

Atelier : Introduction à Python, concepts de base et leurs applications

Partie 1 : Structures de base

1. Les variables

Exercice : Déclarez une variable `age` et assignez-lui votre âge, puis affichez-la.

Solution :

```
age = 25  
  
print("Mon âge est :", age)
```

2. Les commentaires

Exercice : Ajoutez des commentaires à ce programme pour expliquer chaque étape.

Solution :

```
# Demander à l'utilisateur de saisir deux nombres  
  
a = float(input("Entrez le premier nombre : "))  
b = float(input("Entrez le deuxième nombre : "))  
  
# Calculer la somme des deux nombres  
  
somme = a + b  
  
# Afficher le résultat  
  
print("La somme est :", somme)
```

3. L'addition

Exercice : Écrivez un programme Python qui demande à l'utilisateur de saisir deux nombres et affiche leur somme.

Solution :

```
# Demander à l'utilisateur de saisir trois nombres  
  
nombre1 = float(input("Entrez le premier nombre : "))  
nombre2 = float(input("Entrez le deuxième nombre : "))  
nombre3 = float(input("Entrez le troisième nombre : "))  
  
# Calculer le produit des trois nombres  
  
produit = nombre1 * nombre2 * nombre3  
  
# Afficher le résultat  
  
print("Le produit est :", produit)
```

4. La déclaration des chiffres

Exercice : Déclarez deux variables `x` et `y`, assignez-leur des valeurs, puis affichez-les.

Solution :

```
X = int(input("Entrez une première valeur : "))
Y = int(input("Entrez une deuxième valeur : "))
```

5. L'opérateur +

Exercice : Écrivez un programme Python qui demande à l'utilisateur de saisir deux chaînes de caractères et les concatène, puis affiche le résultat.

Solution :

```
chaine1 = input("Entrez la première chaîne : ")
chaine2 = input("Entrez la deuxième chaîne : ")
concatenation = chaine1 + chaine2
print("La concaténation des deux chaînes est :", concatenation)
```

6. Le typecasting (conversion de type)

Exercice 1: Écrivez un programme Python qui convertit un nombre flottant en entier, puis affiche le résultat.

Solution 1:

```
# Demander à l'utilisateur de saisir un nombre flottant
nombre_flottant = float(input("Entrez un nombre flottant : "))
# Convertir le nombre flottant en entier
nombre_entier = int(nombre_flottant)
# Afficher le résultat
print("Le nombre flottant converti en entier est :", nombre_entier)
```

Exercice 2: Écrivez un programme Python qui compte le nombre de caractères dans une chaîne donnée par l'utilisateur et affiche le résultat.

Solution 2:

```
chaine = input("Entrez une chaîne de caractères : ")
nombre_caracteres = len(chaine)
print("Le nombre de caractères dans la chaîne est :", nombre_caracteres)
```

7. La manipulation des listes

Exercice : Écrivez un programme Python qui prend une liste de nombres donnée par l'utilisateur, calcule leur somme et affiche le résultat.

Solution :

```
liste_nombres = input("Entrez une liste de nombres séparés par des espaces : ").split()
liste_nombres = [int(nombre) for nombre in liste_nombres]
somme = sum(liste_nombres)
print("La somme des nombres est :", somme)
```

Partie 2 : Structures conditionnelles

Exercice 1: Écrivez un programme Python qui demande à l'utilisateur de saisir un nombre et vérifie s'il est positif, négatif ou nul.

Solution 1 :

```
nombre = float(input("Entrez un nombre : "))
if nombre > 0:
    print("Le nombre est positif.")
elif nombre < 0:
    print("Le nombre est négatif.")
else:
    print("Le nombre est nul.")
```

Exercice 2 : Écrivez un programme Python qui vérifie si un nombre donné par l'utilisateur est pair ou impair.

Solution 2 :

```
nombre = int(input("Entrez un nombre : "))
if nombre % 2 == 0:
    print("Le nombre est pair.")
else:
    print("Le nombre est impair.")
```

Partie 3 : Déclarations de fonctions

Exercice 1: Écrivez une fonction Python appelée `calcul_moyenne` qui prend deux nombres en argument et retourne leur moyenne.

Solution 1:

```
def calcul_moyenne(a, b):
    return (a + b) / 2

print("La moyenne est :", calcul_moyenne(5, 8))
```

Exercice 2: Écrivez une fonction Python appelée `diviser` qui prend deux nombres en argument et retourne le résultat de la division. Si le dénominateur est égal à zéro, affichez un message d'erreur.

Solution 2:

```
def diviser(dividende, diviseur):
    if diviseur != 0:
        return dividende / diviseur
    else:
        return "Erreur : division par zéro."

print(diviser(10, 2))
print(diviser(8, 0))
```

Partie 4 : Déclarations des variables

Exercice 1: Déclarez une variable `nom` et assignez-lui votre prénom, puis affichez-la.

Solution 1:

```
nom = "Alice"

print("Bonjour, ", nom)
```

Exercice 2: Déclarez deux variables `x` et `y`, assignez-leur des valeurs, puis échangez les valeurs des deux variables et affichez-les.

Solution 2:

```
x = 5
y = 10

# Échange des valeurs des variables
x, y = y, x

print("x est maintenant :", x)
print("y est maintenant :", y)
```

Partie 5 : Opérateurs

Exercice 1 : Écrivez un programme Python qui demande à l'utilisateur de saisir deux nombres et affiche leur somme.

Solution 1 :

```
a = float(input("Entrez le premier nombre : "))
```

```
b = float(input("Entrez le deuxième nombre : "))  
  
somme = a + b  
  
print("La somme est :", somme)
```

Exercice 2 : Écrivez un programme Python qui vérifie si un nombre donné par l'utilisateur est pair ou impair.

Solution 2 :

```
nombre = int(input("Entrez un nombre : "))  
  
if nombre % 2 == 0:  
    print("Le nombre est pair.")  
else: print("Le nombre est impair.")
```

Partie 6 : Conversion de types de données

Exercice 1 : Écrivez un programme Python qui demande à l'utilisateur de saisir un nombre entier et le convertit en flottant, puis affiche le résultat.

Solution 1:

```
entier = int(input("Entrez un nombre entier : "))  
  
flottant = float(entier)  
  
print("Le nombre en flottant est :", flottant)
```

Exercice 2 : Écrivez un programme Python qui convertit une liste de nombres en une chaîne de caractères et affiche le résultat.

Solution 2 :

```
liste_nombres = [1, 2, 3, 4, 5]  
  
chaine_nombres = ".join(map(str, liste_nombres))  
  
print("La liste convertie en chaîne est :", chaine_nombres)
```