



# Programmation 2

Slides modified from <https://github.com/UGE-IGM/courspython>

# Chapitre 3 : Instructions Répétitives

Dans ce chapitre vous allez apprendre à :

- répéter une tâche un nombre fini de fois ou tant qu'une condition reste vraie ;
- expliquer les différents types de boucle en python ;
- reconnaître la nécessité de créer une ou plusieurs itérations dans un programme ;
- choisir le type de boucle le plus adapté à la situation ;
- décrire l'exécution d'un programme simple contenant une ou plusieurs boucle(s) ;
- être capable d'écrire un code comprenant une ou des itérations.

# I Introduction

En programmation, on est souvent amené à répéter plusieurs fois une instruction. Incontournables à tout langage de programmation, les **boucles** vont nous aider à réaliser cette tâche de manière compacte et efficace.

Imaginez par exemple que vous souhaitiez afficher :

```
1 kilomètre(s) à pied ça use, ça use ...  
2 kilomètre(s) à pied ça use, ça use ...  
3 kilomètre(s) à pied ça use, ça use ...  
etc.
```

Dans l'état actuel de vos connaissances, il faudrait taper quelque chose du style :

In [1]:

```
print(1, 'kilomètre(s) à pied ça use, ça use ...')  
print(2, 'kilomètre(s) à pied ça use, ça use ...')  
print(3, 'kilomètre(s) à pied ça use, ça use ...')  
print(4, 'kilomètre(s) à pied ça use, ça use ...')  
print(5, 'kilomètre(s) à pied ça use, ça use ...')  
print(6, 'kilomètre(s) à pied ça use, ça use ...')
```

```
1 kilomètre(s) à pied ça use, ça use ...  
2 kilomètre(s) à pied ça use, ça use ...  
3 kilomètre(s) à pied ça use, ça use ...  
4 kilomètre(s) à pied ça use, ça use ...  
5 kilomètre(s) à pied ça use, ça use ...  
6 kilomètre(s) à pied ça use, ça use ...
```

Mais si on voulait compter jusqu'à 100 ? 1000 ? et plus encore... ?  
Nous allons voir comment réaliser simplement cette tâche grâce aux boucles. Python propose pour cela deux instructions particulières pour construire des boucles :

- l'instruction **for ... in ...**, très puissante, qui sera abordée dans ce chapitre et beaucoup plus en détail au chapitre concernant les itérables ;
- l'instruction **while** abordée en deuxième partie du chapitre.

II Répétition d'une tâche un certain nombre de fois : `for ...`  
`in range(...)`:

La syntaxe `for ... in range(...)` est faite pour répéter une tâche un nombre de fois, fini et connu à l'avance.

### Syntaxe :

```
# début
N = ...
for <var> in range(N):
    # bloc d'instructions
    # (corps de la boucle)
# suite
```

où :

- N est une variable de type `int` , qui doit être initialisée avant le début de la boucle ;
- `var` est le nom d'une variable que vous choisirez : elle est initialisée toute seule à `0` ;

- `range(N)` est une fonction primitive du langage Python permettant de simuler l'intervalle du type `[0;N - 1]` : cela signifie que `var` prendra successivement toutes les valeurs entières de 0 à `N - 1` ;

### Vocabulaire :

- Le bloc d'instructions est appelé **corps de la boucle**.
- Chaque exécution du corps de la boucle est appelée **itération**.
- La variable nommée `var` est appelée **variable d'itération**.

**Remarque :** À chaque nouvelle répétition du corps de la boucle, la variable `var` est incrémentée (c'est-à-dire que sa valeur est augmentée de 1 ). Cela permet de répéter une tâche légèrement modifiée une fois sur l'autre (voir exemple 2).



Il n'est pas possible de modifier `var` dans le corps de la boucle.

Dans la suite, nous allons voir deux exemples d'utilisation de la syntaxe `for ... in range(...)` correspondant à deux cas d'usage :

- le corps de la boucle n'utilise pas la variable d'itération.
- le corps de la boucle utilise la variable d'itération.

Exemple 1 : Boucle dont le corps est indépendant de la variable d'itérations

Si le corps de la boucle ne dépend pas de la variable d'itération, il s'agit d'une répétition stricte du corps de la boucle !  
Par exemple, le code suivant imprimera 5 fois Bonjour.

In [2]:

```
N = 5
for nb in range(0, N):
    print("Bonjour les AP1 !")
    print(nb)
```

Bonjour les AP1 !

0

Bonjour les AP1 !

1

Bonjour les AP1 !

2

Bonjour les AP1 !

3

Bonjour les AP1 !

4

## EXERCICE 1 :

Dans le programme suivant, quelle est la variable d'itération ?  
Qu'elle est sa valeur à la fin de la boucle ?

```
fin = 10  
for cmpt in range(fin):  
    print("Vive Python 3 !")
```

## Exemple 2 : Boucle dont le corps dépend de la variable d'itérations

Si le corps de la boucle dépend de la variable d'itération, la répétition porte sur le nombre de fois où le corps est exécuté. Il faut donc bien réfléchir pour identifier ce qui est répété et comment.

Voici tout d'abord comment la variable d'itération évolue au cours de la boucle :

In [7]:

```
fin = 10
for nb in range(fin):
    print("nb = ", nb)
print("A la fin, nb = ", nb)
```

```
nb = 0
nb = 1
nb = 2
nb = 3
nb = 4
nb = 5
nb = 6
nb = 7
nb = 8
nb = 9
A la fin, nb = 9
```

On peut alors facilement écrire un programme qui affiche :

```
1 kilomètre(s) à pied ça use, ça use ...  
2 kilomètre(s) à pied ça use, ça use ...  
3 kilomètre(s) à pied ça use, ça use ...  
etc.
```

Pour cela, on a besoin de savoir identifier où se trouve la répétition : il s'agit ici de l'affichage d'une ligne ... kilomètre(s) à pied ça use, ça use ....

Plus précisément, la  $k$ -ième ligne commencera toujours par  $k$  kilomètre(s) ....

Nous pouvons donc écrire le programme suivant :

In [1]:

```
N = 6
for nb in range(N):
    print(nb + 1, 'kilomètre(s) à pied ça use, ça use ...')
```

```
1 kilomètre(s) à pied ça use, ça use ...
2 kilomètre(s) à pied ça use, ça use ...
3 kilomètre(s) à pied ça use, ça use ...
4 kilomètre(s) à pied ça use, ça use ...
5 kilomètre(s) à pied ça use, ça use ...
6 kilomètre(s) à pied ça use, ça use ...
```

In [12]:

```
N = 6
for nb in range(N):
    print(f'{nb + 1} kilomètre(s) à pied ça use, ça use ...')
```

```
1 kilomètre(s) à pied ça use, ça use ...
2 kilomètre(s) à pied ça use, ça use ...
3 kilomètre(s) à pied ça use, ça use ...
4 kilomètre(s) à pied ça use, ça use ...
5 kilomètre(s) à pied ça use, ça use ...
6 kilomètre(s) à pied ça use, ça use ...
```

On peut même améliorer le programme en gérant le pluriel avec une conditionnelle dans la boucle :

In [1]:

```
N = 600
for nb in range(N):
    if nb == 0:
        print(f'{nb + 1} kilomètre à pied ça use, ça use ...')
    else:
        print(f'{nb + 1} kilomètres à pied ça use, ça use ...')
```

