

Boucles II:

Tu peux contrôler comment une boucle se passe.

exemple:

- s'arrêter avant la fin

- sauter des itérations (rappel: une itération c'est 1 étape de la boucle)

de différentes manières.

ex: `for i in range(5)`

5 itérations

Break:

- interrompt la boucle:

- "Stop" le plus simple

Example:

```
for i in range(5):
    print(i)
    if i > 3:
        break
```

Output:

0 1 2 3 ↴ break

0 ← i=0
1
2
3 ← i=3

Continue:

- skip une itération

- La boucle fait toutes les itérations sauf celle où elle a rencontré un continue

```
for i in range(5):
    if i > 3:
        continue
    print(i)
```

i=0: 0
i=1: 1
i=2: 2
i=3: Ø (rien)
i=4: 4

```

for i in range(5):
    print(i)
    if i > 3:
        continue
    ↗ changer rien
    ↗ car continue est à la
    dernière instruction de
    l'itération.
    ↗ & la fin ↗ ça fait rien
  
```

Functions:

On définit le nombre de paramètres et on leur donne un nom

```

def nom_fonction(arg1, arg2, ..., argN): (on ne crée rien encore)
  ↗
  |
  |
  |
  |
  ↗ Corps
  ↗ de la
  ↗ fonction
  ↗ return Valeur (optionnel)
  
```

Exemple:

```

def print_list(table):
    for elem in table:
        print(elem)
  
```

return rien
 ↗ retourne pas de valeur
 calculée à l'intérieur

```

def add(a, b):
    return a + b
  
```

$a, b = 3, 4$
 $res = add(a, b)$
 # ici res vaut 7

Une fonction a :

- 1 pt d'entrée
- 1 pt de sortie

Sortie:

- Soit la fin de la fonction
- Soit un return

Généralement return est utilisé pour renvoyer une valeur calculée à l'intérieur de la fonction.

↳ Mais on peut utiliser return sans rien pour sortir de la fonction

def test(a, b):

print(a)

return ← point de sortie.

print(b) ← cette ligne

m'est jamais évalué

les variables définies dans le corps d'une fonction sont locales à cette fonction.

↳ Elle n'existent que dans cette fonction

def test2(a, b):

c = 42

print(a)

return b

print(c) ← # error: pas de variable

qui s'appelle "c"

pas visible

depuis ici.

Entrée: pour entrer dans une

ENTRÉE: pour entrer dans une fonction il faut l'appeler
↳ mom-fonction () valeurs des arguments

Exemple:

print(2)

def sayHello():

print("Hello")

sayHello()

appels de fonction

la valeur qu'on lui passe c'est le résultat de len([1, 2, 3])

print(len([1, 2, 3]))

appel print

la valeur de son argument est [1, 2, 3]

appel len