Logique

I. La logique Python

Les valeurs Python évaluées à False
0 : int
0.0 : float
None : nonetype
’ ’ : string
False : bool

Toutes les autres valeurs sont considérées comme des valeurs vraies

Exemple :

```
valeur = 0

if valeur:
    print('Fizz')
else:
    print('Buzz')
```

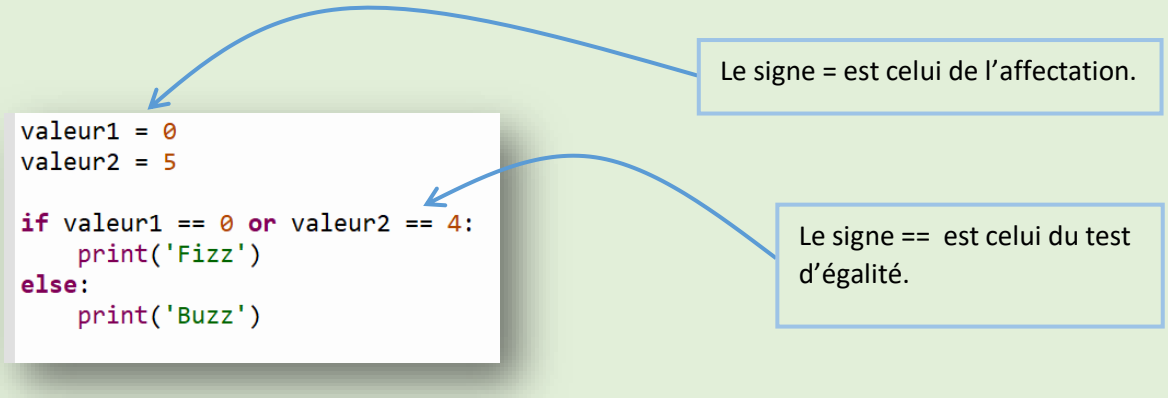
Valeur est évaluée à alse donc ce code affichera Buzz

II. Les connecteurs logiques. (Très important !!!)

a. Le OU(or)

a	b	a OU b
False	True	True
False	False	False
True	True	True
True	False	True

⚠ : En Python si la première valeur est évaluée à True alors la seconde valeur n'est pas évaluée



Valeur1 == 0	Valeur2 == 4	OU
True	False	True

C'est donc Fizz qui sera affiché.

b. Le ET(and)

a	b	a ET b
False	True	False
False	False	False
True	True	True
True	False	False

```
valeur1 = 0
valeur2 = 5

if valeur1 == 0 and valeur2 == 4:
    print('Fizz')
else:
    print('Buzz')
```

Valeur1 == 0	Valeur2 == 4	ET
True	False	False

C'est donc Buzz qui sera affiché.

c. Le NON(not)

a	NON a
True	False
False	True

```
valeur1 = 0
valeur2 = 5

if not valeur1 == 0:
    print('Fizz')
else:
    print('Buzz')
```

Valeur1 == 0	NON (not valeur1 == 0)
True	False

C'est donc Buzz qui sera affiché.

Parmi les quatre expressions suivantes, laquelle s'évalue en True ?

- False and (True and False) → False and False → False
- False or (True and False) → False or False → False
- True and (True and False) → True and False → False
- True or (True and False) → True or False → True

Si a vaut False et b vaut True, que vaut l'expression booléenne NOT(a AND b) ?

- 0
- False
- True
- None

La variable x contient la valeur 3, la variable y contient la variable 4. Quelle expression s'évalue en True parmi les quatre propositions suivantes ?

- x == 3 or y == 5 → True
- x == 3 and y == 5 → False
- x != 3 or y == 5 → False
- y < 4 → False

== Vérifie l'égalité

!= Vérifie l'inégalité

Quand on ajoute deux bits a et b, on obtient un bit de somme s et un bit de retenue r. On peut exprimer s et r à l'aide de formules logiques, lesquelles ?

