

# Complément de cours

## Les variables booléennes

J1 MI 1003, groupe B3, Université Bordeaux

Un **booléen** est une variable qui ne peut prendre que deux valeurs : *vrai* ou *faux*. En Python, ces deux valeurs sont notées **True** et **False**.

### Structure de contrôle

Les tests, comme l'expression `a==10`, renvoient des booléens. Dans ce cas précis, `a==10` renvoie **True** si la valeur de `a` est égale à 10, **False** sinon.

Voici quelques exemples importants :

- `a == b` renvoie **True** si et seulement si les valeurs de `a` et `b` sont égales.
- `a != b` renvoie **True** si et seulement si les valeurs de `a` et `b` sont différentes.
- `a > b` renvoie **True** si et seulement si `a` est strictement plus grand que `b`.
- `a >= b` renvoie **True** si et seulement si `a` est plus grand ou égal que `b`.
- `a < b` renvoie **True** si et seulement si `a` est strictement plus petit que `b`.
- `a <= b` renvoie **True** si et seulement si `a` est plus petit ou égal que `b`.

**Attention** : Il ne faut pas confondre l'opérateur relationnel `==` avec l'opérateur d'affectation `=`. C'est une erreur très commune qui peut causer une perte de temps considérable durant la mise au point.

**Exemple de code** :

```
x = 3
y = 4
a = (x != y)
y = 3
b = (x != y)
c = (x == y)
```

Ainsi après exécution de ces quelques lignes de code, `a` et `c` vaudront **True** et `b` **False**.

### Opérateurs logiques

Lorsque nous avons des expression booléennes, il est utile de les comparer avec l'aide d'opérateurs logiques. Python définit 3 opérateurs logiques :

Opérateurs logiques		
<code>a and b</code>	ET logique booléen	Si <code>a</code> vaut <b>False</b> , renvoie <b>False</b> , sinon renvoie la valeur de <code>b</code> .
<code>a or b</code>	OU logique booléen	Si <code>a</code> vaut <b>True</b> , renvoie <b>True</b> , sinon renvoie la valeur de <code>b</code> .
<code>not a</code>	NON logique	Si <code>a</code> vaut <b>True</b> , renvoie <b>False</b> , sinon <b>True</b> .

**Exemple** :

```
(8 < 9) and (2 < 1)  → True and False  → False
(8 < 9) or (2 < 1)   → True or False   → True
not (2<1)            → not False     → True
```