1 kilomètre à pied ça use, ça use ...  
2 kilomètres à pied ça use, ça use ...  
3 kilomètres à pied ça use, ça use ...  
4 kilomètres à pied ça use, ça use ...  
5 kilomètres à pied ça use, ça use ...  
6 kilomètres à pied ça use, ça use ...  
7 kilomètres à pied ça use, ça use ...  
8 kilomètres à pied ça use, ça use ...  
9 kilomètres à pied ça use, ça use ...  
10 kilomètres à pied ça use, ça use ...  
11 kilomètres à pied ça use, ça use ...  
12 kilomètres à pied ça use, ça use ...  
13 kilomètres à pied ça use, ça use ...  
14 kilomètres à pied ça use, ça use ...  
15 kilomètres à pied ça use, ça use ...  
16 kilomètres à pied ça use, ça use ...  
17 kilomètres à pied ça use, ça use ...  
18 kilomètres à pied ça use, ça use ...  
19 kilomètres à pied ça use, ça use ...  
20 kilomètres à pied ça use, ça use ...  
21 kilomètres à pied ça use, ça use ...  
22 kilomètres à pied ça use, ça use ...  
23 kilomètres à pied ça use, ça use ...  
24 kilomètres à pied ça use, ça use ...  
25 kilomètres à pied ça use, ça use ...  
26 kilomètres à pied ça use, ça use ...  
27 kilomètres à pied ça use, ça use ...  
28 kilomètres à pied ça use, ça use ...  
29 kilomètres à pied ça use, ça use ...  
30 kilomètres à pied ça use, ça use ...  
31 kilomètres à pied ça use, ça use ...

32 kilomètres à pied ça use, ça use ...  
33 kilomètres à pied ça use, ça use ...  
34 kilomètres à pied ça use, ça use ...  
35 kilomètres à pied ça use, ça use ...  
36 kilomètres à pied ça use, ça use ...  
37 kilomètres à pied ça use, ça use ...  
38 kilomètres à pied ça use, ça use ...  
39 kilomètres à pied ça use, ça use ...  
40 kilomètres à pied ça use, ça use ...  
41 kilomètres à pied ça use, ça use ...  
42 kilomètres à pied ça use, ça use ...  
43 kilomètres à pied ça use, ça use ...  
44 kilomètres à pied ça use, ça use ...  
45 kilomètres à pied ça use, ça use ...  
46 kilomètres à pied ça use, ça use ...  
47 kilomètres à pied ça use, ça use ...  
48 kilomètres à pied ça use, ça use ...  
49 kilomètres à pied ça use, ça use ...  
50 kilomètres à pied ça use, ça use ...  
51 kilomètres à pied ça use, ça use ...  
52 kilomètres à pied ça use, ça use ...  
53 kilomètres à pied ça use, ça use ...  
54 kilomètres à pied ça use, ça use ...  
55 kilomètres à pied ça use, ça use ...  
56 kilomètres à pied ça use, ça use ...  
57 kilomètres à pied ça use, ça use ...  
58 kilomètres à pied ça use, ça use ...  
59 kilomètres à pied ça use, ça use ...  
60 kilomètres à pied ça use, ça use ...  
61 kilomètres à pied ça use, ça use ...  
62 kilomètres à pied ça use, ça use ...

63 kilomètres à pied ça use, ça use ...  
64 kilomètres à pied ça use, ça use ...  
65 kilomètres à pied ça use, ça use ...  
66 kilomètres à pied ça use, ça use ...  
67 kilomètres à pied ça use, ça use ...  
68 kilomètres à pied ça use, ça use ...  
69 kilomètres à pied ça use, ça use ...  
70 kilomètres à pied ça use, ça use ...  
71 kilomètres à pied ça use, ça use ...  
72 kilomètres à pied ça use, ça use ...  
73 kilomètres à pied ça use, ça use ...  
74 kilomètres à pied ça use, ça use ...  
75 kilomètres à pied ça use, ça use ...  
76 kilomètres à pied ça use, ça use ...  
77 kilomètres à pied ça use, ça use ...  
78 kilomètres à pied ça use, ça use ...  
79 kilomètres à pied ça use, ça use ...  
80 kilomètres à pied ça use, ça use ...  
81 kilomètres à pied ça use, ça use ...  
82 kilomètres à pied ça use, ça use ...  
83 kilomètres à pied ça use, ça use ...  
84 kilomètres à pied ça use, ça use ...  
85 kilomètres à pied ça use, ça use ...  
86 kilomètres à pied ça use, ça use ...  
87 kilomètres à pied ça use, ça use ...  
88 kilomètres à pied ça use, ça use ...  
89 kilomètres à pied ça use, ça use ...  
90 kilomètres à pied ça use, ça use ...  
91 kilomètres à pied ça use, ça use ...  
92 kilomètres à pied ça use, ça use ...  
93 kilomètres à pied ça use, ça use ...

94 kilomètres à pied ça use, ça use ...  
95 kilomètres à pied ça use, ça use ...  
96 kilomètres à pied ça use, ça use ...  
97 kilomètres à pied ça use, ça use ...  
98 kilomètres à pied ça use, ça use ...  
99 kilomètres à pied ça use, ça use ...  
100 kilomètres à pied ça use, ça use ...  
101 kilomètres à pied ça use, ça use ...  
102 kilomètres à pied ça use, ça use ...  
103 kilomètres à pied ça use, ça use ...  
104 kilomètres à pied ça use, ça use ...  
105 kilomètres à pied ça use, ça use ...  
106 kilomètres à pied ça use, ça use ...  
107 kilomètres à pied ça use, ça use ...  
108 kilomètres à pied ça use, ça use ...  
109 kilomètres à pied ça use, ça use ...  
110 kilomètres à pied ça use, ça use ...  
111 kilomètres à pied ça use, ça use ...  
112 kilomètres à pied ça use, ça use ...  
113 kilomètres à pied ça use, ça use ...  
114 kilomètres à pied ça use, ça use ...  
115 kilomètres à pied ça use, ça use ...  
116 kilomètres à pied ça use, ça use ...  
117 kilomètres à pied ça use, ça use ...  
118 kilomètres à pied ça use, ça use ...  
119 kilomètres à pied ça use, ça use ...  
120 kilomètres à pied ça use, ça use ...  
121 kilomètres à pied ça use, ça use ...  
122 kilomètres à pied ça use, ça use ...  
123 kilomètres à pied ça use, ça use ...  
124 kilomètres à pied ça use, ça use ...

125 kilomètres à pied ça use, ça use ...  
126 kilomètres à pied ça use, ça use ...  
127 kilomètres à pied ça use, ça use ...  
128 kilomètres à pied ça use, ça use ...  
129 kilomètres à pied ça use, ça use ...  
130 kilomètres à pied ça use, ça use ...  
131 kilomètres à pied ça use, ça use ...  
132 kilomètres à pied ça use, ça use ...  
133 kilomètres à pied ça use, ça use ...  
134 kilomètres à pied ça use, ça use ...  
135 kilomètres à pied ça use, ça use ...  
136 kilomètres à pied ça use, ça use ...  
137 kilomètres à pied ça use, ça use ...  
138 kilomètres à pied ça use, ça use ...  
139 kilomètres à pied ça use, ça use ...  
140 kilomètres à pied ça use, ça use ...  
141 kilomètres à pied ça use, ça use ...  
142 kilomètres à pied ça use, ça use ...  
143 kilomètres à pied ça use, ça use ...  
144 kilomètres à pied ça use, ça use ...  
145 kilomètres à pied ça use, ça use ...  
146 kilomètres à pied ça use, ça use ...  
147 kilomètres à pied ça use, ça use ...  
148 kilomètres à pied ça use, ça use ...  
149 kilomètres à pied ça use, ça use ...  
150 kilomètres à pied ça use, ça use ...  
151 kilomètres à pied ça use, ça use ...  
152 kilomètres à pied ça use, ça use ...  
153 kilomètres à pied ça use, ça use ...  
154 kilomètres à pied ça use, ça use ...  
155 kilomètres à pied ça use, ça use ...

