
Arithmétique, variables, instructions

Exercices corrigés

Exercice 1

Quelle est la valeur affichée par l'interprète après la séquence d'instructions suivante ?

```
>>> a = 3
>>> a = 4
>>> a = a+2
>>> a
```

Correction :

La valeur affichée est 6.

Exercice 2

Quelle est la valeur affichée par l'interprète après la séquence d'instructions suivante ?

```
>>> a = 2
>>> b = a*a
>>> b = a*b
>>> b = b*b
>>> b
```

Correction :

La valeur affichée est 64.

Les opérations arithmétiques +, -, * et / sont associatives à gauche : les expressions $1+2+3$, $5-3-2$ et $1/2/2$ sont comprises respectivement comme $(1+2)+3$, $(5-3)-2$ et $(1/2)/2$.

Exercice 3

Réécrire les expressions suivantes en explicitant toutes les parenthèses :

1. $1 + 2 * 3 - 4$
2. $1 + 2 / 4 * 3$
3. $1 - a + a * a / 2 - a * a * a / 6 + a * a * a * a / 24$

Correction :

1. $(1 + (2 * 3)) - 4$
2. $1 + ((2 / 4) * 3)$
3. $((((1 - a) + ((a * a) / 2)) - (((a * a) * a) / 6)) + (((((a * a) * a) * a) / 24))$

Exercice 4

Réécrire les expressions suivantes en utilisant aussi peu de parenthèses que possible sans changer le résultat :

1. $1 + (2 * (3 - 4))$
2. $(1+2) + ((5*3)+4)$
3. $(1-((2-3)+4))+(((5-6)+((7-8)/2)))$

Correction :

1. $1 + 2 * (3 - 4)$
2. $1+2+5*3+4$
3. $1-(2-3+4)+5-6+(7-8)/2$

Exercice 5

Ecrire un programme dans l'éditeur :

- Qui affecte le nombre 5 à une variable x
- Qui ajoute 7 au triple du nombre donné
- Qui multiplie le résultat par le nombre donné
- Qui soustrait au résultat le nombre donné
- Qui affiche le résultat obtenu et le type du résultat obtenu.

On tâchera de créer un minimum de variables. Appliquer le programme à un nombre entier puis à un nombre réel (ex : 5.1).

Correction :

```
x = 5
y = 7 + 3 * x
y = y*x
y = y - x
print("le résultat est :", y)
print("le type du résultat est :", type(y))
→ <class 'int'>
```

```
x = 5.1
y = 7 + 3 * x
y = y*x
y = y - x
print("le résultat est :", y)
print("le type du résultat est :", type(y))
→ <class 'float'>
```

Exercice 6

Qu'affichent les instructions suivantes ?

1. `print("1+")`
2. `print(1+)`

Correction :

1. Affiche 1+
2. Erreur de syntaxe

Exercice 7

Que se passe t'il quand on exécute le programme suivant ?

```
a = input("saisir un nombre : ")
print("le nombre suivant est " , a+1)
```

Le rectifier si nécessaire.

Correction :

Tenter d'ajouter comme ici une chaîne de caractères et un entier provoque une erreur. Pour corriger l'erreur, il faut convertir la chaîne en entier avec l'instruction `int`.

```
a = input("saisir un nombre : ")
b = int(a)
print("le nombre suivant est " , b+1)
```

Exercice 8

Que fait la séquence d'instructions suivante ? On supposera qu'à l'origine les variables `a` et `b` contiennent chacune un nombre entier.

```
tmp = a
a = b
b = tmp
```

Correction:

Ce code échange les valeurs des variables `a` et `b` en se servant d'une troisième variable comme zone de stockage temporaire. Le nom de cette variable, `tmp`, évoque son caractère temporaire. Il y a d'autres façons de le faire. Une syntaxe particulière de Python, que nous n'avons pas expliquée, permet des affectations multiples comme :

```
a, b = 55, 89
```

En particulier, on peut écrire :

```
a, b = b, a
```

pour échanger le contenu de `a` et `b`.

Exercice 9

On met deux entiers dans deux boîtes a et b, par exemple 55 et 89. On remplace le contenu de a par la somme de celui de a et de b. Puis on remplace le contenu de b par le contenu de a moins le contenu de b. Enfin, on remplace le contenu de a par son contenu moins celui de b. Que contiennent a et b à la fin de ces opérations ? Programmer cet algorithme en Python.

Correction:

```
a = a + b
b = a - b
a = a - b
```

Exercice 10

Écrire un programme qui demande à l'utilisateur les longueurs (entières) des deux côtés d'un rectangle et affiche son aire.

Correction :

```
longueur = int(input("longueur (en mètres) : "))
largeur = int(input("largeur (en mètres) : "))
aire = longueur * largeur
print("L'air du rectangle vaut : ", aire, "mètres carrés.")
```

Exercice 11

Écrire un programme qui demande à l'utilisateur d'entrer un nombre de secondes et qui l'affiche sous la forme d'heures/minute/secondes.

Correction :

```
n = int(input("entrer le nombre de secondes : "))
h = n // 3600    # nb d'heures
n = n % 3600    # nb de secondes restantes
m = n // 60     # nb de minutes entières
s = n % 60      # nb de secondes restantes
print(h, "heures", m, "minutes", s, "secondes")
```

Exercice 12

Calculateur de pourboire

Vous travaillez dans un restaurant et devez écrire un programme pour calculer le pourboire que vous devriez laisser après un repas.

Consignes :

1. Demandez à l'utilisateur d'entrer le montant total de l'addition (un nombre flottant).
2. Demandez à l'utilisateur de choisir le pourcentage de pourboire qu'il souhaite laisser (10%, 15%, ou 20%).
3. Calculez et affichez :
 - Le montant du pourboire.
 - Le montant total (addition + pourboire).

Correction :

```
# Demander à l'utilisateur d'entrer le montant de l'addition
montant_addition = float(input("Entrez le montant total de l'addition : "))

# Demander à l'utilisateur de choisir le pourcentage de pourboire
pourcentage_pourboire = int(input("Quel pourcentage de pourboire souhaitez-vous
laisser ? (10, 15, 20) : "))

# Calculer le montant du pourboire
montant_pourboire = (montant_addition * pourcentage_pourboire) / 100

# Calculer le montant total (addition + pourboire)
montant_total = montant_addition + montant_pourboire

# Afficher le montant du pourboire et le montant total
print(f"Montant du pourboire : {montant_pourboire:.2f}")
print(f"Montant total à payer : {montant_total:.2f}")
```