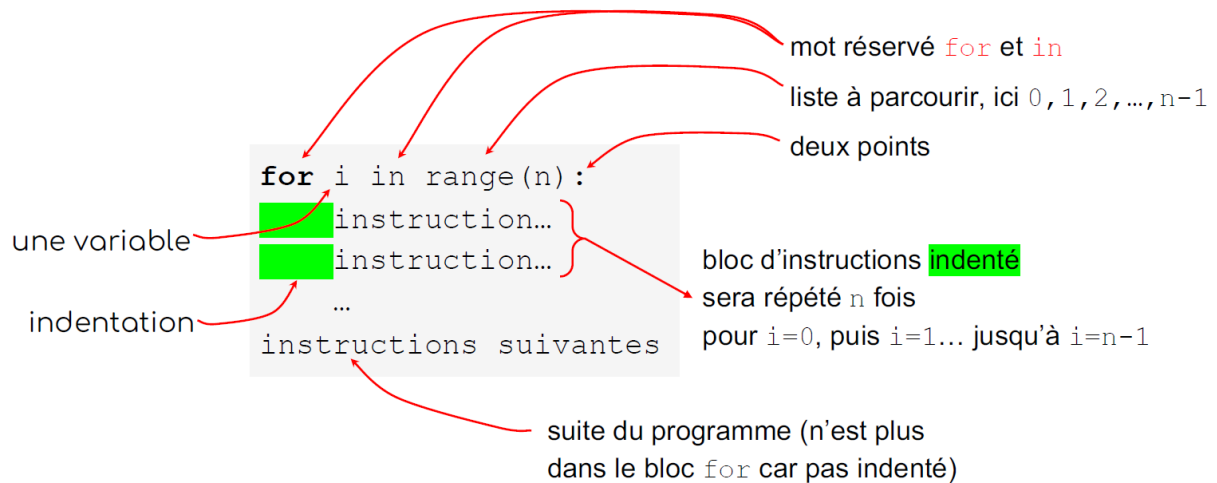


Vidéo boucle for

Les boucles **for** sont généralement utilisées quand le nombre d'itérations de la boucle **est connu** à l'avance. L'intérêt d'une boucle est d'éviter les répétitions de certaines instructions.

Boucle pour (**for**)



for **i** **in** **range**(...):
 instruction 1...
 instruction 2...

range (fin)	# [0 ; fin [⇒	la variable <i>i</i> va de 0 à fin-1
range (debut , fin)	# [debut ; fin [⇒	la variable <i>i</i> va de debut à fin exclu !
range (debut , fin, pas)	# comme précédemment avec un pas.	

Exemple :		
for i in range (5): print (i)	for i in range (1, 5): print (i)	for i in range (1, 5, 2): print (i)
la variable <i>i</i> va de 0 à 4 Les valeurs affichées: 0 1 2 3 4	la variable <i>i</i> va de 1 à 4 Les valeurs affichées: 1 2 3 4	la variable <i>i</i> va de 1 à 4 avec un pas de 2 Les valeurs affichées: 1 3

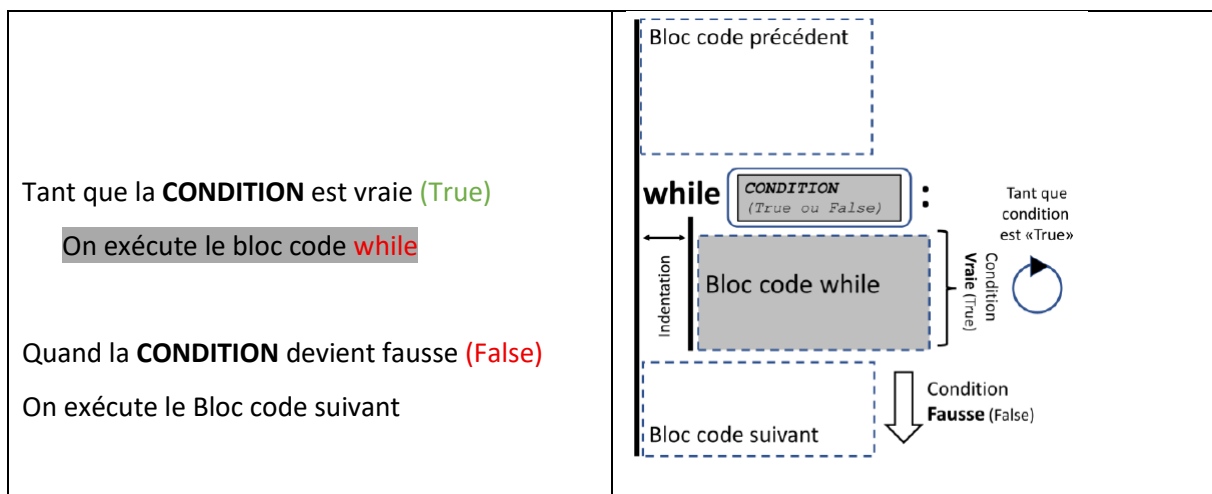
Exemple : La boucle affiche les caractères du mot "Hello"	
mot="Hello" for c in mot: print (c)	H E l l o

Exemple: répéter 5 fois la ligne qui contient 3 fois le mot bonjour	
La variable i va de 0 à 4	
<pre> mot = "bonjour" for i in range(5): print((mot + " ") * 3) </pre>	<pre> bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour bonjour </pre>

Exemple: répéter n fois (ici n=5) une instruction	
<pre> for i in range(5): print(i, "* 2 =", i*2) </pre>	<pre> 0 * 2 = 0 1 * 2 = 2 2 * 2 = 4 3 * 2 = 6 4 * 2 = 8 </pre>

[Vidéo boucle while](#)

Les boucles **while** sont utilisées quand le nombre d'itérations n'est pas connu à l'avance. Une boucle **while** est une boucle qui continue son exécution **tant que** le test logique (condition) est vrai.



Exemple :

```

x = 1
while x < 10:
    print("Valeur de x:", x)
    x = x * 2
print("Fin")

```

```

Valeur de x: 1
Valeur de x: 2
Valeur de x: 4
Valeur de x: 8
Fin

```

Instruction **break**

L'instruction **break** fait sortir d'une boucle **for** ou d'une boucle **while** (elle "casse" la boucle).

<pre>a = 0 while a < 5: print(a) if a == 2: break a = a+1 print("Fin")</pre>	<pre>a = 0 0 < 5 ? affiche 0 0 == 2 ? a = 0 + 1</pre>	<pre>1 < 5 ? affiche 1 1 == 2 ? a = 1 + 1</pre>	<pre>2 < 5 ? affiche 2 2 == 2 ? break Fin</pre>	<pre>0 1 2 Fin</pre>
---	---	---	---	--