Booléens et fonctions logiques

De la logique aux circuits

Résumé

La logique est une forme de raisonnement qui permet de déterminer si une affirmation est vraie ou fausse. En informatique, cette logique repose sur l'**algèbre de Boole**, où les variables ne peuvent prendre que deux valeurs : **vrai** (1) ou **faux** (0). Dans ce cours, vous découvrirez les opérateurs logiques fondamentaux, leurs propriétés et leurs applications.

① Introduction : la logique au quotidien

Considérons la phrase suivante :

« *Pour obtenir le permis de conduire, il faut réussir l'examen du code* ***et*** *l'examen de conduite.* »

Le permis dépend de deux conditions qui doivent être **simultanément** vraies.

Exercice 1 *Première table de vérité* Complétez la table ci-dessous qui représente toutes les situations possibles :

Code réussi	Conduite réussie	Permis obtenu
Non	Non	
Non	Oui	
Oui	Non	
Oui	Oui	

Exercice 2 *Une autre situation* Considérons maintenant :

« *On peut entrer dans la salle de cinéma avec un billet* ***ou*** *avec une carte d'abonnement.* »

1. Quelle est la différence avec la situation précédente ?

2. Complétez la table de vérité correspondante :

Billet	Abonnement	Entrée
Non	Non	
Non	Oui	
Oui	Non	
Oui	Oui	

② Les booléens

Définition 1 **(Booléen)** Un **booléen** est une valeur qui ne peut être que l'une de deux possibilités :

— **Vrai** (noté 1, ou `True` en Python) ;

— **Faux** (noté 0, ou `False` en Python).

Ce nom vient du mathématicien **George Boole** (1815–1864) qui a formalisé les règles du raisonnement logique.

Définition 2 **(Fonction logique)** Une **fonction logique** (ou fonction booléenne) est une fonction qui :

— prend en entrée un ou plusieurs booléens ;

— renvoie un booléen en sortie.

Définition 3 **(Table de vérité)** Une **table de vérité** est un tableau qui montre la valeur de sortie d'une fonction logique pour toutes les combinaisons possibles des entrées.

Exercice 3 *Nombre de lignes d'une table de vérité*

1. Combien de lignes comporte une table de vérité pour une fonction à 1 entrée ?

2. Combien de lignes pour 2 entrées ?

3. Combien de lignes pour 3 entrées ?

4. Pour une fonction à n entrées, combien y a-t-il de lignes ? Justifiez.

③ Les trois opérateurs fondamentaux

Toute fonction logique peut être construite à partir de trois opérateurs de base : **NON**, **ET** et **OU**.

3.1 L'opérateur NON (négation)

L'opérateur **NON** inverse la valeur d'un booléen : il transforme vrai en faux et faux en vrai.

Notation	Mathématique	Python
NON a	$\overline{a}$	`not a`

Exercice 4 *Table de vérité du NON* Complétez la table de vérité de l'opérateur NON :

a	$\overline{a}$
0	
1	

3.2 L'opérateur ET (conjonction)

L'opérateur **ET** renvoie vrai si et seulement si **les deux** entrées sont vraies.

Notation	**Mathématique**	**Python**
a **ET** b	$a \cdot b$ ou $a \wedge b$	`a and b`

Exercice 5 *Table de vérité du ET* Complétez la table de vérité de l'opérateur ET :

a	b	$a \cdot b$
0	0	
0	1	
1	0	
1	1	

Remarque : le résultat est identique à celui d'une multiplication. Expliquez pourquoi.

3.3 L'opérateur OU (disjonction)

L'opérateur **OU** renvoie vrai si **au moins une** des entrées est vraie.

