

Les opérateurs logiques (ou, non, et, etc...) qui apparaissent en mathématiques sont omniprésents en informatique. Ils permettent de construire des formules logiques lisibles par les composants électroniques au cœur du fonctionnement des ordinateurs.

On doit à George Boole une branche des mathématiques qui est à l'origine du mode de fonctionnement des circuits intégrés présents dans tous les systèmes automatisés.

Les booléens sont des objets qui peuvent avoir deux valeurs :

- Vrai, représenté par True en python
- Faux représenté par False en python

Une fonction booléenne est une fonction qui, à une ou plusieurs variables booléenne associe un booléen. Pour définir une fonction booléenne à deux variables ou plus, il est plus pratique d'utiliser un tableau. Le tableau de valeurs exhaustif d'une fonction booléenne  $\varphi$  est appelé **table de vérité** de  $\varphi$ .

Les booléens sont un type de variable couramment utilisés en programmation, lors de branchements conditionnels.

Exemple :  $4 > 5$  a pour réponse False ;  $6 > 5$  donne True.

Les fonctions booléennes peuvent s'exprimer à l'aide des briques élémentaires suivantes :

- La fonction **non**  
(not en python)

L'inverse de la proposition

not False = True

not True = False

- La fonction **ou**  
(or en python)

« ou inclusif » - il suffit que x OU y soit vrai pour que la proposition soit vraie (l'un ou l'autre)

| x     | y     | x or y |
|-------|-------|--------|
| False | True  | True   |
| False | False | False  |
| True  | True  | True   |
| True  | False | True   |

- La fonction **et**  
(and en python)

x et y doivent être vrais pour que la proposition soit vraie

| x     | y     | x and y |
|-------|-------|---------|
| False | True  | False   |
| False | False | False   |
| True  | True  | True    |
| True  | False | False   |

Notons que les fonctions **non** et **ou** suffisent à exprimer toutes les fonctions logiques : elles forment un **système complet de connecteurs**.

Les différents tests de comparaison en python sont les suivants :

|              |    |                       |    |                       |    |
|--------------|----|-----------------------|----|-----------------------|----|
| Egal à       | == | Strictement supérieur | >  | Strictement inférieur | <  |
| Différent de | != | Supérieur ou égal     | >= | Inférieur ou égal     | <= |

### Vérifions que nous avons compris les bases : Exercice flash !

1. Compléter les tables de vérité ci-dessous pour montrer que l'on peut remplacer la fonction **et** par la combinaison suivante des fonctions **non** et **ou** (et donc prouvant que non et ou forment un système complet de connecteurs) :

$$x \text{ and } y = \text{not}(\text{not } x) \text{ or } (\text{not } y))$$

| x     | y     | x and y | not x | not y | (not x) or (not y) | not ((not x) or (not y)) |
|-------|-------|---------|-------|-------|--------------------|--------------------------|
| False | True  |         |       |       |                    |                          |
| False | False |         |       |       |                    |                          |
| True  | True  |         |       |       |                    |                          |
| True  | False |         |       |       |                    |                          |

2. Construire la table de vérité de  $((\text{not } x) \text{ and } y) \text{ or } (x \text{ and } (\text{not } y))$

Comment pourrait-on traduire cette fonction booléenne en français ?

### Pour aller un peu plus loin...

Construire la table de vérité de  $(\text{not } x) \text{ or } y$  et de  $((\text{not } x) \text{ or } y) \text{ and } ((\text{not } y) \text{ or } x)$

**Ce qu'il faut retenir**

❖ **Booléens et fonctions booléennes**

Les booléens sont des objets qui peut avoir deux valeurs :

- Vrai, représenté par True en python
- Faux représenté par False en python

Une fonction booléenne est une fonction qui, à une ou plusieurs variables booléenne associe un booléen. Pour définir une fonction booléenne à deux variables ou plus, il est plus pratique d'utiliser un tableau. Le tableau de valeurs exhaustif d'une fonction booléenne  $\phi$  est appelé **table de vérité** de  $\phi$ .

Les fonctions booléennes peuvent s'exprimer à l'aide des briques élémentaires suivantes :

• La fonction **non**  
(not en python)

• La fonction **ou**  
(or en python)

• La fonction **et**  
(and en python)

L'inverse de la proposition

not False = True

not True = False

« ou inclusif » - il suffit que x OU y soit vrai pour que la proposition soit vraie (l'un ou l'autre)

| x     | y     | x or y |
|-------|-------|--------|
| False | True  | True   |
| False | False | False  |
| True  | True  | True   |
| True  | False | True   |

x et y doivent être vrais pour que la proposition soit vraie

| x     | y     | x and y |
|-------|-------|---------|
| False | True  | False   |
| False | False | False   |
| True  | True  | True    |
| True  | False | False   |

Les fonctions **non** et **ou** suffisent à exprimer toutes les fonctions logiques : elles forment un **système complet de connecteurs**.

**Savoir faire**

❖ **Construire une table de vérité**