156 kilomètres à pied ça use, ça use ...  
157 kilomètres à pied ça use, ça use ...  
158 kilomètres à pied ça use, ça use ...  
159 kilomètres à pied ça use, ça use ...  
160 kilomètres à pied ça use, ça use ...  
161 kilomètres à pied ça use, ça use ...  
162 kilomètres à pied ça use, ça use ...  
163 kilomètres à pied ça use, ça use ...  
164 kilomètres à pied ça use, ça use ...  
165 kilomètres à pied ça use, ça use ...  
166 kilomètres à pied ça use, ça use ...  
167 kilomètres à pied ça use, ça use ...  
168 kilomètres à pied ça use, ça use ...  
169 kilomètres à pied ça use, ça use ...  
170 kilomètres à pied ça use, ça use ...  
171 kilomètres à pied ça use, ça use ...  
172 kilomètres à pied ça use, ça use ...  
173 kilomètres à pied ça use, ça use ...  
174 kilomètres à pied ça use, ça use ...  
175 kilomètres à pied ça use, ça use ...  
176 kilomètres à pied ça use, ça use ...  
177 kilomètres à pied ça use, ça use ...  
178 kilomètres à pied ça use, ça use ...  
179 kilomètres à pied ça use, ça use ...  
180 kilomètres à pied ça use, ça use ...  
181 kilomètres à pied ça use, ça use ...  
182 kilomètres à pied ça use, ça use ...  
183 kilomètres à pied ça use, ça use ...  
184 kilomètres à pied ça use, ça use ...  
185 kilomètres à pied ça use, ça use ...  
186 kilomètres à pied ça use, ça use ...

187 kilomètres à pied ça use, ça use ...  
188 kilomètres à pied ça use, ça use ...  
189 kilomètres à pied ça use, ça use ...  
190 kilomètres à pied ça use, ça use ...  
191 kilomètres à pied ça use, ça use ...  
192 kilomètres à pied ça use, ça use ...  
193 kilomètres à pied ça use, ça use ...  
194 kilomètres à pied ça use, ça use ...  
195 kilomètres à pied ça use, ça use ...  
196 kilomètres à pied ça use, ça use ...  
197 kilomètres à pied ça use, ça use ...  
198 kilomètres à pied ça use, ça use ...  
199 kilomètres à pied ça use, ça use ...  
200 kilomètres à pied ça use, ça use ...  
201 kilomètres à pied ça use, ça use ...  
202 kilomètres à pied ça use, ça use ...  
203 kilomètres à pied ça use, ça use ...  
204 kilomètres à pied ça use, ça use ...  
205 kilomètres à pied ça use, ça use ...  
206 kilomètres à pied ça use, ça use ...  
207 kilomètres à pied ça use, ça use ...  
208 kilomètres à pied ça use, ça use ...  
209 kilomètres à pied ça use, ça use ...  
210 kilomètres à pied ça use, ça use ...  
211 kilomètres à pied ça use, ça use ...  
212 kilomètres à pied ça use, ça use ...  
213 kilomètres à pied ça use, ça use ...  
214 kilomètres à pied ça use, ça use ...  
215 kilomètres à pied ça use, ça use ...  
216 kilomètres à pied ça use, ça use ...  
217 kilomètres à pied ça use, ça use ...

218 kilomètres à pied ça use, ça use ...  
219 kilomètres à pied ça use, ça use ...  
220 kilomètres à pied ça use, ça use ...  
221 kilomètres à pied ça use, ça use ...  
222 kilomètres à pied ça use, ça use ...  
223 kilomètres à pied ça use, ça use ...  
224 kilomètres à pied ça use, ça use ...  
225 kilomètres à pied ça use, ça use ...  
226 kilomètres à pied ça use, ça use ...  
227 kilomètres à pied ça use, ça use ...  
228 kilomètres à pied ça use, ça use ...  
229 kilomètres à pied ça use, ça use ...  
230 kilomètres à pied ça use, ça use ...  
231 kilomètres à pied ça use, ça use ...  
232 kilomètres à pied ça use, ça use ...  
233 kilomètres à pied ça use, ça use ...  
234 kilomètres à pied ça use, ça use ...  
235 kilomètres à pied ça use, ça use ...  
236 kilomètres à pied ça use, ça use ...  
237 kilomètres à pied ça use, ça use ...  
238 kilomètres à pied ça use, ça use ...  
239 kilomètres à pied ça use, ça use ...  
240 kilomètres à pied ça use, ça use ...  
241 kilomètres à pied ça use, ça use ...  
242 kilomètres à pied ça use, ça use ...  
243 kilomètres à pied ça use, ça use ...  
244 kilomètres à pied ça use, ça use ...  
245 kilomètres à pied ça use, ça use ...  
246 kilomètres à pied ça use, ça use ...  
247 kilomètres à pied ça use, ça use ...  
248 kilomètres à pied ça use, ça use ...

249 kilomètres à pied ça use, ça use ...  
250 kilomètres à pied ça use, ça use ...  
251 kilomètres à pied ça use, ça use ...  
252 kilomètres à pied ça use, ça use ...  
253 kilomètres à pied ça use, ça use ...  
254 kilomètres à pied ça use, ça use ...  
255 kilomètres à pied ça use, ça use ...  
256 kilomètres à pied ça use, ça use ...  
257 kilomètres à pied ça use, ça use ...  
258 kilomètres à pied ça use, ça use ...  
259 kilomètres à pied ça use, ça use ...  
260 kilomètres à pied ça use, ça use ...  
261 kilomètres à pied ça use, ça use ...  
262 kilomètres à pied ça use, ça use ...  
263 kilomètres à pied ça use, ça use ...  
264 kilomètres à pied ça use, ça use ...  
265 kilomètres à pied ça use, ça use ...  
266 kilomètres à pied ça use, ça use ...  
267 kilomètres à pied ça use, ça use ...  
268 kilomètres à pied ça use, ça use ...  
269 kilomètres à pied ça use, ça use ...  
270 kilomètres à pied ça use, ça use ...  
271 kilomètres à pied ça use, ça use ...  
272 kilomètres à pied ça use, ça use ...  
273 kilomètres à pied ça use, ça use ...  
274 kilomètres à pied ça use, ça use ...  
275 kilomètres à pied ça use, ça use ...  
276 kilomètres à pied ça use, ça use ...  
277 kilomètres à pied ça use, ça use ...  
278 kilomètres à pied ça use, ça use ...  
279 kilomètres à pied ça use, ça use ...

280 kilomètres à pied ça use, ça use ...  
281 kilomètres à pied ça use, ça use ...  
282 kilomètres à pied ça use, ça use ...  
283 kilomètres à pied ça use, ça use ...  
284 kilomètres à pied ça use, ça use ...  
285 kilomètres à pied ça use, ça use ...  
286 kilomètres à pied ça use, ça use ...  
287 kilomètres à pied ça use, ça use ...  
288 kilomètres à pied ça use, ça use ...  
289 kilomètres à pied ça use, ça use ...  
290 kilomètres à pied ça use, ça use ...  
291 kilomètres à pied ça use, ça use ...  
292 kilomètres à pied ça use, ça use ...  
293 kilomètres à pied ça use, ça use ...  
294 kilomètres à pied ça use, ça use ...  
295 kilomètres à pied ça use, ça use ...  
296 kilomètres à pied ça use, ça use ...  
297 kilomètres à pied ça use, ça use ...  
298 kilomètres à pied ça use, ça use ...  
299 kilomètres à pied ça use, ça use ...  
300 kilomètres à pied ça use, ça use ...  
301 kilomètres à pied ça use, ça use ...  
302 kilomètres à pied ça use, ça use ...  
303 kilomètres à pied ça use, ça use ...  
304 kilomètres à pied ça use, ça use ...  
305 kilomètres à pied ça use, ça use ...  
306 kilomètres à pied ça use, ça use ...  
307 kilomètres à pied ça use, ça use ...  
308 kilomètres à pied ça use, ça use ...  
309 kilomètres à pied ça use, ça use ...  
310 kilomètres à pied ça use, ça use ...

