LYCÉE BOISSY D'ANGLAS
LYCÉE DES MÉTIERS DE LA CRÉATION INDUSTRIELLE

Représentation des données : types et valeurs de base

Les booléens

Contenu : - Valeurs booléennes : 0, 1.
-Opérateurs booléens : and, or, not.
Expressions booléennes
Capacités attendues : Dresser la table d'une expression booléenne.

NSI - 1ère
COURS

1. Qu'est-ce qu'un booléen ?

C'est une variable qui ne peut prendre que deux valeurs : _____ou _____

George Boole (1815-1864) est un logicien, mathématicien et philosophe britannique. Il est le créateur de la logique moderne, fondée sur une structure algébrique et sémantique, que l'on appelle aujourd'hui algèbre de Boole.

On associe 0 à Faux et 1 à Vrai.
En Python: deux valeurs de type booléen : _____et _____.

2. Opérateurs de comparaisons

Prenons l'exemple suivant : x = 23 et y = 15

Les opérateurs de comparaisons usuels

Opérateur	Expression	Signification	Valeur
	x==y	Égal	0 (Faux)
	x != y	Non égal	1 (Vrai)
	X > y	Plus grand que	1
	X < y	Plus petit que	0
	X >= y	Plus grand ou égal	1
	X <= y	Plus petit ou égal	0

3. Les opérateurs booléens

Il existe des opérations spécifiques pour les booléens : _____, _____et _____que l'on appelle aussi __ _____, _____et le ___ logiques

3.1. Le NON logique

Le NON logique d'un booléen a se définit par :
_____vaut _____ si et seulement si __vaut _____
On peut définir cet opérateur par sa _____:

Table de vérité de l'opérateur NON

a	NON a

3.2. Le ET logique

Le ET logique entre deux booléens a et b se définit par :
_____vaut _____ si et seulement si _ vaut _____ et _ vaut _____
On peut définir cet opérateur par sa table de vérité :

Table de vérité de l'opérateur ET

a	b	a ET b

Application :

Dans le programme ci-dessous l'instruction ne sera exécutée que si les deux conditions sont vraies :

```
if condition1 and condition2 :
    instruction
```

3.3. Le OU logique

Le OU logique entre deux booléens a et b se définit par :
_____est _____ si et seulement si _ vaut _____ ou _ vaut _____
On peut définir cet opérateur par sa table de vérité :

Table de vérité de l'opérateur OU

a	b	a OU b

Application :
Dans le programme ci-dessous l'instruction ne sera exécutée que si l'une au moins des deux conditions est vraie

```
if condition1 or condition2 :
    instruction
```

3.4. Un peu d'algèbre...

Combinons un peu ces opérateurs :
Écrivons la table de vérité de : _____

a	b	NON a	NON b	(NON a) ET (NON b)

Rajoutons deux colonnes : _____ et _____

a	b	NON a	NON b	(NON a) ET (NON b)	NON((NON a) ET (NON b))	a OU b

On vient de montrer que : _____
Ce qui est équivalent à : _____
Il y a également un parallèle entre ces opérateurs et les opérateurs ensemblistes $\overline{a}$ (NON) , $a \cap b$ (ET) et $a \cup b$ (OU)
La formule que nous venons de montrer s'écrit alors : $a \cup b = \overline{a} \cap \overline{b}$ (1ère loi de Morgan).
Et comme $\overline{\overline{a}} = a$, on a : $a \cup b = \overline{\overline{a}} \cap \overline{b}$

3.5. L'opérateur XOR

_____ est _____ si et seulement si _ est _____et _ est _____si _ est _____ et _ est _____
C'est l'opérateur OU- Exclusif, dont la table de vérité est :
On montre que : _____

Table de vérité de l'opérateur XOR

a	b	a XOR b

4. Règles de priorité

- Les opérateurs booléens peuvent s'enchaîner.
- De la même façon que les opérateurs arithmétiques / et * sont évalués avant + ou - , il y a un ordre de priorité pour les opérateurs booléens.
- Voici l'ordre :
  - **not est évalué en premier.**
  - **and est évalué en deuxième.**
  - **or est évalué en dernier.**
- Il est possible de changer cet ordre en ajoutant des parenthèses ().

