Python (2)

Opérateurs et chaînes de caractères

Exercice 1 *QCM – vérification des prérequis* Pour chaque question, une seule réponse est correcte.

1. Que vaut l'expression `7 // 2` en Python ?

— A. `3.5`

— B. `3`

— C. `1`

— D. `3.0`

2. Que vaut `2 ** 3` ?

— A. `6`

— B. `8`

— C. `5`

— D. Erreur

3. Quelle est la valeur de `10 % 3` ?

— A. `3`

— B. `3.33`

— C. `1`

— D. `0`

4. Que vaut l'expression `"bon" + "jour"` ?

— A. `"bon jour"`

— B. `"bonjour"`

— C. `"bonjourbonjour"`

— D. Erreur

Exercice 2 *Heures, minutes et secondes* Un chronomètre indique une durée de 3 725 secondes. On veut l'écrire en heures, minutes et secondes.

```
total = 3725

heures = total // 3600
reste = total % 3600
minutes = reste // 60
secondes = reste % 60
print(f"{heures} h {minutes} min {secondes} s")
```

1. Sans exécuter le programme, calculer à la main la valeur de chaque variable.
2. Qu'affiche le programme ? Vérifier dans Thonny.
3. Pourquoi utilise-t-on `//` pour calculer les heures, et `%` pour calculer le reste ?
4. Modifier le programme pour une durée de 10 000 secondes. Quel résultat obtient-on ?

Exercice 3 *Opérateurs* **3.1.** Essayez de deviner les résultats des opérations suivantes, puis vérifiez dans l'interpréteur : `3 / 2`; `3 // 2`; `3 % 2`; `3 ** 2`; `2e3`.

3.2. Calculez la moyenne de chacune des séries de nombres suivantes : 2 et 4 ; puis 4, 8 et 9 ; puis 12, `14/6` et 15.

3.3. Le volume d'une sphère est donné par la formule $V = \frac{4}{3}\pi r^3$. Calculer le volume d'une sphère de rayon 5. Indication : écrire `from math import pi`.

3.4. À l'aide de l'opérateur modulo `%`, testez la parité de chacun des nombres suivants : `1`; `5`; `20`; `60/7`.

3.5. Trouvez des valeurs `x` et `y` telles que l'expression `x < 1/3 < y` soit vraie (renvoie `True`).

Exercice 4 *Chaînes de caractères – manipulations de base* **4.1.** Écrire un programme qui demande à l'utilisateur de saisir une chaîne de caractères `c` et lui renvoie un message indiquant si la chaîne contient la lettre `'p'` tout en indiquant sa position.

4.2. Écrire un programme qui demande à l'utilisateur de saisir une chaîne de caractères `c`, affiche le nombre de caractères de cette chaîne, puis échange le premier et le dernier caractère.

4.3. Écrire un programme qui compare la longueur de deux chaînes de caractères saisies par l'utilisateur et affiche un message adapté à chaque cas.

4.4. Soit la chaîne `c = "abcdef"`. Insérer le caractère `","` entre chaque caractère de la chaîne `c`. Résultat attendu : `"a,b,c,d,e,f"`.

Exercice 5 *Chaînes de caractères – fonctions natives et méthodes* **5.1.** Initialisez la variable `s` en lui affectant la chaîne `"abc"`. Utilisez une fonction native pour déterminer sa longueur, puis écrivez une séquence d'opérations qui transforme `"abc"` en `"aaabbbccc"`.

5.2. Initialisez la variable `s` en lui affectant la chaîne `"aaabbbccc"`. Utilisez une méthode pour trouver la première occurrence de `"b"` puis de `"ccc"`. Utilisez une méthode pour remplacer tous les `"a"` par des `"X"`, puis pour ne remplacer que le premier. Indication : la méthode `replace()` accepte un troisième argument facultatif ; `s.replace(a, b, n)` ne remplace que les `n` premières occurrences de `a`.

5.3. Transformez la chaîne `"aaa bbb ccc"` en `"AAA BBB CCC"` puis en `"AAA bbb CCC"`.