

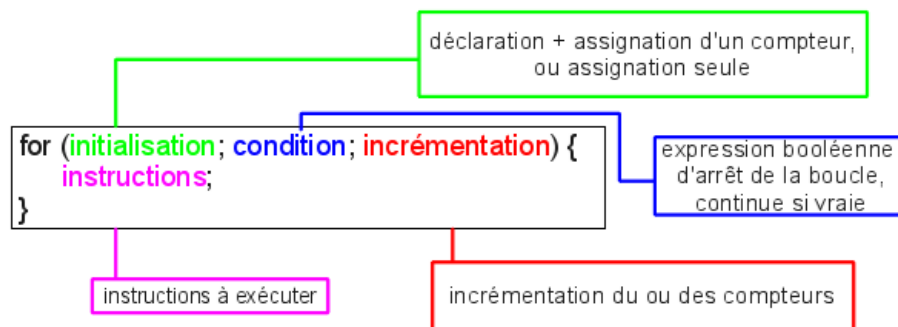


Introduction au langage de programmation PHP

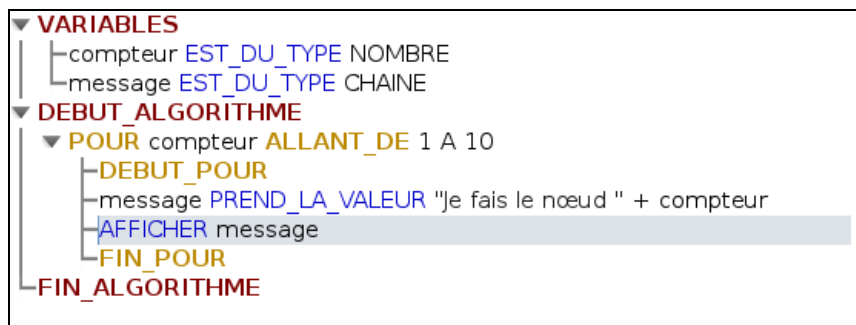
[Accueil](#) ► [Mes cours](#) ► [Développement logiciel](#) ► [Intro PHP](#) ► [Les structures de contrôle](#) ► [La boucle for](#)

La boucle for

La boucle `for` est équivalente à la structure POUR d'AlgoBox : elle permet la répétition d'instructions en utilisant un compteur. PHP va donner un peu de souplesse à cette structure et nous permettre, dans une boucle `for`, d'utiliser plusieurs compteurs ainsi que d'incrémenter ou de décrémenter à notre guise ces compteurs. Les déclarations des compteurs pourront se faire directement dans la boucle, ou à l'extérieur comme nous l'avons fait avec AlgoBox. Voici la syntaxe de la boucle `for` :



Nous allons implémenter en PHP cet algorithme :



Ce qui nous donne :

```

1 <?php
2
3 // for_1.php
4 for ($i = 1; $i <= 10; $i++) {
5     echo "je fais le noeud $i" . PHP_EOL;
6 }
  
```

Ici le compteur est incrémenté de 1 à chaque tour de boucle : vous avez noté l'utilisation de l'opérateur d'incrémentation (`++`). Nous pourrions tout aussi bien incrémenter le compteur d'une autre valeur, et même le décrémenter. Par exemple si je souhaite afficher les valeurs de 10 à 0 inclus en faisant "reculer" le compteur de 2 pas à chaque tour de boucle :

```

1 <?php
2
3 // for_2.php
4 for ($i = 10; $i >= 0; $i -=2) {
5     echo $i . PHP_EOL;
6 }
  
```

Cette fois nous avons utiliser l'opérateur d'assignation `-=`. À chaque tour de boucle `$i` prend la valeur `$i - 2`.

Nous pourrions aussi utiliser plusieurs compteurs, et incrémenter/décrémenter ces compteurs. Si je souhaite afficher les 10 premières additions dont la somme est 10, je peux écrire :

```
1 <?php
2
3 // for_3.php
4 for ($i = 0, $j = 10; $i < 10; $i++, $j--) {
5     $somme = $i + $j;
6     echo "$i + $j = $somme" . PHP_EOL;
7 }
```

Notez que dans l'initialisation et l'incrémentation, les instructions sont séparées par des virgules.

Dans une boucle `for`, l'initialisation, la condition ainsi que l'incrémentation sont facultatives. Seuls les points-virgules sont obligatoires. Je pourrais réécrire l'exemple précédent en déclarant mes compteurs à l'extérieur de la boucle, et incrémenter et décrémenter mes compteurs dans le "corps" de la boucle. Il ne me resterait plus entre parenthèses que la condition et les points-virgules :

```
1 <?php
2
3 // for_4.php
4 $i = 0;
5 $j = 10;
6 for (; $i < 10;) {
7     $somme = $i + $j;
8     echo "$i + $j = $somme" . PHP_EOL;
9     $i++;
10    $j--;
11 }
```

Cela ne ressemble plus vraiment à une boucle `for`...

Fin

NAVIGATION



Accueil

■ Ma page

Pages du site

Mon profil

Cours actuel

Intro PHP

Participants

Le langage PHP : introduction

Les types et les variables

Les opérateurs

Les structures de contrôle

Boucles et conditions

La condition if

L'opérateur ternaire

T.P. conditions

La structure switch

La boucle for

T.P. boucles for

La boucle while

La boucle do-while

T.P. boucles while et do-while

Break et continue

T.P. imbrications

Les structure de données

Les fonctions
Les erreurs
Les fichiers
Les expressions rationnelles
PHP et HTML
Petite application

[Mes cours](#)

ADMINISTRATION



Administration du cours

Réglages de mon profil

Connecté sous le nom « [Arnaud Lemais](#) » ([Déconnexion](#))
[Intro PHP](#)