La fonction print en Python

Print en Python

La fonction print permet d'afficher le contenu d'un ou plusieurs objet(s) dans la console. La syntaxe de la fonction est print suivi de parenthèses : print (a, b, c). Le message sera transformé en chaîne de caractères peu importe le type des objets passés en paramètres.

Si le contenu d'un objet ne peux pas être affiché de manière lisible pour un humain, print affichera la référence de l'objet.

test.py

```
    print('commentcoder.com')  # commentcoder.com
    print('commentcoder.com', 42, [1, 2, 3])  # commentcoder.com 42 [1,
2, 3]

    def test():
       pass

print(test)  # <function test at 0x142424242>
```

La fonction print permet entre autres :

- D'afficher tout ce qu'on veut en Python
- D'afficher du texte en couleur
- De modifier son formatage avec l'aide d'autres paramètres comme sep et end

Commençons sans plus tarder à explorer toutes les capacités de la fonction print en Python 3 (et Python 2 en fin d'article) !

https://youtu.be/n2aAnQVZOfA?t=53

Comment utiliser la fonction print en Python ?

Pour utiliser la fonction print en Python, on lui passe des paramètres qui seront transformés en strings :

test002.py

```
# Afficher une chaîne de caractères :
print("Bienvenue sur commentcoder.com)
# Affichera "commentcoder.com" sur la sortie standard

# Afficher un nombre :
print(42)
# Affichera "42" sur la sortie standard
```

```
# Afficher le contenu d'une variable :
nombre = 42
print(nombre)
# Affichera "42" sur la sortie standard
```

Il est aussi possible d'afficher plusieurs objets Python avec la fonction print :

test003.py

```
## Afficher plusieurs objets :
print(1, 2, 3)
# Afficher "1 2 3"

# On peut aussi mixer les types d'objets :
print('École', 42)
# Afficher "École 42"

# En utilisant l'unpacking/destructuration
liste = [1, 2, 3, 4, 5]
print(liste[0], *liste[1:-1], liste[-1])
# Affichera 1 2 3 4 5
# Explications :
# liste[0] = 1 (le premier élément de la liste)
# liste[1:-1] = [2, 3, 4] (les éléments en partant du 1er non-inclus, jusqu'au dernier non-inclus)
# L'opérateur * permet de déstructurer la liste [2, 3, 4] en 3 nombres 2, 3 et 4
# liste[-1] = 5 (le dernier élément de la liste)
```