
Annotations de types (Type hinting)

Exercice

Exercice : Implémenter une fonction avec des annotations de types

Vous allez écrire une série de fonctions Python annotées avec des types pour améliorer la lisibilité du code et éviter des erreurs potentielles.

Objectif 1 : Annoter une fonction simple

Écrivez une fonction qui prend un nombre entier et retourne son double. Ajoutez des annotations de types à la fonction.

```
def doubler(nombre):  
    return nombre * 2
```

1. Annoter correctement cette fonction avec des types.
2. Tester cette fonction avec plusieurs entrées.

Objectif 2 : Annoter une fonction avec plusieurs types

Écrivez une fonction qui prend en entrée un prénom (type str) et un âge (type int), et qui renvoie un message de bienvenue (type str).

```
def message_bienvenue(prenom, age):  
    return f"Bienvenue {prenom}, tu as {age} ans."
```

1. Annoter cette fonction avec des types appropriés.
2. Tester cette fonction avec plusieurs entrées.

Objectif 3 : Annotations complexes avec des collections

Écrivez une fonction qui prend en entrée une liste de nombres entiers et qui retourne la somme de ces nombres. Ajoutez des annotations de types pour indiquer que l'entrée est une liste d'entiers et la sortie est un entier.

```
def somme_liste(liste):  
    return sum(liste)
```

1. Annoter cette fonction avec les types adéquats.
2. Tester cette fonction avec différentes listes de nombres.

Objectif 4 : Annotation avec un dictionnaire

Annoter la fonction pour indiquer que l'entrée est un dictionnaire avec des chaînes de caractères (str) pour les noms et prénoms, et un entier (int) pour l'âge.

```
def infos_eleve(eleve):  
    return f"Nom : {eleve['nom']}, Prénom : {eleve['prenom']}, Âge : {eleve['age']} ans"
```