

# Python



**Welcome!**  
Rida Messaoudene



# Notation

## Savoir être

Assiduité, retard, comportement,  
Participation

## Projet personnel

Qcm,  
developpement  
de site



# Table of Contents

01

Rappels

02

Logique

03

Web Scrapping



# 01

## Rappels

# List

- structure de données native en Python, accessible en utilisant le type ***list***.
- Peut contenir des éléments de différents types de données, tels que des entiers, des chaînes de caractères, des objets, etc.
- Les éléments d'une liste sont accessibles par leur indice.
- La taille d'une liste peut être modifiée dynamiquement en ajoutant ou en supprimant des éléments.

```
noms = ["Alice", "Bob", "Charlie", "David"]
```



# Dictionnaires

- Structure de données qui permet de stocker et d'organiser des données de manière associée.
- Clés uniques.
- Les dictionnaires sont mutables.

```
mon_dictionnaire = {  
    "clé1": "valeur1",  
    "clé2": "valeur2",  
    "clé3": "valeur3"  
}
```



# Dictionnaires

- Structure de données qui permet de stocker et d'organiser des données de manière associée.
- Clés uniques.
- Les dictionnaires sont mutables.

```
mon_dictionnaire = {  
    "clé1": "valeur1",  
    "clé2": "valeur2",  
    "clé3": "valeur3"  
}
```

# Boucles

```
avec_soleil = True
en_semaine = False
if avec_soleil and not en_semaine:
    print("on va à la plage !")
elif avec_soleil and en_semaine:
    print("on va au travail !")
else:
    print("on reste à la maison !")
```

# Fonctions

```
def calculer_somme(a, b):  
    resultat = a + b  
    return resultat
```

```
res = calculer_somme(2, 3)  
print("le résultat est : {res}")
```

# Import

```
import mon_module  
resultat = mon_module.ma_fonction()
```

# Import spécifique

```
from mon_module import ma_fonction  
resultat = ma_fonction()
```





# Cas pratique



# 02

## Scraping

# Requests

```
import requests  
  
url = "https://www.gov.uk/search/news-and-communications"  
page = requests.get(url)  
  
# Voir le code html source  
print(page.content)
```

# Beautiful Soup

- Parser HTML
- pip install [beautifulsoup4](#)

```
import requests  
  
from bs4 import BeautifulSoup  
  
url = "https://www.gov.uk/search/news-and-communications"  
page = requests.get(url)  
soup = BeautifulSoup(page.content, 'html.parser')
```



# Écriture CSV

```
# Créer une liste pour les en-têtes
en_tete = ["titre", "description"]

# Créer un nouveau fichier pour écrire dans le fichier appelé « data.csv »
with open('data.csv', 'w') as fichier_csv:
    # Créer un objet writer (écriture) avec ce fichier
    writer = csv.writer(fichier_csv, delimiter=',')
    writer.writerow(en_tete)
    # Parcourir les titres et descriptions – zip permet d'itérer sur deux listes ou plus à la fois
    for titre, description in zip(titres, descriptions):
        # Créer une nouvelle ligne avec le titre et la description à ce moment de la boucle
        ligne = [titre, description]
        writer.writerow(ligne)
```

# Cas pratique

# Thanks

Avez-vous des questions ?  
rida@lamerkanterie.fr

lamerkanterie.fr

