

1 Un peu d'histoire

George Boole, né le 2 novembre 1815 à Lincoln (Royaume-Uni) et mort le 8 décembre 1864 à Ballintemple (Irlande), est un logicien, mathématicien et philosophe britannique. Il est le créateur de la logique moderne, père de *l'algèbre de Boole*.

En effet, de 1844 à 1854, il crée une algèbre binaire, dite booléenne, n'acceptant que deux valeurs numériques : 0 et 1. Cette algèbre aura de nombreuses applications en téléphonie et en informatique, notamment grâce à Claude Shannon en 1938.

Claude Elwood Shannon (30 avril 1916 à Petoskey, Michigan - 24 février 2001 à Medford, Massachusetts) est un ingénieur en génie électrique et mathématicien américain. Il est l'un des pères, si ce n'est le père fondateur, de la théorie de l'information.

2 Les variables booléennes

Définition 2.1

- ★ Les *valeurs booléennes* sont les valeurs 0 et 1 correspondant à l'état d'un bit.
- ★ Une *variable booléenne* est donc une variable ne pouvant prendre que l'une ou l'autre de ces deux valeurs.

En Python les données de type `bool` sont les données qui ne peuvent prendre qu'une des deux valeurs suivantes : `True` (vrai pour 1) ou `False` (faux pour 0).

Exemple 2.2 L'instruction `2 == 3` renvoie `False`. L'instruction `1 in [2, 5, 1, 3.2]` renvoie `True`

3 Les opérateurs booléens

3.1 L'opérateur non (`not`)

Définition 3.1 L'opérateur `non` transforme 1 en 0 et 0 en 1, autrement dit :

$$\text{non}(1) = 0 \quad \text{et} \quad \text{non}(0) = 1$$

Exercice 3.2 Compléter la table de vérité suivante dans laquelle `a` désigne une variable booléenne :

a	non (a)
1	
0	

Remarque 3.3 L'expression `non(a)` où `a` désigne une variable booléenne, réalise la négation de `a`. Ainsi si `a` vaut 1 (est vraie) alors `non(a)` vaut 0 (est fausse) et si `a` vaut 0 (est fausse) alors `non(a)` vaut 1 (est vraie).

En Python l'opérateur est `not`. Attention, on peut évidemment l'appliquer aux valeurs booléennes de Python (`True` et `False`) mais également à n'importe quel objet : il renvoie `True` uniquement si la valeur passée en argument est `False`, `None`, une liste ou collection vide ou 0.

Exemple 3.4 Soit `a` une variable de type `int` (un nombre entier). L'instruction `not (a%2)` renvoie `True` si le nombre est pair (le reste de la division euclidienne par 2 vaut 0) et `False` sinon.

3.2 L'opérateur ou (`or`)

Définition 3.5 Soient `a` et `b` deux variables booléennes.

L'opérateur `ou` appliqué à `a` et `b` renvoie 1 si au moins l'une des deux variables vaut 1.

a	b	a ou b
0	0	
0	1	
1	0	
1	1	

Exercice 3.6 Compléter la table de vérité suivante :

En Python l'opérateur est `or`.

Exemple 3.7 Soit `a` une variable de type numérique. L'instruction `a < 6 or a%2` renvoie `True` si `a` est strictement inférieur à 6 ou bien si `a` est impair ou bien si `a` est impair et strictement inférieur à 6.

3.3 L'opérateur et (and)

Définition 3.8 Soient a et b deux variables booléennes.

L'opérateur `et` appliqué à a et b renvoie 1 si les deux variables valent 1, et 0 sinon.

a	b	a et b
0	0	
0	1	
1	0	
1	1	

Exercice 3.9 Compléter la table de vérité suivante :

En Python l'opérateur est `and`.

Exemple 3.10 Soit a une variable de type numérique. L'instruction `a < 6 and a%2` renvoie `True` uniquement si a est strictement inférieur à 6 et impair.

3.4 Le ou exclusif (xor)

Définition 3.11 Soient a et b deux variables booléennes.

L'opérateur `xor` appliqué à a et b renvoie 1 si et seulement si exactement une seule des deux variables booléennes vaut 1.

a	b	a xor b
0	0	
0	1	
1	0	
1	1	

Exercice 3.12 Compléter la table de vérité suivante :

En Python l'opérateur est `^`.

Remarque 3.13 L'opérateur `xor` renvoie 1 uniquement si les deux variables booléennes a et b ne sont pas égales.

En Python l'instruction `not (a==b)` ou encore `a!=b` est donc équivalente à l'instruction `a^b`.

4 Expressions booléennes

Définition 4.1 Une *expression booléenne* est une expression qui a pour valeur 0 ou 1.

Remarque 4.2 Une expression booléenne est une expression construite à l'aide de variables booléennes et des opérateurs booléens.

Exemple 4.3 L'expression $(a \text{ et } b) \text{ ou } \text{non}(c)$, où a, b et c sont des variables booléennes, est une expression booléenne.

Exercice 4.4 Compléter la table de vérité suivante de l'expression booléenne $(a \text{ et } b) \text{ ou } \text{non}(c)$:

a	b	c	a et b	non(c)	(a et b) ou non(c)
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Remarque 4.5 (Séquentialité des opérateurs et et ou)

- ★ Dans une expression $a \text{ et } b$, a est évaluée en premier. Si a est faux, l'expression prend la valeur de a , sinon la valeur de b .
- ★ Dans une expression $a \text{ ou } b$, a est évaluée en premier. Si a est vrai, l'expression prend la valeur de a , sinon la valeur de b .