

Exercice 1. —

1. a. Taper dans la console les instructions suivantes :

```
50 + 3      ;      50 - 3      ;      50 * 3      ;      50 / 3
50 ** 3     ;      50 // 3     ;      50 % 3      ;      8e9
```

- b. Expliquer les instructions de la seconde ligne.

2. À quoi correspondent les instructions `0b11111100011` et `0x7E5` ?

Exercice 2. — Conversion implicite / conversion explicite

1. La fonction `type` permet de connaître le `type` d'un objet en Python.

Taper dans la console les instructions `56 / 7` puis `type(56)` et enfin `type(8.0)`.

Conjecturer le comportement de l'opérateur `/`.

2. La fonction `int` permet la conversion d'un `flottant` en `entier`.

Sans utiliser la console, conjecturer le résultat des instructions suivantes :

```
int(12.34) ; int(-273.15) ; int(22 / 7) ; int(-399e-2)
```

Exercice 3. — Être ou ne pas être

1. a. Taper dans la console les instructions suivantes :

```
2 > 8      ;      2 <= 8      ;      -4.10 < -4.9
1e3 >= 2**9 ;      6 != 6.0    ;      0.1 + 0.2 == 0.3
```

- b. À quoi correspondent les opérateurs `==` et `!=` ?

- c. Comment expliquer le résultat de l'instruction `0.1 + 0.2 == 0.3` et quelle leçon en tirer ?

2. Taper dans la console l'instruction `type(1e3 == 1000)`.

Le type `bool` est un type un peu particulier puisqu'il ne comporte que deux objets : `True` et `False`, qui représentent le vrai et le faux.

À ce type sont associés – en plus des opérateurs vus à la question 1.a. – les trois opérateurs `not`, `and` et `or`.

À l'aide de la console, donner les **tables de vérité** des opérateurs `not`, `and` et `or`.

Exercice 4. — Échange standard

Les données (calculées ou non) peuvent être mémorisées à l'aide de **variables**. Pour ce faire, il faut attribuer un **nom** à cette variable et lui associer une **valeur** à l'aide de l'opérateur `=`.

1. Dans la console, taper l'instruction `a = 5`.

Que renvoient les instructions ci-dessous ?

```
a      ;      3 * (a + 7)      ;      2*a**2 - 10*a
```

2. a. Dans l'éditeur, recopier les instructions suivantes.

```
a = 3
b = 8
print(a, b)
```

Exécuter ce script – à l'aide de la touche F5 du clavier (pour Pyzo) ou Ctrl+F9 (pour PYSRIPTER) – puis commenter ce qui s'est passé.

- b. Compléter ce script (c'est-à-dire cette suite d'instructions) de sorte que le contenu des variables `a` et `b` soit échangé indépendamment des valeurs contenues dans `a` et `b` initialement.

On pourra ajouter à la fin du script l'instruction `print(a, b)` pour vérifier que le script réponde bien au problème.

Exercice 5. — The final Countdown

1. Dans l'éditeur, taper les instructions :

```
for i in range(11):
    print(i)
```

Exécuter ce script puis commenter.

2. Modifier le script précédent de sorte que le compte s'effectue à rebours.

3. Dans l'éditeur, recopier et compléter le script suivant.

```
for nombre in range(1, 11):
    print(nombre, 'x 2 =', nombre*2)
```

Écrire un script qui affiche dans le shell les 20 premiers multiples de 17 (0 exclu).

4. Écrire un script qui affiche dans le shell les restes des 15 premiers multiples de 6 dans la division euclidienne par 7.

Exercice 6. — If 6 Was 9 ...

1. Taper dans l'éditeur les instructions suivantes.

```
age = int(input('Entrer votre âge : '))
if age >= 18:
    print('Vous êtes majeur(e).')
else:
    print('Vous êtes mineur(e).')
```

Exécuter ce script puis expliquer la première ligne.

- Écrire un script qui demande à l'utilisateur un entier et affiche dans le shell si ce dernier est divisible par 7 ou non.
- Écrire un script qui demande à l'utilisateur un entier et affiche dans le shell si ce dernier est strictement positif, nul ou strictement négatif.
- Écrire un script qui demande à l'utilisateur les coefficients (entiers) a , b et c de la fonction polynôme $f(x) = ax^2 + bx + c$ et qui affiche dans la console le nombre de racines de cette fonction (c'est-à-dire le nombre de solutions de l'équation $f(x) = 0$).
- Une année est *bissextile* si elle est :
 - divisible par 4 et non divisible par 100;
 - divisible par 400.

Écrire un script qui demande à l'utilisateur une année et qui affiche dans le shell si celle-ci est bissextile ou non.

Exercice 7. — Travail à la chaîne

- Dans la console, taper les instructions ci-dessous.

```
st1 = 'Bonjour'
type(st1)
st1[3]
len(st1)
```

Commenter les résultats obtenus par les deux dernières instructions.

- Que font les instructions `st1[2:]`; `st1[:4]`; `st1[2:4]` et `st1[1:5:3]`?
- Taper dans la console les instructions :

```
mot1 = 'NSI'
mot2 = 'ISN'
mot1 + mot2
4 * mot1
```

Commenter les résultats obtenus par les deux dernières instructions.

- PARCOURS D'UNE CHAÎNE. Les chaînes de caractères sont *itérables* : on peut les parcourir à l'aide d'une boucle `for`.

Dans l'éditeur, recopier et compléter le script suivant qui compte le nombre d'occurrences du caractère `a` dans la chaîne de caractères `phrase`.

```
phrase = "Voilà l'alpha et l'oméga."
compteur = 0
```

```
for lettre in phrase:
    if lettre == 'a':
        compteur = ...
print('Le caractère a est présent {} fois'.format(compteur))
```

Expliquer le résultat obtenu une fois le script complété et exécuté.

Exercice 8. — Leibniz

- Dans la console, taper les instructions :

```
from random import randint
randint(1, 3)
```

Que fait l'instruction `randint(1, 3)`?

On pourra soit exécuter plusieurs fois dans la console la fonction `randint` et conjecturer, soit utiliser la fonction prédéfinie `help` en tapant dans la console `help(nom_de_la_fonction)`.

- On lance un dé équilibré à 6 faces.

Écrire un script dans l'éditeur qui simule le lancer de dé et qui teste si la valeur du dé est 5.

- On s'intéresse maintenant à la somme de deux dés cubiques bien équilibrés.

Écrire un script qui simule 10000 lancers et qui affiche dans le shell le nombre d'occurrences du nombre 5 ainsi que sa fréquence d'apparition.

À l'aide de votre script, semble-t-il plus judicieux de parier sur le 5 ou sur le 8?