



### VARIABLES

#### Modifier une valeur

- Incrémenter :
 

```
1 a=a+2
```
- Concaténer :
 

```
2 c=c+"python"
```

#### Affecter une valeur

```
1 a=5 ← type entier
2 b=5.1 ← type flottant
3 c="code" ← type chaîne
4 d=True ← type booléen
```

#### Étendre une liste de valeurs

```
>>> L.append(12)
>>> L
[4,6,8,10,12]
```

#### Modifier une liste de valeurs

```
>>> L=[4,6,8,10]
>>> L[0]=1
>>> L[len(L)-1]=9
>>> L
[1,6,8,9]
```

#### Affecter une liste de valeurs

- Liste en extension :
 

```
1 L=[4,6,8,10]
```
- Liste en compréhension :
 

```
1 L=[2*i for i in range(2,6)]
```

### Opérateurs booléens

```
1 condition1 or condition2
2 condition1 and condition2
3 # L'opposé du booléen condition1 :
4 not condition1
```

### Expressions booléennes

```
1 # x est égal à y ?
2 x==y
3 # x est inférieur ou égal à y ?
4 x<=y
5 # x appartient à la liste L ?
6 x in L
```

### Instruction conditionnelle

```
1 if condition:
2     ...
3 else:
4     ...
```

### TESTS

### FONCTIONS

#### Structurer un programme

```
1 # x et y sont les paramètres de la fonction
2 def ma_fonction(x,y):
3     ...
4     return ...
```

STOP et renvoi d'une valeur.

#### Appeler dans la console

```
>>> ma_fonction(3,10)
3.605551275463989
```

### MODULES

#### Fonctions mathématiques

```
1 import math
2 math.cos(x)
3 math.sin(x)
4 math.log(x)
5 math.exp(x)
6 math.sqrt(x)
7 math.abs(x)
```

#### Génération aléatoire

```
1 import random as rd
2 # Entier entre a et b inclus :
3 rd.randint(a,b)
4 # Flottant dans [0 ; 1[ :
5 rd.random()
```

Création d'un alias pour raccourcir les appels au module.

### BOUCLES

#### Boucle finie

```
1 for i in range(0,n):
2     ...
```

#### Parcourir une liste

```
1 for i in range(len(L)-1):
2     ...
```

ou

```
1 for e in L:
2     ...
```

#### Boucle conditionnelle

```
1 while condition:
2     ...
```

#### Représentations graphiques

• dans un repère :

```
1 import matplotlib.pyplot as plt
2 # Tracer un nuage de points :
3 plt.plot(Liste_x, Liste_y)
4 # Afficher la fenêtre de sortie :
5 plt.show()
```

• d'une série statistique :

```
1 import matplotlib.pyplot as plt
2 # Tracer un diagramme en barres :
3 plt.bar(Liste_valeurs, Liste_effectifs)
4 # Afficher la fenêtre de sortie :
5 plt.show()
```