures de Contrôle

Conditions : elseif

- Permet l'exécution d'une série d'instructions if sans avoir besoin des les imbriquer

- ELSEIF (toujours associé à un IF)

Exemple :

```
$note = 75;

if ($note >= 90) {
    echo "A";
} elseif ($note >= 80) {
    echo "B";
} elseif ($note >= 70) {
    echo "C";
} elseif ($note >= 60) {
    echo "D";
} else {
    echo "E";
}
```



Structures de Contrôle

Conditions : switch

Permet de tester plusieurs valeurs possibles d'une expression et d'exécuter le code correspondant.

```
$jour = "lundi";

switch ($jour) {
    case "lundi":
        echo "C'est le début de la semaine.";
        break;
    case "mardi":
        echo "On est mardi.";
        break;
    case "mercredi":
        echo "Mercredi, bientôt le milieu de la semaine.";
        break;
    case "jeudi":
        echo "Jeudi, presque le week-end.";
        break;
    case "vendredi":
        echo "Enfin vendredi, week-end en vue !";
        break;
    case "samedi":
        echo "C'est samedi";
        break;
    case "dimanche":
        echo "C'est le week-end !";
        break;
    default:
        echo "Ce n'est pas un jour de la semaine valide.";
}
```

Structures de Contrôle

Boucle : while

- Exécute une instruction tant qu'une condition est vraie.

Exemple :

```
$count = 0;
```

```
while ($count < 3) {  
    echo "Le compteur est à $count. ";  
    $count++;  
}
```

// Affiche : Le compteur est à 0. Le compteur est à 1. Le compteur est à 2.

// À chaque itération, la valeur de \$compteur est augmentée de 1

Structures de Contrôle

Boucle : for

- Utilisée pour exécuter un bloc de code un nombre spécifié de fois.

Exemple :

```
for ($i = 1; $i <= 5; $i++) {  
    echo $i . " ";  
}  
// Affiche : 1 2 3 4 5
```


Structures de Contrôle

Boucle : foreach (sans clé)

Tableau à 1 dimension

- Permet de parcourir des tableaux (BOUCLE)
- Permet de répéter des instructions plusieurs fois
- FOREACH

```
// Définition d'un tableau à 1 dimension
$liste = ['banane', 'orange', 'kiwi'];

// Utilisation de la boucle foreach pour parcourir le tableau
foreach ($liste as $fruit) {
    echo $fruit . '<br>';
}

/*
Résultat :
banane
orange
kiwi
*/
```

Structures de Contrôle

Boucle : foreach (sans clé)

Tableau à 2 dimensions

- Permet de parcourir des tableaux (BOUCLE)
- Permet de répéter des instructions plusieurs fois
- FOREACH

// Définition d'un tableau associatif (tableau à 2 dimensions)

```
$personne = array(  
    "nom" => "Doe",  
    "prenom" => "John",  
    "age" => 30,  
    "ville" => "New York"  
);
```

// Utilisation de la boucle foreach pour parcourir le tableau

```
foreach ($personne as $valeur) {  
    echo $valeur . '<br>';  
}
```

```
/*  
Résultat :  
nom : Doe  
prenom : John  
age : 30  
ville : New York  
*/
```

Structures de Contrôle

Boucle : foreach (avec clé)

Tableau à 2 dimensions

- Permet de parcourir des tableaux (BOUCLE)
- Permet de répéter des instructions plusieurs fois
- FOREACH

// Définition d'un tableau associatif (tableau à 2 dimensions)

```
$personne = array(  
    "nom" => "Doe",  
    "prenom" => "John",  
    "age" => 30,  
    "ville" => "New York"  
);
```

// Utilisation de la boucle foreach pour parcourir le tableau

```
foreach ($personne as $cle => $valeur) {  
    echo "$cle : $valeur <br>";  
}
```

```
/*  
Résultat :  
nom : Doe  
prenom : John  
age : 30  
ville : New York  
*/
```

Structures de Contrôle

Instruction d'arrêt : break

- Permet de sortir d'une structure conditionnelle telle que for, while, foreach ou switch
- Remarque : break accepte un argument numérique optionnel qui doit indiquer combien de structures emboîtées doivent être interrompues

```
for ($i = 1; $i <= 10; $i++) {  
    if ($i > 3) {  
        break;  
    }  
    echo "Itération $i\n";  
}
```

Dès que `$i` devient supérieur à 3, l'instruction `break` est exécutée, ce qui entraîne la sortie immédiate de la boucle.

Ainsi, le message `Itération $i` ne sera affiché que pour les valeurs 1, 2 et 3.

Structures de Contrôle

Instruction d'arrêt : continue

- sert dans une boucle à d'éluder les instructions de l'itération courante et de continuer l'exécution à la condition de l'évaluation et donc de commencer la prochaine itération.

```
for ($i = 1; $i <= 5; $i++) {  
    if ($i % 2 == 0) {  
        continue; // saute l'itération si $i est un  
nombre pair  
    }  
    echo "Nombre impair: $i\n";  
}
```

La boucle **for** itère de 1 à 5.

À chaque itération, il y a une vérification pour voir si le nombre courant **\$i** est pair (divisible par 2).

Si c'est le cas, l'instruction **continue** est exécutée, et le reste du code dans cette itération de la boucle (l'instruction **echo**) est ignoré.

Le script continue donc à la prochaine itération de la boucle. En conséquence, seuls les nombres impairs (1, 3, 5) seront affichés.