

Python : rappel

variables :

int

2, 100

float

2.5, 3.14, 5.0

bool

True

False

str

"hello"

''

list

[1, 2, 3]

[]

opérateurs :

+

-

*

/

//

%

**

← puissance

← int, float

↑
div
entière

↑
modulo

+ : list, str → concatenation

* int * str

```
>>> 5 * "ab"
```

```
'ababababab'
```

```
>>> "ab" + "cd"
```

```
'abcd'
```

```
>>> [1, 2] + [3, 4]
```

```
[1, 2, 3, 4]
```

- `print(...)` → affiche à l'écran

- `reponse = input(question)` → pose une question à l'utilisateur

- `if` condition 1:
bloc 1

- `elif` condition 2:
bloc 2

- `else :`
bloc final

exemple:

```
1 reponse = input("Veux-tu continuer? (O/N)")
2 if reponse == 'O':
3     print("Chouette, on continue!")
4 elif reponse == 'N':
5     print("Au revoir!")
6 else:
7     print("Mauvaise réponse!")
```

while

condition:

bloc

exemple:

```
while True:
    reponse = input("Veux-tu continuer? (O/N)")
    if reponse == 'O':
        print("Chouette, on continue!")
        break
    elif reponse == 'N':
        print("Au revoir!")
        break
    else:
        print("Mauvaise réponse!")
```

while True:

répète indéfiniment

"break" permet de sortir de la boucle infinie

pas d'espace! fonction ;

def nom_fonction (paramètres) :
 bloc

return . . . ← si on veut retourner
une valeur

```
1 def carre(nb):  
2     resultat = nb ** 2  
3     return resultat  
4  
5 print( carre(2) )
```

boucles

for

str, list, range

for

var

in

collection :

bloc

avec une liste

```
1 # Boucle for avec une liste:
2 mots = ['chat', 'fenêtre', 'défenestrer']
3 for m in mots:
4     print(m, len(m)) # len est une fonction
```

avec un str

```
1 # Boucle for avec une chaîne de caractère:
2 mot = 'Carlos'
3 for lettre in mot:
4     print(lettre)
```

avec range

```
1 for i in range(10):
2     print(i)
```

nb de fois
que la boucle
va se répéter