



FICHE RESSOURCE PROGRAMMATION

Cette fiche présente les structures dans les langages Python et Arduino.

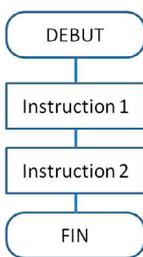
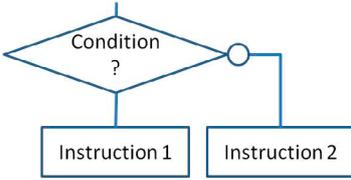
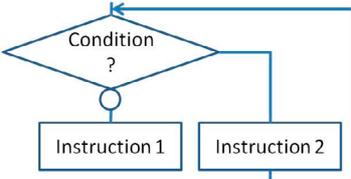
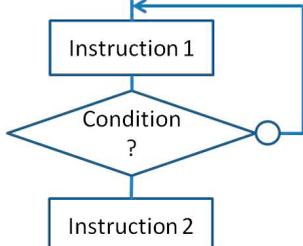
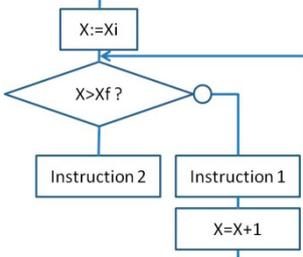
Ressource

Langage Python : voir sur le web

[Langage Arduino](#)

La fiche ne présente pas tous les cas. À vous de chercher sur le web pour les détails sur les structures.

Les structures le plus courantes sont présentées ci-dessous.

Algorithme	Langage Python	Langage Arduino
<p>Séquence linéaire :</p> 	<p>DÉBUT</p> <p>Instruction 1 Instruction 2</p> <p>FIN</p>	<pre> { Instruction 1; Instruction 2; } </pre>
<p>Séquence alternative :</p>  <p>Condition : expression booléenne</p>	<p>SI ALORS SINON (IF-THEN-ELSE-ENDIF)</p> <p>SI <i>expression booléenne</i> ALORS</p> <p>Instruction1 ...</p> <p>SINON</p> <p>Instruction 2 ...</p> <p>FIN_SI</p>	<pre> if (<i>expression booléenne</i>) { Instruction 1; } else { Instruction 2; } Si plusieurs tests : If-elif-else </pre>
<p>Séquence répétitive :</p> 	<p>TANT_QUE (WHILE-ENDWHILE)</p> <p>TANT_QUE <i>expression booléenne</i></p> <p>Instruction 2 ...</p> <p>FIN_TANT_QUE</p>	<pre> while (<i>expression booléenne vraie</i>) { Instruction2; Instruction 1; } </pre>
<p>Séquence répétitive :</p> 	<p>RÉPÉTÉ JUSQU'A (REPAT-UNTIL)</p> <p>RÉPÉTER</p> <p>Instruction1</p> <p>JUSQU'A <i>expression booléenne</i></p> <p>Instruction 2</p>	<p>N'existe pas directement en python.</p> <pre> while True : Instruction 1 If <i>expression booléenne</i> : Break Instruction 2 </pre>
 <p>X : variable; Xi : valeur début et Xf : valeur finale</p>	<p>POUR (FOR - TO - ENDFOR)</p> <p>POUR <i>variable ALLANT DE valeur début A valeur fin (PAR_PAS_DE incrément)</i></p> <p>Instruction1</p> <p>FIN_POUR</p> <p>Instruction 2</p>	<pre> for X in range (Xi, Xf) : Instruction1 Instruction 2 for (Xi; X>Xf; X++) { Instruction1; } { Instruction 2; } Remarque : X++ : incrément de 1 </pre>

Retrouvez éducol sur