311 kilomètres à pied ça use, ça use ...  
312 kilomètres à pied ça use, ça use ...  
313 kilomètres à pied ça use, ça use ...  
314 kilomètres à pied ça use, ça use ...  
315 kilomètres à pied ça use, ça use ...  
316 kilomètres à pied ça use, ça use ...  
317 kilomètres à pied ça use, ça use ...  
318 kilomètres à pied ça use, ça use ...  
319 kilomètres à pied ça use, ça use ...  
320 kilomètres à pied ça use, ça use ...  
321 kilomètres à pied ça use, ça use ...  
322 kilomètres à pied ça use, ça use ...  
323 kilomètres à pied ça use, ça use ...  
324 kilomètres à pied ça use, ça use ...  
325 kilomètres à pied ça use, ça use ...  
326 kilomètres à pied ça use, ça use ...  
327 kilomètres à pied ça use, ça use ...  
328 kilomètres à pied ça use, ça use ...  
329 kilomètres à pied ça use, ça use ...  
330 kilomètres à pied ça use, ça use ...  
331 kilomètres à pied ça use, ça use ...  
332 kilomètres à pied ça use, ça use ...  
333 kilomètres à pied ça use, ça use ...  
334 kilomètres à pied ça use, ça use ...  
335 kilomètres à pied ça use, ça use ...  
336 kilomètres à pied ça use, ça use ...  
337 kilomètres à pied ça use, ça use ...  
338 kilomètres à pied ça use, ça use ...  
339 kilomètres à pied ça use, ça use ...  
340 kilomètres à pied ça use, ça use ...  
341 kilomètres à pied ça use, ça use ...

342 kilomètres à pied ça use, ça use ...  
343 kilomètres à pied ça use, ça use ...  
344 kilomètres à pied ça use, ça use ...  
345 kilomètres à pied ça use, ça use ...  
346 kilomètres à pied ça use, ça use ...  
347 kilomètres à pied ça use, ça use ...  
348 kilomètres à pied ça use, ça use ...  
349 kilomètres à pied ça use, ça use ...  
350 kilomètres à pied ça use, ça use ...  
351 kilomètres à pied ça use, ça use ...  
352 kilomètres à pied ça use, ça use ...  
353 kilomètres à pied ça use, ça use ...  
354 kilomètres à pied ça use, ça use ...  
355 kilomètres à pied ça use, ça use ...  
356 kilomètres à pied ça use, ça use ...  
357 kilomètres à pied ça use, ça use ...  
358 kilomètres à pied ça use, ça use ...  
359 kilomètres à pied ça use, ça use ...  
360 kilomètres à pied ça use, ça use ...  
361 kilomètres à pied ça use, ça use ...  
362 kilomètres à pied ça use, ça use ...  
363 kilomètres à pied ça use, ça use ...  
364 kilomètres à pied ça use, ça use ...  
365 kilomètres à pied ça use, ça use ...  
366 kilomètres à pied ça use, ça use ...  
367 kilomètres à pied ça use, ça use ...  
368 kilomètres à pied ça use, ça use ...  
369 kilomètres à pied ça use, ça use ...  
370 kilomètres à pied ça use, ça use ...  
371 kilomètres à pied ça use, ça use ...  
372 kilomètres à pied ça use, ça use ...

373 kilomètres à pied ça use, ça use ...  
374 kilomètres à pied ça use, ça use ...  
375 kilomètres à pied ça use, ça use ...  
376 kilomètres à pied ça use, ça use ...  
377 kilomètres à pied ça use, ça use ...  
378 kilomètres à pied ça use, ça use ...  
379 kilomètres à pied ça use, ça use ...  
380 kilomètres à pied ça use, ça use ...  
381 kilomètres à pied ça use, ça use ...  
382 kilomètres à pied ça use, ça use ...  
383 kilomètres à pied ça use, ça use ...  
384 kilomètres à pied ça use, ça use ...  
385 kilomètres à pied ça use, ça use ...  
386 kilomètres à pied ça use, ça use ...  
387 kilomètres à pied ça use, ça use ...  
388 kilomètres à pied ça use, ça use ...  
389 kilomètres à pied ça use, ça use ...  
390 kilomètres à pied ça use, ça use ...  
391 kilomètres à pied ça use, ça use ...  
392 kilomètres à pied ça use, ça use ...  
393 kilomètres à pied ça use, ça use ...  
394 kilomètres à pied ça use, ça use ...  
395 kilomètres à pied ça use, ça use ...  
396 kilomètres à pied ça use, ça use ...  
397 kilomètres à pied ça use, ça use ...  
398 kilomètres à pied ça use, ça use ...  
399 kilomètres à pied ça use, ça use ...  
400 kilomètres à pied ça use, ça use ...  
401 kilomètres à pied ça use, ça use ...  
402 kilomètres à pied ça use, ça use ...  
403 kilomètres à pied ça use, ça use ...

404 kilomètres à pied ça use, ça use ...  
405 kilomètres à pied ça use, ça use ...  
406 kilomètres à pied ça use, ça use ...  
407 kilomètres à pied ça use, ça use ...  
408 kilomètres à pied ça use, ça use ...  
409 kilomètres à pied ça use, ça use ...  
410 kilomètres à pied ça use, ça use ...  
411 kilomètres à pied ça use, ça use ...  
412 kilomètres à pied ça use, ça use ...  
413 kilomètres à pied ça use, ça use ...  
414 kilomètres à pied ça use, ça use ...  
415 kilomètres à pied ça use, ça use ...  
416 kilomètres à pied ça use, ça use ...  
417 kilomètres à pied ça use, ça use ...  
418 kilomètres à pied ça use, ça use ...  
419 kilomètres à pied ça use, ça use ...  
420 kilomètres à pied ça use, ça use ...  
421 kilomètres à pied ça use, ça use ...  
422 kilomètres à pied ça use, ça use ...  
423 kilomètres à pied ça use, ça use ...  
424 kilomètres à pied ça use, ça use ...  
425 kilomètres à pied ça use, ça use ...  
426 kilomètres à pied ça use, ça use ...  
427 kilomètres à pied ça use, ça use ...  
428 kilomètres à pied ça use, ça use ...  
429 kilomètres à pied ça use, ça use ...  
430 kilomètres à pied ça use, ça use ...  
431 kilomètres à pied ça use, ça use ...  
432 kilomètres à pied ça use, ça use ...  
433 kilomètres à pied ça use, ça use ...  
434 kilomètres à pied ça use, ça use ...

435 kilomètres à pied ça use, ça use ...  
436 kilomètres à pied ça use, ça use ...  
437 kilomètres à pied ça use, ça use ...  
438 kilomètres à pied ça use, ça use ...  
439 kilomètres à pied ça use, ça use ...  
440 kilomètres à pied ça use, ça use ...  
441 kilomètres à pied ça use, ça use ...  
442 kilomètres à pied ça use, ça use ...  
443 kilomètres à pied ça use, ça use ...  
444 kilomètres à pied ça use, ça use ...  
445 kilomètres à pied ça use, ça use ...  
446 kilomètres à pied ça use, ça use ...  
447 kilomètres à pied ça use, ça use ...  
448 kilomètres à pied ça use, ça use ...  
449 kilomètres à pied ça use, ça use ...  
450 kilomètres à pied ça use, ça use ...  
451 kilomètres à pied ça use, ça use ...  
452 kilomètres à pied ça use, ça use ...  
453 kilomètres à pied ça use, ça use ...  
454 kilomètres à pied ça use, ça use ...  
455 kilomètres à pied ça use, ça use ...  
456 kilomètres à pied ça use, ça use ...  
457 kilomètres à pied ça use, ça use ...  
458 kilomètres à pied ça use, ça use ...  
459 kilomètres à pied ça use, ça use ...  
460 kilomètres à pied ça use, ça use ...  
461 kilomètres à pied ça use, ça use ...  
462 kilomètres à pied ça use, ça use ...  
463 kilomètres à pied ça use, ça use ...  
464 kilomètres à pied ça use, ça use ...  
465 kilomètres à pied ça use, ça use ...