Notation	**Mathématique**	**Python**
a **OU** b	$a + b$ ou $a \vee b$	`a or b`

Exercice 6 *Table de vérité du OU* Complétez la table de vérité de l'opérateur OU :

a	b	$a + b$
0	0	
0	1	
1	0	
1	1	

Ce OU est dit **inclusif** : quand a et b sont tous les deux vrais, $a + b$ est vrai. C'est le « ou » de la phrase « Je viendrai qu'il pleuve ou qu'il neige ».

④ Propriétés des opérateurs

Exercice 7 *Découverte des propriétés* Pour chaque propriété, vérifiez-la en calculant les deux membres pour toutes les valeurs possibles de a.

1. $\overline{\overline{a}} = a$ (double négation)

a	$\overline{a}$	$\overline{\overline{a}}$
0		
1		

2. $a \cdot 1 = a$ (élément neutre du ET)

3. $a \cdot 0 = 0$ (élément absorbant du ET)

4. $a + 0 = a$ (élément neutre du OU)

5. $a + 1 = 1$ (élément absorbant du OU)

6. $a \cdot \overline{a} = 0$ (complémentarité)

7. $a + \overline{a} = 1$ (tiers exclu)

⑤ Propriétés algébriques

Les opérateurs ET et OU obéissent à des règles similaires à celles de l'arithmétique classique, mais avec quelques différences importantes.

Propriétés Soient a, b et c des booléens.

Propriété	Pour le ET $(\cdot)$	Pour le OU $(+)$
Commutativité	$a \cdot b = b \cdot a$	$a + b = b + a$
Associativité	$(a \cdot b) \cdot c = a \cdot (b \cdot c)$	$(a + b) + c = a + (b + c)$
Idempotence	$a \cdot a = a$	$a + a = a$
Distributivité	$a \cdot (b + c) = a \cdot b + a \cdot c$	$a + (b \cdot c) = (a + b) \cdot (a + c)$

Remarque 1. La seconde distributivité $(a + (b \cdot c) = (a + b) \cdot (a + c))$ n'a pas d'équivalent en arithmétique classique. C'est une particularité de l'algèbre de Boole.

Exercice 8 *Vérification de l'idempotence* Vérifiez que $a + a = a$ et $a \cdot a = a$ pour toutes les valeurs de a.

Exercice 9 *Vérification de la distributivité du OU* Vérifiez par une table de vérité que $a + (b \cdot c) = (a + b) \cdot (a + c)$.

a	b	c	$b \cdot c$	$a + (b \cdot c)$	$(a + b)$	$(a + c)$	$(a + b) \cdot (a + c)$
0	0	0					
0	0	1					
0	1	0					
0	1	1					
1	0	0					
1	0	1					
1	1	0					
1	1	1					

Exercice 10 *Lois d'absorption* Démontrez algébriquement les lois d'absorption :

1. $a + a \cdot b = a$
2. $a \cdot (a + b) = a$

6 Construire des expressions logiques

Exercice 11 *Le OU exclusif (XOR)* Soit l'expression $\mathrm{E} = \overline{a} \cdot b + a \cdot \overline{b}$.

Complétez la table de vérité en calculant chaque colonne intermédiaire :

a	b	$\overline{a}$	$\overline{b}$	$\overline{a} \cdot b$	$a \cdot \overline{b}$	E
0	0					
0	1					
1	0					
1	1					

Cette fonction s'appelle le **OU exclusif** (XOR), noté $a \oplus b$. En Python, le OU exclusif de deux booléens s'écrit `a ^ b`. Dans quel cas renvoie-t-elle vrai ?

Exercice 12 *Les lois de De Morgan* Augustus De Morgan (1806–1871) a établi deux lois fondamentales.

Première loi : $\overline{a + b} = \overline{a} \cdot \overline{b}$

Vérifiez cette loi en complétant la table :

a	b	$a + b$	$\overline{a + b}$	$\overline{a}$	$\overline{b}$	$\overline{a} \cdot \overline{b}$
0	0					
0	1					
1	0					
1	1					

Seconde loi : $\overline{a \cdot b} = \overline{a} + \overline{b}$

Vérifiez cette loi :

a	b	$a \cdot b$	$\overline{a \cdot b}$	$\overline{a}$	$\overline{b}$	$\overline{a} + \overline{b}$
0	0					
0	1					
1	0					
1	1					

7 Autres fonctions logiques

En combinant les trois opérateurs de base, on peut créer d'autres fonctions utiles.