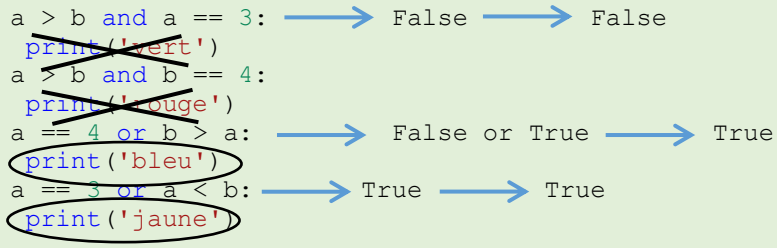
- r=a ET b ; s=a OU b
- r=a ET b ; s=a ET b
- r=a ET b ; s=a OU $EXCLUSIF$
- r=a OU $EXCLUSIF$ b ; s=a ET b

Sachant que l'expression not(a or b) a la valeur True, quelles peuvent être les valeurs des variables booléennes a et b ?

- True et True
- False et True
- True et False
- False et False

Qu'affiche le programme suivant :

```
a = 3
b = 4
if a > b and a == 3:
    print('vert')
if a > b and b == 4:
    print('rouge')
if a == 4 or b > a:
    print('bleu')
if a == 3 or a < b:
    print('jaune')
```



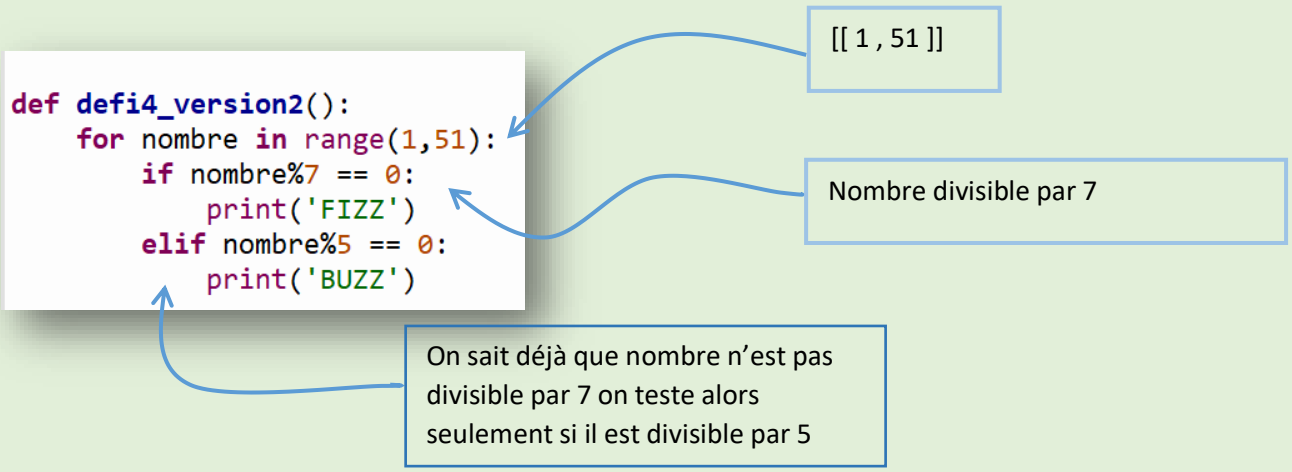
- vert rouge
- bleu jaune
- bleu
- vert jaune

On considère une formule booléenne form des variables booléennes a et b dont voici la table de vérité.

a	b	form
True	True	False
False	True	False
True	False	True
False	False	False

Quelle est cette formule booléenne form ?

- a and b
- a or b
- a and not(b)
- not(a) or b


def defi4_version2():
for nombre in range(1,51):
if nombre%7 == 0:
print('FIZZ')
elif nombre%5 == 0:
print('BUZZ')
[[1 , 51]]
Nombre divisible par 7
On sait déjà que nombre n'est pas divisible par 7 on teste alors seulement si il est divisible par 5