Python Cheat Sheet pour débutant

Les bases

Obtenir de l'aide et connaître les types d'objets

python

```python
# Tout ce qui suit le symbole dièse n'est pas exécuté par Python et
# permet de fournir des indications sur le fonctionnement de votre code

12 * 2  # Commentaire explicatif

help(print)  # Afficher la documentation pour la fonction print

type('x')  # Obtenir le type d'un objet
```

Importer une bibliothèque Python

python

```python
import pandas          # Importer une bibliothèque sans alias

import pandas as pd    # Importer une bibliothèque avec un alias

from pandas import Series  # Importer un objet d'une bibliothèque
```

Chaînes de caractères

Créer des chaînes de caractères

python

```python
# Créer une chaîne de caractères avec des guillemets simples ou doubles
chaine = "Jedha"

# Insérer une citation dans une chaîne avec le caractère d'échappement \
citation = "Il a dit, \"Jedha\""

# Créer des chaînes multi-lignes avec des triples guillemets
multi_ligne = """
Une chaîne
multi-lignes
"""
```

Combiner et séparer des chaînes

python

```python
# Concaténer des chaînes avec +
"Data" + "Science"  # 'DataScience'

# Répéter des chaînes avec *
4 * "cyber "   # 'cyber cyber cyber cyber '
3 * "cyber"    # 'cybercybercyber'

# Diviser une chaîne à partir d'un délimiteur
"boulangeries".split("e")  # ['boulang', 'ri', 's']
```

Muter des chaînes

python

```python
str = "Jedha et Python"

str.upper()    # 'JEDHA ET PYTHON'
str.lower()    # 'jedha et python'
str.title()    # 'Jedha Et Python'
str.replace("J", "P")  # 'Pedha et Python'
```

Listes

Créer une liste

python

```python
# Créer des listes à l'aide de [], avec les éléments séparés par des virgules
numbers = [5, 7, 2]
```

Trier, inverser et compter les éléments d'une liste

python

```python
sorted(numbers)    # Retourne [2, 5, 7]

numbers.sort()     # Trie la liste en place (retourne None)
print(numbers)     # Retourne [2, 5, 7]

reversed(numbers)  # Retourne [2, 7, 5]

numbers.reverse()  # Inverse la liste en place (retourne None)
print(numbers)     # Retourne [2, 5, 7]

numbers.count(7)   # Retourne 1
```

Concaténer et répéter des listes

python

```python
x = [2, 4, 6]
y = [8, 10, 12]

x + y     # [2, 4, 6, 8, 10, 12]
2 * x     # [2, 4, 6, 2, 4, 6]
```

Sélectionner des éléments d'une liste

python

```python
fruits = ['pomme', 'banane', 'cerise', 'datte', 'figue']

fruits[0]    # 'pomme'       — premier élément
fruits[-1]   # 'figue'       — dernier élément
fruits[1:3]  # ['banane', 'cerise']  — de l'index 1 (inclus) à 3 (exclu)
fruits[2:]   # ['cerise', 'datte', 'figue']  — de l'index 2 à la fin
fruits[:3]   # ['pomme', 'banane', 'cerise'] — du début à l'index 3 (exclu)
```

Dictionnaires

Créer des dictionnaires

python

```python
# Créer un dictionnaire avec {}
{'x': 5, 'y': 10, 'z': 15}
```

Fonctions et méthodes de dictionnaire

python

```python
d = {'x': 5, 'y': 10, 'z': 15}

d.keys()    # dict_keys(['x', 'y', 'z'])
d.values()  # dict_values([5, 10, 15])
d['x']      # 5
```

Boucles "for" et "while"

python

```python
# Boucle for pour itérer sur une liste
for item in [1, 2, 3]:
    print(item)  # Affiche 1 puis 2 puis 3

# Boucle while jusqu'à ce que la condition soit remplie
count = 0
while count < 5:
    count += 1
    print(count)  # Affiche les nombres de 1 à 5
```

Opérateurs

Opérateurs arithmétiques

python

```python
5 + 7    # Addition
10 - 3   # Soustraction
3 * 7    # Multiplication
15 / 5   # Division
15 // 4  # Division entière
2 ** 3   # Puissance
17 % 5   # Modulo (reste de la division)
```

Opérateurs de comparaison numérique

python

```python
5 == 5   # Égalité
5 != 4   # Inégalité
6 > 3    # Supérieur à
7 >= 7   # Supérieur ou égal à
4 < 8    # Inférieur à
6 <= 6   # Inférieur ou égal à
```

Opérateurs logiques

python

```python
not (3 == 3)         # NOT logique
(2 != 2) and (2 < 3)  # AND logique
(3 >= 3) or (2 < 1)   # OR logique
(2 != 2) ^ (3 < 2)    # XOR logique
```

Utilisation de fonctions

python

```python
# Définir une fonction
def ajouter(a, b):
    return a + b

# Utiliser la fonction
resultat = ajouter(5, 7)
print(resultat)  # Affiche 12
```

Classes et Objets

python

```python
# Définir une classe
class Chien:
    def __init__(self, nom, age):  # Attributs
        self.nom = nom
        self.age = age

    def aboyer(self):  # Méthode
        print("Woof!")

# Créer un objet
mon_chien = Chien("Rex", 5)
mon_chien.aboyer()    # Affiche "Woof!"
print(mon_chien.nom)  # Affiche "Rex"
```

Gestion des fichiers

python

```python
# Ouvrir et lire un fichier
with open('fichier.txt', 'r') as file:
    contenu = file.read()
    print(contenu)

# Écrire dans un fichier en écrasant son contenu
with open('fichier.txt', 'w') as file:
    file.write('Bonjour, monde!')

# Ajouter du contenu à un fichier existant
with open('fichier.txt', 'a') as file:
    file.write('\nAjouter cette ligne au fichier.')
```