Exercice 13 *NAND* Complétez les tables de vérité des fonctions suivantes :

1. **NAND** (NON-ET) : $a \uparrow b = \overline{a \cdot b}$

a	b	$a \uparrow b$
0	0	
0	1	
1	0	
1	1	

2. **NOR** (NON-OU) : $a \downarrow b = \overline{a + b}$

a	b	$a \downarrow b$
0	0	
0	1	
1	0	
1	1	

3. **XNOR** (NON-OU exclusif) : $\overline{a \oplus b}$

a	b	$\overline{a \oplus b}$
0	0	
0	1	
1	0	
1	1	

Dans quel cas la fonction XNOR renvoie-t-elle vrai ?

⑧ Universalité de la fonction NAND

Exercice 14 *Exprimer NON avec NAND* La fonction de Sheffer (NAND) est définie par : $S(a, b) = \overline{a \cdot b}$.

Montrez que $\overline{a} = S(a, a)$ en complétant la table :

a	$a \cdot a$	$\overline{a \cdot a} = S(a, a)$
0		
1		

Comparez avec la table de $\overline{a}$. Conclusion ?

Exercice 15 *Exprimer ET et OU avec NAND*

1. Sachant que $\overline{a} = S(a, a)$, montrez que $a \cdot b = S(S(a, b), S(a, b))$.

   Indication : $S(S(a, b), S(a, b)) = \overline{S(a, b)} = \overline{\overline{a \cdot b}}$

2. En utilisant la loi de De Morgan, montrez que $a + b = S(S(a, a), S(b, b))$.

9 Applications

Exercice 16 *Le multiplexeur* Un **multiplexeur** est un circuit qui sélectionne une entrée parmi plusieurs selon la valeur d'un signal de commande.

On définit : $\text{mux}(x, y, z) = (\overline{x} \cdot y) + (x \cdot z)$

où x est le signal de sélection, y et z sont les deux entrées.

Complétez la table de vérité :

x	y	z	$\overline{x}$	$\overline{x} \cdot y$	$x \cdot z$	$\textbf{mux}(x, y, z)$
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

Quand $x = 0$, quelle entrée est sélectionnée ? Et quand $x = 1$?

Exercice 17 *Problème logique* Med est content si :

— Bob **et** Jon sont tous les deux là, mais **sans** Rut ;

— **ou** si Rut est là avec Bob (mais pas Jon) ;

— **ou** si Rut est là avec Jon (mais pas Bob).

On note R, B et J les booléens indiquant la présence de Rut, Bob et Jon.

1. Construisez la table de vérité avec en sortie « Med est content » :

R	B	J	**Content**
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

2. Écrivez l'expression logique correspondant au choix de Med.

3. Observez la table : peut-on exprimer le choix de Med par une phrase plus simple ?

10 Simplification d'expressions logiques

Exercice 18 ***Utiliser les propriétés*** Simplifiez les expressions suivantes en utilisant les propriétés de l'algèbre de Boole. Justifiez chaque étape.

1. A = $a + a \cdot b$

2. B = $a \cdot (a + b)$

3. C = $a + \overline{a} \cdot b$

   Indication : utilisez la distributivité $a + (b \cdot c) = (a + b) \cdot (a + c)$

4. D = $(a + b) \cdot (a + \overline{b})$

Synthèse

Exercice 19 *Récapitulatif* Complétez le tableau récapitulatif des opérateurs logiques :

Opérateur	Notation	Description	Python
NON			
ET			
OU			
NAND			
NOR			
XOR			