

# Aide-mémoire – Langage Python

Opérateurs mathématiques : opérateurs usuels `+, -, *, /, //`, modulo `%`

Opérateurs booléens : comparaison `<, >, <=, >=`, égalité `==, !=`, combinaison `and, or, not`

---

```
if (Condition) :  
    ____{ Condition vraie }
```

```
if (Condition) :  
    ____{ Condition vraie }  
else :  
    ____{ Condition fausse }
```

```
while (Condition) :  
    ____{ Condition vraie }
```

```
for Paramètre in Range :  
    ____{ Bloc d'instructions }
```

```
def Fonction (Paramètres séparés par des virgules) :  
    ____{ Bloc d'instructions }
```

Exemple :

```
if age < age_min :  
    ____print("Moins de", age_min, "ans")  
else :  
    ____print("Pas de problème")
```

Exemple :

```
i = 0  
while i < nbSommets(G) :  
    ____s = sommetNumero(G,i)  
    ____print(s, "est un sommet de", G)  
    ____i = i + 1
```

Exemple :

```
for a in range(10) :  
    ____print(a, "à la puissance", a*a)
```

Exemple :

```
def odd_even(n) :  
    ____if (n % 2) == 0 :  
        ____return "pair"  
    ____else :  
        ____return "impair"
```

---

Les caractères “\_\_\_\_” représentent des tabulations obligatoires.

La fonction `range` permet de générer des suites d'entiers utilisables par la primitive `for` :

```
list(range(10))           [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
list(range(3, 7))        [3, 4, 5, 6]  
list(range(1, 20, 4))    [1, 5, 9, 13, 17]
```