466 kilomètres à pied ça use, ça use ...  
467 kilomètres à pied ça use, ça use ...  
468 kilomètres à pied ça use, ça use ...  
469 kilomètres à pied ça use, ça use ...  
470 kilomètres à pied ça use, ça use ...  
471 kilomètres à pied ça use, ça use ...  
472 kilomètres à pied ça use, ça use ...  
473 kilomètres à pied ça use, ça use ...  
474 kilomètres à pied ça use, ça use ...  
475 kilomètres à pied ça use, ça use ...  
476 kilomètres à pied ça use, ça use ...  
477 kilomètres à pied ça use, ça use ...  
478 kilomètres à pied ça use, ça use ...  
479 kilomètres à pied ça use, ça use ...  
480 kilomètres à pied ça use, ça use ...  
481 kilomètres à pied ça use, ça use ...  
482 kilomètres à pied ça use, ça use ...  
483 kilomètres à pied ça use, ça use ...  
484 kilomètres à pied ça use, ça use ...  
485 kilomètres à pied ça use, ça use ...  
486 kilomètres à pied ça use, ça use ...  
487 kilomètres à pied ça use, ça use ...  
488 kilomètres à pied ça use, ça use ...  
489 kilomètres à pied ça use, ça use ...  
490 kilomètres à pied ça use, ça use ...  
491 kilomètres à pied ça use, ça use ...  
492 kilomètres à pied ça use, ça use ...  
493 kilomètres à pied ça use, ça use ...  
494 kilomètres à pied ça use, ça use ...  
495 kilomètres à pied ça use, ça use ...  
496 kilomètres à pied ça use, ça use ...

497 kilomètres à pied ça use, ça use ...  
498 kilomètres à pied ça use, ça use ...  
499 kilomètres à pied ça use, ça use ...  
500 kilomètres à pied ça use, ça use ...  
501 kilomètres à pied ça use, ça use ...  
502 kilomètres à pied ça use, ça use ...  
503 kilomètres à pied ça use, ça use ...  
504 kilomètres à pied ça use, ça use ...  
505 kilomètres à pied ça use, ça use ...  
506 kilomètres à pied ça use, ça use ...  
507 kilomètres à pied ça use, ça use ...  
508 kilomètres à pied ça use, ça use ...  
509 kilomètres à pied ça use, ça use ...  
510 kilomètres à pied ça use, ça use ...  
511 kilomètres à pied ça use, ça use ...  
512 kilomètres à pied ça use, ça use ...  
513 kilomètres à pied ça use, ça use ...  
514 kilomètres à pied ça use, ça use ...  
515 kilomètres à pied ça use, ça use ...  
516 kilomètres à pied ça use, ça use ...  
517 kilomètres à pied ça use, ça use ...  
518 kilomètres à pied ça use, ça use ...  
519 kilomètres à pied ça use, ça use ...  
520 kilomètres à pied ça use, ça use ...  
521 kilomètres à pied ça use, ça use ...  
522 kilomètres à pied ça use, ça use ...  
523 kilomètres à pied ça use, ça use ...  
524 kilomètres à pied ça use, ça use ...  
525 kilomètres à pied ça use, ça use ...  
526 kilomètres à pied ça use, ça use ...  
527 kilomètres à pied ça use, ça use ...

528 kilomètres à pied ça use, ça use ...  
529 kilomètres à pied ça use, ça use ...  
530 kilomètres à pied ça use, ça use ...  
531 kilomètres à pied ça use, ça use ...  
532 kilomètres à pied ça use, ça use ...  
533 kilomètres à pied ça use, ça use ...  
534 kilomètres à pied ça use, ça use ...  
535 kilomètres à pied ça use, ça use ...  
536 kilomètres à pied ça use, ça use ...  
537 kilomètres à pied ça use, ça use ...  
538 kilomètres à pied ça use, ça use ...  
539 kilomètres à pied ça use, ça use ...  
540 kilomètres à pied ça use, ça use ...  
541 kilomètres à pied ça use, ça use ...  
542 kilomètres à pied ça use, ça use ...  
543 kilomètres à pied ça use, ça use ...  
544 kilomètres à pied ça use, ça use ...  
545 kilomètres à pied ça use, ça use ...  
546 kilomètres à pied ça use, ça use ...  
547 kilomètres à pied ça use, ça use ...  
548 kilomètres à pied ça use, ça use ...  
549 kilomètres à pied ça use, ça use ...  
550 kilomètres à pied ça use, ça use ...  
551 kilomètres à pied ça use, ça use ...  
552 kilomètres à pied ça use, ça use ...  
553 kilomètres à pied ça use, ça use ...  
554 kilomètres à pied ça use, ça use ...  
555 kilomètres à pied ça use, ça use ...  
556 kilomètres à pied ça use, ça use ...  
557 kilomètres à pied ça use, ça use ...  
558 kilomètres à pied ça use, ça use ...

559 kilomètres à pied ça use, ça use ...  
560 kilomètres à pied ça use, ça use ...  
561 kilomètres à pied ça use, ça use ...  
562 kilomètres à pied ça use, ça use ...  
563 kilomètres à pied ça use, ça use ...  
564 kilomètres à pied ça use, ça use ...  
565 kilomètres à pied ça use, ça use ...  
566 kilomètres à pied ça use, ça use ...  
567 kilomètres à pied ça use, ça use ...  
568 kilomètres à pied ça use, ça use ...  
569 kilomètres à pied ça use, ça use ...  
570 kilomètres à pied ça use, ça use ...  
571 kilomètres à pied ça use, ça use ...  
572 kilomètres à pied ça use, ça use ...  
573 kilomètres à pied ça use, ça use ...  
574 kilomètres à pied ça use, ça use ...  
575 kilomètres à pied ça use, ça use ...  
576 kilomètres à pied ça use, ça use ...  
577 kilomètres à pied ça use, ça use ...  
578 kilomètres à pied ça use, ça use ...  
579 kilomètres à pied ça use, ça use ...  
580 kilomètres à pied ça use, ça use ...  
581 kilomètres à pied ça use, ça use ...  
582 kilomètres à pied ça use, ça use ...  
583 kilomètres à pied ça use, ça use ...  
584 kilomètres à pied ça use, ça use ...  
585 kilomètres à pied ça use, ça use ...  
586 kilomètres à pied ça use, ça use ...  
587 kilomètres à pied ça use, ça use ...  
588 kilomètres à pied ça use, ça use ...  
589 kilomètres à pied ça use, ça use ...

590 kilomètres à pied ça use, ça use ...  
591 kilomètres à pied ça use, ça use ...  
592 kilomètres à pied ça use, ça use ...  
593 kilomètres à pied ça use, ça use ...  
594 kilomètres à pied ça use, ça use ...  
595 kilomètres à pied ça use, ça use ...  
596 kilomètres à pied ça use, ça use ...  
597 kilomètres à pied ça use, ça use ...  
598 kilomètres à pied ça use, ça use ...  
599 kilomètres à pied ça use, ça use ...  
600 kilomètres à pied ça use, ça use ...

