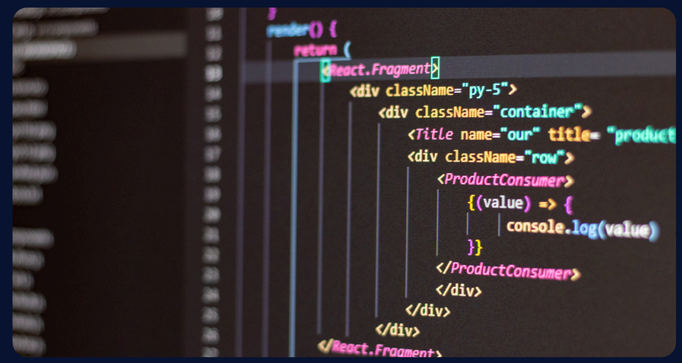


Seance 4 : Python

Laffet takwa

Exercices

Écrire une fonction somme_chiffres(n) qui prend un nombre entier n et retourne la somme de tous ses chiffres.

```
somme_chiffres(1234)  # 1 + 2 + 3 + 4 = 10
somme_chiffres(409)   # 4 + 0 + 9 = 13
```

Qu’est-ce qu’une fonction récursive

une fonction récursive est une fonction qui s’appelle elle-même pour résoudre un problème en le divisant en sous-problèmes plus petits.

- Idée principale :
- Résoudre un problème complexe en le ramenant à un problème plus simple.

Structure d’une fonction récursive

Une fonction récursive comporte toujours deux parties essentielles :

1. Cas de base (ou condition d’arrêt)
    - C’est la condition qui arrête la récursion, sinon la fonction s’appellerait indéfiniment et provoquerait une erreur (RecursionError).
    - Exemple : pour la factorielle, quand n == 0.
2. Appel récursif
    - C’est l’endroit où la fonction s’appelle elle-même sur un sous-problème plus petit.

```python
def fonction_recursive(parametre):
    if condition_de_base(parametre):
        return valeur_de_base
    else:
        return fonction_recursive(sous_probleme(parametre))
```

Exemple 1 : Factorielle

La factorielle de n est le produit de tous les entiers de 1 à n.
Formule :

- n! = n × (n-1)!
- 0! = 1 (cas de base)

```
def factorielle(n):
    # Cas de base
    if n == 0:
        return 1
    # Appel récursif
    else:
        return n * factorielle(n-1)

# Test
print(factorielle(5))  # 120
```

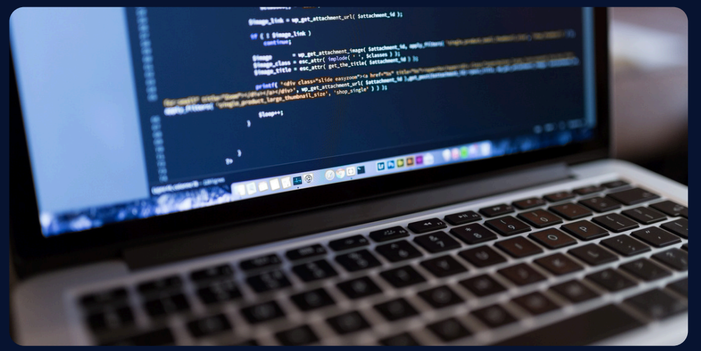
Points importants à retenir

1. Toujours définir un cas de base pour éviter une boucle infinie.
2. Ne pas trop récuser : Python a une limite (environ 1000 appels récursifs par défaut).
3. On peut la modifier avec sys.setrecursionlimit(n) mais attention à la mémoire.
4. Éviter la récursion inutile : certaines fonctions peuvent être mieux faites avec des boucles pour gagner en performance.
5. La récursion peut être élégante pour des problèmes “arborescents” : arbres, graphes, dossiers, fractales…

Exercices

Écrire une fonction récursive somme_chiffres(n) qui prend un nombre entier n et retourne la somme de tous ses chiffres.

```
somme_chiffres(1234)  # 1 + 2 + 3 + 4 = 10
somme_chiffres(409)   # 4 + 0 + 9 = 13
```



Thank You