

## Les structures itératives

Algo	C et dérivé	Basic
<p>On connaît à l'avance le nombre d'itérations</p> <p><b>Pour</b> <math>i \leftarrow 0</math> jusqu'à 9</p> <p><b>Fin Pour</b></p> <p>En général les tableaux commencent à 0. c'est équivalent à:</p> <p><b>Pour</b> <math>i \leftarrow 1</math> jusqu'à 10</p> <p><b>Fin Pour</b></p> <p>Nous considérerons <math>i</math> comme une variable de type Entier. Cette variable sert de compteur.</p> <p>Pour l'exemple, nous allons montrer comment boucler 10 fois. Évidemment vous devez adapter votre code à votre besoin.</p>	<pre>int i; for (i=0; i&lt;10; i++){   bloc instructions; }</pre> <p>ou en C++ et dans tout les autres dérivés.</p> <pre>for (int i=0; i&lt;10; i++){   bloc instructions; }</pre> <p>La portée de la variable <math>i</math> n'est que dans la boucle.  C'est à dire que les autres partie du programme ne peuvent utiliser <math>i</math> (le compteur)</p> <p>On peut décrémenter le compteur en utilisant <math>i--</math></p>	<pre>For I=1 To 10   blocInstructions Next</pre> <pre>For I=10 To 1 Step -1   blocInstructions Next</pre> <p>Pour changer le pas dans une boucle For on utilise le mot clé optionnel <b>step</b></p>
<p>On ne connaît pas le nombre d'itérations. On n'est pas sur de devoir entrer dans la boucle.</p> <p><b>Tant Que</b> (Conditions_entrée)</p> <p><b>Fin Tant Que</b></p>	<pre>while (condition_entree){   bloc Instructions; }</pre>	<pre>while condition_entree   bloc Instructions wend</pre>
<p>On ne connaît pas le nombre d'itérations. On doit entrer au moins une fois dans la boucle.</p> <p><b>Rejeter</b></p> <p><b>Jusqu'à</b> (Condition_sortie)</p>	<pre>do {   bloc Instructions; } while (Condition_entree);</pre> <p>ou</p> <pre>do {   bloc Instructions; } while (Non Condition_sortie)</pre> <p>Le répéter Jusqu'à n'existe pas en C. Ce serait plutôt un:  Faire Tant que la condition est vérifié</p>	<pre>Do   bloc Instructions Loop Until Condition_sortie</pre>