Il est désormais facile de compter jusqu'à 100, 1000 ou plus.

## EXERCICE 2 :

Dans le programme suivant, quelle est la variable d'itération ?  
Qu'elle est sa valeur à la fin de la boucle ?  
Qu'affiche t-il ? En particulier, que se passe t-il sur la première ligne ?

```
N = 7
for nb_etoile in range(N):
    print("*" * nb_etoile)
```

## Exemple 3 : Découvrons le module `Turtle`

Le module `Turtle` est un module Python inspiré de la "*programmation Logo*". Des fonctions définies dans ce module permettent de déplacer une tortue sur l'écran et d'afficher ou non sa trace à l'écran.

Par exemple, on trouve les fonctions :

- `forward(dist)` : Avance la tortue d'une distance de `dist` pixels
- `backward(dist)` : Recule d'une distance donnée
- `left(alpha)` : Pivote la tortue vers la gauche d'un angle de `alpha` degrés
- `right(alpha)` : Pivote la tortue vers la droite d'un angle de `alpha` degrés
- `up()` : Relève le crayon ; cela permet d'avancer sans dessiner

- `down ( )` : Abaisse le crayon, pour recommencer à dessiner
- `goto (x, y)` : Envoie la tortue au point de coordonnées (x, y) en ligne droite

Ce module permet de faire de très jolis dessins. Par exemple, on peut tracer des polygones réguliers facilement.

In [4]:

```
"""
Tracé de polygones réguliers.
"""
# On importe les fonctions du module turtle
import turtle

# Initialisation des paramètres
nbCotes = 4
lgCote = 250
angle = 360 / nbCotes

# Tracé
for compteur in range(nbCotes):
    turtle.forward(lgCote) # Tracer un côté
    turtle.left(angle)    # Tourner à gauche

turtle.exitonclick() # Sortie par click de souris
```

## Vérifier vos connaissances

A ce stade, vous devriez être capable :

- de comprendre la notion de boucle ;
- d'expliquer le fonctionnement d'une boucle `for ... in range(...)` ;
- d'identifier la variable d'itération et de comprendre comment elle évolue au cours d'une boucle `for ... in range(...)` ;
- de détecter la nécessité d'introduire une boucle `for ... in range(...)` dans un programme ;
- d'écrire correctement une boucle `for ... in range(...)`.

### EXERCICE 3 : LES TABLES DE MULTIPLICATIONS

Écrire un programme de trois lignes qui affiche la table de multiplication par 7 :

```
1 * 7 = 7  
2 * 7 = 14  
...  
10 * 7 = 70
```

**Attention :** La ligne  $0 * 7 = 0$  ne doit apparaître, et vous ne devez pas utiliser de `if` !

In [ ]:

## EXERCICE 4

Écrivez un programme qui lit un entier positif  $n$  donné par l'utilisateur, puis  $n$  nombres entiers (toujours entrés par l'utilisateur) et compte et affiche en fin de programme combien parmi ces valeurs sont strictement positives, strictement négatives, nulles.

In [ ]:

## EXERCICE 5

1. Écrivez un programme qui demande à l'utilisateur un entier et affiche la factorielle de ce nombre.
2. Modifiez votre programme pour afficher toutes les factorielles des nombres inférieurs au nombre saisi.

*Remarque : Pour la question 2, il suffit de rajouter une ligne de code.*

In [ ]:

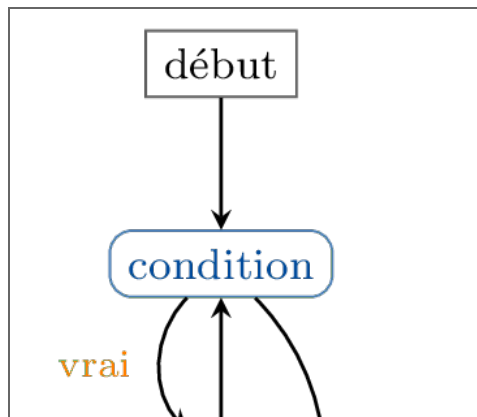
### III Répétition d'une tâche tant qu'une condition est vérifiée :

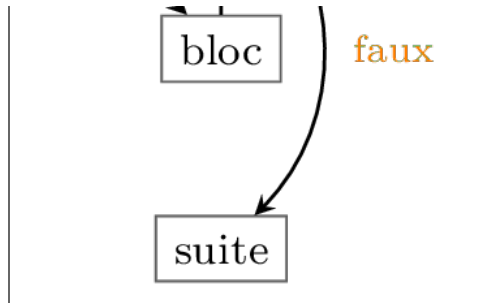
`while . . . :`

On peut répéter un bloc d'instructions un nombre de fois inconnu avant le début de la répétition grâce à une boucle `tant que`.

#### Principe de base :

- **Si** une certaine condition est vraie, on va exécuter un certain bloc d'instructions ;
- **Sinon**, on va passer directement à la suite du programme ;
- Après chaque exécution du bloc, on réévalue la condition.





## Syntaxe :

```
# début  
while <condition>:  
    # bloc d'instructions  
    # (corps de la boucle)  
# suite
```

où `condition` est une **expression booléenne**.

## Vocabulaire :

- Le bloc d'instructions est appelé **corps de la boucle**.
- Chaque exécution du corps de la boucle est appelée **itération**.

## Fonctionnement détaillé :

La `condition` est une expression booléenne dépendant de variables définies dans la partie `début`. Le corps de la boucle est alors exécuté une première fois uniquement si `condition` est évaluée à `True`. Au cours de chaque exécution du corps de la boucle, les variables dont dépend `condition` peuvent (doivent !)

avoir évoluées. Après chaque exécution du corps, on réévalue donc condition :

- Si `condition` vaut `False`, le corps de la boucle n'est pas exécuté ; à la place, on exécute la suite du code.
- Si `condition` vaut `True`, le corps de la boucle est (de nouveau) exécuté.

**Remarque :** Il peut n'y avoir aucune itération, ou un nombre infini en fonction de la condition !

La boucle `while` est très générale. Elle permet notamment de reproduire le comportement d'une boucle `for ... in range(...)`. Néanmoins

- la boucle `for ... in range(...)` se trouve être plus simple à manipuler lorsque l'on peut l'utiliser (voir Exemple 1)
- la boucle `while` ne permet pas de réaliser toutes les boucles que l'on pourrait vouloir faire (voir Exemple 2). Notamment, une boucle `while` est nécessaire si l'on ne connaît pas à l'avance le nombre d'itérations à effectuer.

Exemple 1 : une boucle `while` permet de simuler une boucle `for`  
`... in range(...)`

In [1]:

```
# Simulation par une boucle `while` de la boucle `for <var> in range(...)`

nombre = int(input("Entrez un nombre : "))

var = 0 # Initialiser le compteur
while var < nombre:
    print(f'{var + 1} kilomètre(s) à pied ça use, ça use ...')
    var = var + 1 # Ne pas oublier !!!
```

Entrez un nombre : 23

1 kilomètre(s) à pied ça use, ça use ...  
2 kilomètre(s) à pied ça use, ça use ...  
3 kilomètre(s) à pied ça use, ça use ...  
4 kilomètre(s) à pied ça use, ça use ...  
5 kilomètre(s) à pied ça use, ça use ...  
6 kilomètre(s) à pied ça use, ça use ...  
7 kilomètre(s) à pied ça use, ça use ...  
8 kilomètre(s) à pied ça use, ça use ...  
9 kilomètre(s) à pied ça use, ça use ...  
10 kilomètre(s) à pied ça use, ça use ...  
11 kilomètre(s) à pied ça use, ça use ...  
12 kilomètre(s) à pied ça use, ça use ...  
13 kilomètre(s) à pied ça use, ça use ...  
14 kilomètre(s) à pied ça use, ça use ...  
15 kilomètre(s) à pied ça use, ça use ...  
16 kilomètre(s) à pied ça use, ça use ...  
17 kilomètre(s) à pied ça use, ça use ...  
18 kilomètre(s) à pied ça use, ça use ...  
19 kilomètre(s) à pied ça use, ça use ...  
20 kilomètre(s) à pied ça use, ça use ...  
21 kilomètre(s) à pied ça use, ça use ...  
22 kilomètre(s) à pied ça use, ça use ...  
23 kilomètre(s) à pied ça use, ça use ...

