

TP 5 : Découverte des types de données en Python



Bienvenue dans le monde des **types de données** en Python ! Les variables en Python peuvent prendre plusieurs formes, et dans ce TP, nous allons explorer les principales : **entiers**, **floats**, **chaînes de caractères (strings)**, **listes** et **booléens**.

1. Les entiers (**int**) :

Les entiers sont les nombres **sans virgule**, donc tout ce qui est "complet" : 1, 42, -100, etc.

Exemple :

```
mon_nombre = 42 # Ceci est un entier
print(mon_nombre) # On affiche l'entier
```

Exercice 1 :

Déclarez une variable **mon_age** qui contient votre âge (ou l'âge de votre tortue imaginaire) et affichez une réponse qui donnera votre âge dans 12 ans avec **print()**.

2. Les nombres décimaux (**float**) :

Les **floats** sont les nombres avec des **virgules** (en fait des points en Python, mais on ne va pas trop chipoter). Ils sont parfaits pour les calculs précis comme la distance parcourue par une tortue.

Exemple :

```
distance_tortue = 3.14 # Un float (nombre décimal)
print(distance_tortue)
```

Exercice 2 : Déclarez une variable **ma_taille** avec une valeur décimale (en cm, ou en pizzas si vous préférez !) et affichez-la.

3. Les chaînes de caractères (**str**) :

Les **strings** sont des chaînes de texte, littéralement des phrases ou des mots. Elles sont encadrées par des guillemets (' ', " "). Que ce soit un nom, une question ou une blague sur les tortues, tout est une **string**.

Exemple :

```
prenom = "Python" # Une chaîne de caractères
print("Bonjour", prenom)
print(prenom[0]) # permet de récupérer le 1er caractère de la chaîne (qui commence à 0)
print(prenom[3]) # permet de récupérer le 4ème caractère de la chaîne (qui commence à 0)
```

Exercice 3 : Créez une variable **mon_prenom** contenant votre prénom une autre **age** pour votre âge. Ensuite, affichez un message de bienvenue du style : "Bonjour, je suis [votre prénom] et j'adore Python ! et j'ai [votre âge] ans".

4. Les listes (**list**) :

Les **listes** permettent de regrouper plusieurs valeurs dans une seule variable. C'est comme une boîte à outils où vous pouvez stocker des entiers, des floats, des strings, ou même d'autres listes. Bref, une vraie boîte à tout faire.

Exemple :

```
ma_liste = [1, 2, 3, "quatre", 5.5]
print(ma_liste)
print(ma_liste[0]) ] # permet de récupérer le 1er élément de la chaîne (qui commence à 0)
print(ma_liste[3]) # permet de récupérer le 4ème élément de la chaîne (qui commence à 0)
```

Faire une somme avec for :

Imaginons que nous avons des nombres qui sont stockés dans une liste : [12, 3, 6, 18, 7, 9, 15].

Nous voulons additionner tous les nombres de cette liste. Avec une boucle **for**, c'est possible :

```
liste = [12, 3, 6, 18, 7, 9, 15]
somme = 0 #on initialise la variable somme à 0
for i in range (0, len(liste)): #len(liste) compte le nbre d'éléments de la liste
    somme = somme + liste[i] #On rajoute 1 à 1 chaque nombre de la liste

print(somme) #On affiche le résultat de la somme
```

Exercice 4 : Créez une liste **mes_notes** contenant vos trois dernières notes en maths (ou de pizzas, à vous de voir). Affichez ensuite cette liste.

Bonus :

- Affichez la première note avec `print(mes_notes[0])` (les indices commencent à 0 en Python).
- Créez une fonction **moyenne** qui prendra comme argument la liste de vos notes et retournera la moyenne de celles-ci.

Voici le début de ce code :

```
def moyenne(notes):
    ...
    return M
```

5. Les booléens (**bool**) :

Les **booléens** ne peuvent prendre que deux valeurs : **True** (vrai) ou **False** (faux). Ils sont souvent utilisés pour vérifier des conditions.

Exemple :

```
aime_python = True
print(aime_python)
2 == 3
2 != 3
2 < 3
2 <= 3
```

Récapitulatif des commandes à retenir :

- **int** : pour les nombres entiers.
- **float** : pour les nombres décimaux.
- **str** : pour les chaînes de caractères.
- **list** : pour les listes de valeurs.
- **bool** : pour les valeurs True/False.

Avec ces notions, vous êtes prêts à dompter les variables comme un maître Python