5. EXERCICES

5.1. Distributivité

Nous allons voir que les opérateurs ET et OU sont distributif l'un vis à vis de l'autre
Soient a, b et c trois booléens
Établir les tables de vérités de :

- a ET (b OU c)
- a ET b OU a ET c

En déduire la distributivité de ET par rapport à OU

5.2. Équations booléennes

Soient a, b et c trois booléens
Établir les tables de vérités de :

- a OU (b ET c)
- a OU b ET a OU c

En déduire la distributivité du OU par rapport au ET

5.3. La seconde loi de Morgan

On a vu la 1ère loi de Morgan : $\overline{a \cup b} = \overline{a} \cap \overline{b}$ (1ère loi de Morgan)
Vous allez montrer, à l'aide d'une table de vérité la seconde loi de Morgan :

$\overline{a \cap b} = \overline{a} \cup \overline{b}$ (2nde loi de Morgan)

Indice

Il faut remplir un tableau :

a	b	a ET b	NON(a ET b)	NON a	NON b	(NON a) OU (NON b)
...	...	...	...	...	...	...

5.4. Donner l'affichage final de :

```
A = not True
B = not 3**4 < 4**3
C = not 10 % 3 <= 10 % 2
D = not 3**2 + 4**2 != 5**2
E = not not False
print(A,B,C,D,E)
```

5.5. Donner l'affichage final de :

```
A =  False and False
B = -(-(-(-2))) == -2 and 4 >= 16**0.5
C = 19 % 4 != 300 / 10 / 10 and False
D = -(1**2) < 2**0 and 10 % 10 <= 20 - 10 * 2
E = True and True
print(A,B,C,D,E)
```

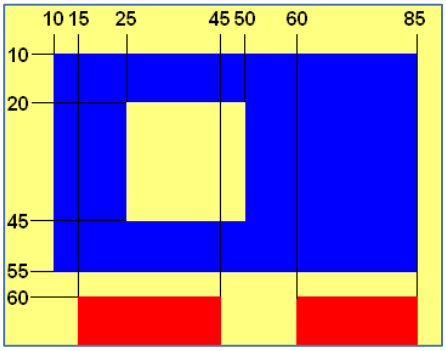
5.6. Donner l'affichage final de :

```
A =  2**3 == 108%100 or 'Bonjour' == 'Salut'
B = True or False
C = 100**0.5 >= 50 or False
D = True or True
E = 1**100 == 100**1 or 3 * 2 * 1 != 3 + 2 + 1
F = -(1**2) < 2**0 and 10 % 10 <= 20 - 10 * 2
print(A,B,C,D,E)
```

5.7. Donner l'affichage final de :

```
A = False or not True and True
B = False and not True or True
C = True and not (False or False)
D = not not True or False and not True
E = False or not (True and True)
print(A,B,C,D,E)
```

5.8. Jetons (Source : France IOI)



On place un jeton sur une table, à l'exception des frontières entre les différentes zones.
L'utilisateur renseigne dans un premier temps le nombre de jeton à jouer.

Puis il rentre les coordonnées de chaque jeton (coordonnées x et y). Le programme doit indiquer alors la couleur de la zone où a été placée ce jeton.

Écrire un programme de telle sorte qu'il y ait au maximum une instruction "si" par possibilité de texte affiché.

La zone de la table va de x = 0 à x = 90, et de aussi de y = 0 à y = 70.

5.9. Fonction mux(x, y, z)

On définit la fonction multiplexeur, notée **mux**, de B^3 dans B par : mux(x,y,z) = (non(x) et y) ou (x et z)
Compléter la table suivante :

Table : mux(x;y;z)

x	y	z	non(x)	non(x) et y	x et z	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				

5.10. Booléens et listes

Soit, liste1 = [3,5,7], liste2 = [10,3,18,25],
Et les tests suivants : *test1 = 3 in (liste1 and liste2), test2 =10 in (liste1 and liste2), test3 = 10 in (liste1 or liste2), test4 = 20 in (liste1 and liste2).*
Donner les valeurs des tests.