**Que vaut la variable d'itération à l'issue de la boucle ?**

Dans cet exemple, la variable d'itération serait `var`. Le corps de la boucle contient deux instructions : le `print` et l'incrémentement de `var`.



La difficulté de ce type de boucle est de ne pas oublier l'incrémentement de la variable d'itération. C'est une erreur commune faite par beaucoup d'étudiant qui ne se retrouve pas si l'on utilise la syntaxe `for ... in range(...)` qui reste plus simple dans ce genre de cas !

In [42]:

```
var
```

Out[42]:

10

Ah... Elle aurait valoir 9... On a donc **presque** simulé la boucle `for ... in range(...)`

## Exemple 2 : Saisie contrôlée

Lorsque l'on demande à l'utilisateur une donnée, on se doit de vérifier qu'il saisit une donnée valide avant même d'effectuer le moindre traitement sur cette donnée.

Une saisie sera dite **saisie contrôlée** lorsque le programmeur demande à l'utilisateur de resaisir sa donnée tant que celle-ci n'est pas valide. Il s'agit donc d'une boucle `while`, dont la condition traduit *"la donnée n'est pas valide"*.

Dans le cas où l'on veut, par exemple, un entier `nb` positif, la condition de la boucle est : `nb < 0`. D'où le programme suivant :

In [1]:

```
# On veut saisir faire saisir à l'utilisateur en entier positif (ou nul)  
nb = int(input('Donnez moi en entier positif (en chiffres) : '))  
while nb < 0:  
    print('Erreur de saisie.')  
    nb = int(input('Donnez moi en entier positif (en chiffres) : '))
```

Donnez moi en entier positif (en chiffres) : -3

Erreur de saisie.

Donnez moi en entier positif (en chiffres) : 31

In [51]:

```
nb
```

Out[51]:

-3

**Remarque :** Si l'utilisateur ne rentre pas un entier, la conversion en entier va provoquer une erreur. Notre saisie contrôlée n'est donc pas parfaite... Elle suppose que l'utilisateur donnera tout de même un entier...

On peut aussi faire des saisies contrôlées dont l'objet est un peu plus complexe. Par exemple, saisir un nombre entier entre 0 et 20 :

In [50]:

```
nb = int(input('Donnez moi un entier entre 0 et 20 (en chiffres) : '))
while nb < 0 or nb > 20: # Négation de 0 <= nb ET nb <= 20
    print('Erreur de saisie.')
    nb = int(input('Donnez moi un entier entre 0 et 20 (en chiffres) : '))
```

Donnez moi en entier positif (en chiffres) : -3

**Remarque :**

$$\text{Non}(0 \leq \text{nb} \leq 20) = \text{Non}(0 \leq \text{nb} \text{ ET } \text{nb} \leq 20)$$

$$= \text{Non}(0 \leq \text{nb}) \text{ OU } \text{Non}(\text{nb} \leq 20)$$

$$= \text{nb} < 0 \text{ OU } \text{nb} > 20$$

## EXERCICE 6 :

Au cours précédent, nous avons écrit un programme permettant de calculer la moyenne d'un cours contenant deux contrôles continus et un examen.

Ajoutez des saisies contrôlées lors de la saisie des trois notes pour rendre le programme un peu plus robuste.

In [ ]:

```
# La note finale est le maximum de la note d'examen  
# et de la moyenne entre examen et moyenne de contrôle  
# continu  
  
note_cc1 = float(input('Note du premier contrôle : '))  
note_cc2 = float(input('Note du second contrôle : '))  
note_exam = float(input("Note de l'examen : "))  
  
if note_cc1 < 0 or note_cc2 < 0 or note_exam < 0:  
    print('Erreur de saisie (note négative).')  
else:  
    moyenne_cc = (note_cc1 + note_cc2) / 2  
  
    if moyenne_cc < note_exam:  
        note_finale = note_exam  
        print('Note de contrôle non prise en compte.')  
    else:  
        note_finale = (moyenne_cc + note_exam) / 2  
        print('Note de contrôle prise en compte.')  
    print('Note finale : ', note_finale)  
  
    if note_finale < 10:  
        print('Encore un petit effort !')  
    elif note_finale < 12:  
        print('Ça passe !')  
    elif note_finale < 14:
```

```
    print('Pas mal !')  
else: # Quelles valeurs possibles ?  
    print('Bravo !')
```

### Exemple 3 : la racine carrée entière

Soit  $N$  un entier. On souhaite trouver le plus grand entier `racine` tel que `racine * racine <= N`, en incrémentant progressivement la valeur de `n`.

In [1]:

```
N = 142
racine = 0
while racine * racine <= N:
    racine += 1 # même chose que racine = racine + 1
# Ici, racine ** 2 > N
racine -= 1 # même chose racine = racine - 1
racine
```

Out[1]:

11



\*A quel moment la condition n'est-elle plus vérifiée ? Cela explique pourquoi il faut afficher `n - 1` et non `n`.\*

In [59]:

```
# Vérification du calcul précédent :  
N = 142  
int(N ** (1/2))
```

Out[59]:

11

Vérifier vos connaissances :

A ce stade, vous devriez être capable :

- de comprendre la notion de boucle `while` ;
- décrire le fonctionnement d'une boucle `while` ;
- d'identifier la variable d'itération et de comprendre comment elle évolue au cours d'une boucle `while` ;
- détecter la nécessité d'introduire une boucle `while` dans un programme ;
- d'écrire correctement une boucle `while` .

## EXERCICE 7 : DE PLUS EN PLUS GRAND

Écrivez un programme qui laisse l'utilisateur saisir autant de nombres positifs qu'il veut, tant que chaque nombre est strictement supérieur à celui d'avant. Si un nombre saisi ne vérifie pas la règle, le programme s'arrête et indique combien de nombres comporte la suite strictement croissante saisie par l'utilisateur.

## EXERCICE 8 : MOYENNE DE NOTES

1. Écrivez un programme qui permet à l'utilisateur de saisir autant de nombres entiers qu'il veut et calcule et affiche la moyenne de ces nombres. Quand il a terminé l'utilisateur écrit `stop` au lieu d'un nombre.

**Remarque :** Exceptionnellement, on ne fera pas de saisie contrôlée dans cet exercice.

2. Modifiez le programme pour qu'il redemande chaque nombre saisi s'il n'est pas compris entre 0 et 20.

IV Comparaison entre les boucles `for ... in`  
`range(...)` et boucles `while`

Comme on vient de le voir, il existe deux façons de procéder pour écrire des boucles : `for ... in range(...)` et `while`. Il est important de choisir judicieusement quel type de boucle utiliser en fonction du problème à résoudre.

La boucle `for ... in range(...)` se trouve être plus simple à manipuler que la boucle `while`, mais elle ne permet pas de réaliser toutes les boucles que l'on pourrait vouloir faire. Toujours manipuler une boucle `while` pourrait suffire, mais cela donnerait des programmes très lourds à lire pour l'humain.

