TD2. Opérateurs booléens

Bases de la Programmation Impérative – Chapitre 1 : bases
version du 25 décembre 2023

L'objectif de ce TD est de se familiariser avec les opérateurs logiques manipulant des valeurs booléennes, c'est-à-dire `True` et `False` en python, et avec la structure de contrôle conditionnelle `if`.

Exercice 1 : opérateurs logiques

On considère le programme suivant :

```python
#!/usr/bin/env python3
"""Premier programme avec variables booléenes et opérateurs logiques."""

a = 10
b = 20
c = 30
d = a < b
print("d =", d)
e = b < c and a > b
print("e =", e)
d = d or e
print("d =", d)
f = (not d) or (a + b > c and e)
print("f =", f)
```

Question 1

Exécuter le programme en notant, comme dans le TD précédent, l'évolution des variables au cours de l'exécution. Noter également ce qu'affiche le programme sur la sortie standard.

Exercice 2 : opérateurs logiques toujours

On considère le programme suivant :

```python
#!/usr/bin/env python3
"""Qu'affiche ce programme ?"""


def est_pair(entier):
    """Affiche et renvoie la parité sous forme d'un booléen.

    Retourne True si entier est pair, False sinon."""
    parite = entier % 2 == 0  # % est l'opérateur "modulo" en Python
    print(entier, "est pair =", parite)
    return parite


a = 2 < 3 < 4 or est_pair(4)
b = est_pair(6) and est_pair(3)
print("a =", a, ", b =", b)
```

Question 1

Qu'affiche ce programme sur la sortie standard quand on l'exécute ?

Exercice 3 : réécritures

On considère le programme suivant :

```
#!/usr/bin/env python3

...
b = 7
if a == 3:
    b = 4
    e = a + 1
elif a == 5:
    b = 0
    e = a - 1
else:
    e = 4
```

Question 1

Comment simplifier ce code ?

Question 2

En sachant que `int(False)` vaut `0` et que `int(True)` vaut `1`, trouver une expression arithmétique calculant la valeur de `b` en fonction de `a`.

Question 3

On suppose maintenant que `a` est entier compris entre `0` et `6`. À l'aide d'un tuple de 7 éléments, supprimer toutes les comparaisons pour affecter la bonne valeur à `b`.

Question 4

Pour ceux qui connaissent déjà python, comment raccourcir le test suivant sachant que `s` est une chaîne de caractères, `i` un entier et `t` un `tuple` ?

```
if len(s) != 0 and i != 0 and len(t) == 0 :
    // do something
```

Exercice 4 : date correcte

Question 1

Écrire la fonction `date_correcte(jour, mois, annee)` qui renvoie `True` si les trois entiers donnés en argument forment une date correcte et `False` sinon. On considère pour cette première question que le mois de février a toujours 28 jours.

Question 2

Prendre en compte les années bissextiles. Une année est bissextile si elle rentre dans l'un des deux cas suivants :

- l'année est divisible par 4 et non divisible par 100 ;
- l'année est divisible par 400.

Par exemple, l'an 2000 était bissextile mais 2100 ne le sera pas.

Exercice 5 : surprenant non ? (pour aller plus loin)

Question 1

Qu'affiche le programme suivant ?

```
1  s = "toto"
2  i = 41
3  print(s or i)
```

Question 2

Qu'affiche le programme suivant ?

```
1  s = "toto"
2  i = 41
3  print(s and i)
```

Exercice 6 : fizzbuzz ? (pour aller plus loin)

Question 1

Écrire le plus "joliment" possible une fonction `fizzbuzz(nombre)` retournant :

- `"fizz"` si `nombre` est un multiple de 3
- `"buzz"` si `nombre` est un multiple de 5
- `"fizzbuzz"` si `nombre` est un multiple de 3 et de 5
- `str(nombre)` sinon