
Partie 1 – Les bases : comprendre les instructions

Exercice 1 – Un premier calcul

On exécute le programme suivant :

```
1 x = float(input("x = "))
2 y = x + 8
3 print(y)
```

1. Quelle valeur s'affiche si l'utilisateur saisit 5 ?
2. Quelle valeur s'affiche s'il saisit -3 ?
3. Que calcule ce programme à partir de la valeur saisie ?

Exercice 2 – Compléter un programme

On souhaite saisir un nombre, le multiplier par 3, puis ajouter 2 et afficher le résultat.

Compléter le programme :

```
1 x = float(input("x = "))
2 y = .....
3 print(.....)
```

Exercice 3 – Comprendre les affectations

On exécute :

```
1 x = 4
2 x = x + 3
3 x = 2*x
4 print(x)
```

1. Quelle est la valeur de `x` après la première instruction ?
2. Quelle est sa valeur après la deuxième ?
3. Quelle valeur est affichée à la fin ?

Exercice 4 – Écrire un algorithme simple

Écrire un programme Python qui demande l'âge d'une personne et affiche son âge dans 5 ans.

Partie 2 – Les conditions : choisir entre deux possibilités

Exercice 5 – Positif ou négatif?

```
1 x = float(input("x = "))
2 if x >= 0:
3     print("Positif ou nul")
4 else:
5     print("Négatif")
```

Indiquer ce qui s'affiche pour chacune des valeurs suivantes : 7, -2 et 0.

Exercice 6 – Un tarif simple

Un cinéma propose le tarif suivant :

- moins de 12 ans : 6 euros ;
- à partir de 12 ans : 9 euros.

Compléter le programme :

```
1 age = int(input("Age : "))
2 if age < .....:
3     prix = .....
4 else:
5     prix = .....
6 print(prix)
```

Exercice 7 – Une réduction

Un magasin accorde une réduction de 5 euros si le montant des achats est supérieur ou égal à 50 euros.

```
1 prix = float(input("Montant des achats : "))
2 if prix >= 50:
3     prix = prix - 5
4 print(prix)
```

1. Quel montant s'affiche pour un achat de 40 euros ?
2. Quel montant s'affiche pour un achat de 60 euros ?
3. Quel montant s'affiche pour un achat de 50 euros ?

Exercice 8 – Écrire une condition

Un élève est admis si sa moyenne est supérieure ou égale à 10.

Écrire un programme qui demande sa moyenne et affiche `Admis` ou `Non admis`.

Partie 3 – La boucle for : répéter un calcul**Exercice 9 – Comprendre une boucle**

```
1 for i in range(1, 5):  
2     print(i)
```

1. Quelles valeurs prend la variable `i` ?
2. Combien de fois l'instruction `print(i)` est-elle exécutée ?
3. Qu'affiche le programme ?

Exercice 10 – Une tirelire

Une tirelire contient initialement 20 euros. On y ajoute 5 euros chaque semaine.

```
1 x = 20  
2 for i in range(1, 4):  
3     x = x + 5  
4     print(x)
```

1. Combien de fois ajoute-t-on 5 euros ?
2. Quelle somme contient la tirelire à la fin ?
3. Que faudrait-il modifier pour simuler 6 semaines ?

Exercice 11 – Compléter une boucle

On souhaite afficher les nombres de 1 à 10.

```
1 for i in range(.....):  
2     print(i)
```

Compléter l'instruction `range` pour obtenir le résultat souhaité.

Exercice 12 – Calculer une somme

On souhaite calculer la somme des cinq premiers entiers strictement positifs.

```
1 S = 0
2 for i in range(1, 6):
3     S = S + i
4 print(S)
```

1. Compléter le tableau suivant.

Valeur de i	Valeur de S après l'ajout
1	
2	
3	
4	
5	

2. Quelle valeur est affichée ?

Partie 4 – La boucle `while` : répéter jusqu'à une condition

Exercice 13 – Une somme qui augmente

```
1 x = 10
2 n = 0
3 while x < 25:
4     x = x + 5
5     n = n + 1
6 print(n)
```

1. Quelle est la valeur initiale de x ?
2. Quelle est la valeur de x après le premier passage dans la boucle ?
3. Combien de fois la boucle est-elle exécutée ?
4. Quelle valeur est affichée ?

Exercice 14 – Compléter une boucle

Une tirelire contient 10 euros. On ajoute 2 euros chaque semaine. On veut déterminer le nombre de semaines nécessaires pour atteindre ou dépasser 20 euros.

Compléter le programme :

```
1 x = 10
2 n = 0
3 while x ..... 20:
4     x = .....
5     n = .....
6 print(n)
```

Exercice 15 – Comparer for et while

Pour chaque situation, indiquer la boucle la plus adaptée.

1. Répéter un calcul exactement 8 fois.
2. Ajouter 10 euros chaque mois jusqu'à atteindre 100 euros.
3. Afficher les nombres de 1 à 20.
4. Doubler un nombre jusqu'à ce qu'il dépasse 1 000.

Partie 5 – Les fonctions Python

Exercice 16 – Utiliser une fonction

```
1 def carre(x):
2     return(x**2)
3 print(carre(3))
```

1. Quel calcul réalise la fonction `carre` ?
2. Quelle valeur affiche le programme ?
3. Que faudrait-il écrire pour afficher le carré de 7 ?

Exercice 17 – Compléter une fonction

On souhaite créer une fonction qui reçoit un nombre `x` et renvoie le nombre obtenu en multipliant `x` par 4 puis en retirant 1.

```
1 def calcul(x):  
2   y = .....  
3   return(.....)
```

Compléter les deux lignes manquantes.

Exercice 18 – Une fonction avec deux paramètres

```
1 def somme(a, b):  
2   return(a+b)  
3   print(somme(4, 6))
```

1. Combien de paramètres possède la fonction ?
2. Quelle valeur renvoie-t-elle pour `somme(4,6)` ?
3. Quelle valeur afficherait `print(somme(10,3))` ?