En conséquence, la boucle `for ... in range(...)` est le meilleur choix quand on connaît à l'avance le nombre de fois que l'on veut effectuer l'itération. À l'opposé, la boucle `while` doit être utilisée lorsque l'on ne connaît pas le nombre d'itérations à effectuer à l'avance.

### Comparaison des deux types de boucles

*# début*

N = ...

for <var> in range(N):

*# bloc d'instructions*

*# (corps de la boucle)*

*# suite*

Besoin de connaître à l'avance le nombre d'itérations

Ne permet pas de résoudre tous les problèmes

Simple à manipuler :

- *Incrémentation automatique*
- *Pas de condition d'arrêt à écrire*

*# début*

while <condition>:

*# bloc d'instructions*

*# (corps de la boucle)*

*# suite*

Pas besoin de connaître le nombre d'itérations

Permet de résoudre tous les problèmes

Risque d'erreurs :

- *oublie d'incrémentation ;*
- *difficulté à trouver la condition d'arrêt*

Vérifier vos connaissances :

A ce stade, vous devriez être capable :

- de citer les deux types de boucles en Python ;
- de citer les avantages d'une boucle `for ... in range(...)` sur une boucle `while` ;
- de décider si vous aller implémenter une boucle `for ... in range(...)` ou une boucle `while` ;
- d'argumenter vos choix de boucles dans vos programmes.

## EXERCICE 9 :

On considère le programme suivant :

```
actuel = 1
avant = 1
while actuel < 1000:
    print(actuel)
    actuel, avant = actuel + avant, actuel
```

1. Que réalise le programme suivant ?
2. Ce programme contient une boucle `while` . Est-ce un choix judicieux ? Si non, traduisez-le en un programme contenant une boucle `for`.

In [66]:

```
actuel = 1
avant = 1
while actuel < 1000:
    print(actuel)
    actuel, avant = actuel + avant, actuel
```

1  
2  
3  
5  
8  
13  
21  
34  
55  
89  
144  
233  
377  
610  
987

## EXERCICE 10 :

Pour chacune des situations suivantes, indiquer s'il vous utiliseriez plutôt une boucle `while` ou plutôt une boucle `for ... in range(...)` pour :

1. écrire un programme calculant les 100 premiers termes de la

suite définie par  $U_0 = 1$  et  $U_{n+1} = \frac{U_n}{3U_n + 2}$ .

2. écrire un programme demandant à l'utilisateur un mot de 5 lettres.

3. écrire un programme dessinant la forme suivante :

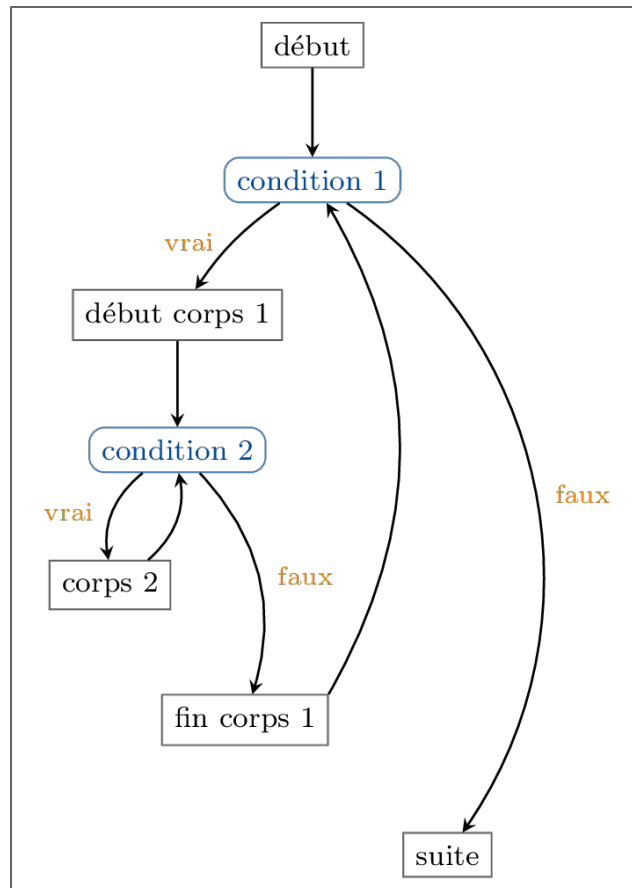
```
*      *  
 *    *  
  *  *  
   **
```

4. écrire un programme demandant à l'utilisateur un entier et déterminant combien de fois de suite cet entier est divisible par 2.

In [ ]:

## V Boucles imbriquées

Améliorons l'exercice 6. Nous avons des saisies de l'utilisateur qui ne sont pas contrôlées. **Ce n'est pas bien !!!** Nous avons donc besoin de faire une boucle à l'intérieur d'une autre boucle, ce qui est heureusement possible.



Double boucle `while` imbriquée

## Syntaxe d'une double boucle **while** :

```
# début
while <condition 1>:
    # début corps 1
    while <condition 2>:
        # corps 2
    # fin corps 1
# suite
```

## Syntaxe d'une boucle **while** contenant une boucle **for ... in range(...)**:

```
# début
while <condition 1>:
    # début corps 1
    N = ...
    for <variable> in range(N):
        # corps 2
    # fin corps 1
# suite
```

## Syntaxe d'une boucle **for ... in range(...)**: contenant une boucle **while**

```
# début
N = ...
for <variable> in range(N):
```

```
# début corps 1
while <condition 2>:
    # corps 2
# fin corps 1
# suite
```

## Syntaxe d'une double boucle for ... in range(...) :

```
# début
M =
for <variable> in range(M):
    # début corps 1
    N =
    for <variable> in range(N):
        # corps 2
    # fin corps 1
# suite
```

Exemple 1 : Ajout d'une saisie contrôlée dans le programme de l'exercice 8

In [ ]:

```
nb_nombre_saisi = 0
somme = 0
saisie = None # variable en fait pas réellement initialisée
while saisie != "stop":
    saisie = input("Donnez moi un entier : ")
    if saisie != "stop":
        nb = int(saisie)
        while nb < 0 or nb > 20:
            nb = int(input("Donnez moi un entier : "))
        nb_nombre_saisi += 1
        somme += nb
print("moyenne = ", somme / nb_nombre_saisi)
```

Exemple 2 : Dessin d'une rosace, avec le module Turtle :

In [38]:

```
"""
Tracé de rosaces régulières.
"""
# On importe les fonctions du module turtle
import turtle

# Paramètres
speed(0)

# Initialisation des paramètres
nb_cotes = 6
lg_cote = 50
nb_petales = 12
angle_rosace = 360 / nb_petales

# Tracé
for cmpt in range(nb_petales):
    # Dessin d'un polygone
    for compteur in range(nb_cotes):
        turtle.forward(lg_cote) # Tracer un côté
        turtle.left(angle_rosace) # Tourner à gauche
    print("compteur = ", compteur, "cmpt = ", cmpt)
    # Rotation de la tortue pour modifier l'angle de départ
    left(360 / nb_petales)

turtle.exitonclick() # Sortie par click de souris
```

```
compteur = 5 cmpt = 0
compteur = 5 cmpt = 1
compteur = 5 cmpt = 2
compteur = 5 cmpt = 3
compteur = 5 cmpt = 4
compteur = 5 cmpt = 5
compteur = 5 cmpt = 6
compteur = 5 cmpt = 7
compteur = 5 cmpt = 8
compteur = 5 cmpt = 9
```

Vérifier vos connaissances

A ce stade, vous devriez être capable :

- d'identifier les situations où l'on a besoin de boucles imbriquées ;
- d'écrire correctement des boucles imbriquées.

## EXERCICE 11 :

Écrire un programme qui affiche toutes les tables de multiplications de 1 à 10.