

Le but de ce TP est de vous familiariser encore plus avec les expressions booléennes en Python.

Exercice 1: Tableaux de vérité

Pour chaque combinaison possible des variables P, Q, et K écrivez dans le tableau ci-dessous les valeurs des expressions A, B, C et D.

A = not P or Q and K

B = not P or not Q and (P or K)

C = (P or Q) and not K or P and Q

D = (P and Q or P and not Q) or P and Q and K or not P and not Q and K

Aidez-vous en ajoutant des parenthèses aux expressions pour mettre en évidence l'ordre d'exécution des opérateurs et en les simplifiant si c'est possible.

A = _____
 B = _____
 C = _____
 D = _____

Tableau de vérité:

#	P	Q	K	A	B	C	D
1	0	0	0				
2	0	0	1				
3	0	1	0				
4	0	1	1				
5	1	0	0				
6	1	0	1				
7	1	1	0				
8	1	1	1				

Exercice 2: Priorités des opérateurs

Rappel: Les opérateurs dans une expression sont évalués selon les priorités mentionnées dans le tableau ci-dessous (les opérateurs en haut sont exécutés avant ceux en bas). Les opérateurs avec la même priorité sont exécutés de gauche à droite (ici on parle de gauche à droite dans le code, pas des lignes du tableau ci-dessous).

#	Opérateurs	Remarques
1	x[index], x[index1:index2]	Suscription
2	**	Puissance
3	-x	Négation
4	*, /, %	Multiplication, division, modulo
5	+, -	Addition et soustraction
6	<, <=, >, >=, !=, ==, in	Comparaisons et inclusion *
7	not x	Booléen NOT
8	and	Booléen AND
9	or	Booléen OR

* L'opérateur `in` vérifie si une sous-chaine se trouve dans une autre, par exemple:

```
>>> "b" in "abc"
True
>>> "bc" in "abc"
True
>>> "ac" in "abc"
False
```

Ajoutez des parenthèses dans les expressions suivantes et calculez la valeur de l'expression, en supposant que `i`, `j`, `k` et `word` sont initialisés de la manière suivante:

```
i = 2; j = 4; k = 10; word = "bonjour"
```

#	Expression	Résultat
1	<code>2 + 2 * 2 == 8 and True and i ** j / k * 2 ** 2</code>	
2	<code>5 ** 2 / 3 == 5 ** 2 / 3.0</code>	
3	<code>- 5 ** 2.0 / 3 == 5 ** 2 / 3.0</code>	
4	<code>False or False and True or True</code>	

#	Expression	Résultat
5	<code>False != 1 or 10 / 2 < 10</code>	
6	<code>k < j or i + 2 == k - 2</code>	
7	<code>i != 2 and j / i ** 0 < k</code>	
8	<code>j / (i - 2) < k and i != 0</code>	
9	<code>k <= j or i+2 == k-1</code>	
10	<code>2 + i == j + 2 or i < j != True and False == not True</code>	
11	<code>2 + i == j + 2 or i < j != True and not False == True</code>	
12	<code>k > j > i</code>	
13	<code>j in word</code>	
14	<code>"j" in word and "" in word</code>	
15	<code>word [5:4] * 3 == "" and word [0] + word [1] in word</code>	
16	<code>str(j) in word * k + str(j) and word [2] > word [1]</code>	

Exercice 3: Random

Soit la librairie `random` de Python (ne pas oublier l'instruction **`import random`** au début de votre script) qui contient les fonctions permettant de générer des nombres aléatoires. La fonction `random()` retourne un nombre décimal compris entre `[0, 1[` (zéro compris et 1 non-compris).

3.1. En vous aidant de l'interpréteur de commandes Python et des exemples suivants, plus ceux que vous imaginerez vous-même, expliquez en quelques mots ce que fait/retourne chacune des fonctions suivantes. Détaillez le type (entier, décimal, booléen, etc.) de ce qui est retourné, l'intervalle dans lequel il se trouve et la signification des divers paramètres:

```
print random.uniform(10, 20)
```

que fait `random.uniform(a, b)`:

```
print random.randint(100, 1000)
```

que fait `random.randint(a, b)`:

```
print random.randrange(100, 1000, 2)
```

que fait `random.randrange(a, b, c)`:

3.2. Déclarez une variable réelle `d` et affectez-lui une valeur aléatoire comprise entre -7 et $+4$.

3.3. Déclarez une variable entière `n` et affectez-lui une valeur entière aléatoire entre -7 et $+4$, bornes comprises.

3.4. Ecrivez un script qui demande la borne supérieure et inférieure à l'utilisateur et qui affichera:

Un nombre entier entre les bornes comprises;

Un nombre réel entre les bornes, mais la borne supérieure non comprise.

3.5. Modifiez votre script de l'exercice 3.4 pour qu'il affiche un nombre entier suivi par "franc" ou "francs" selon sa valeur.