Cours et Exercices : La Récursivité en Python

La **récursivité** est une technique de programmation où une fonction s'appelle elle-même pour résoudre un problème en le divisant en sous-problèmes plus simples. C'est un concept fondamental en informatique, que l'on retrouve aussi dans la nature et l'art.

Quelques analogies de la récursivité :

Les poupées russes : un objet contenant un objet similaire à l'intérieur.

La Vache qui rit : sur la boîte, la vache porte des boucles d’oreilles représentant... encore une boîte de La Vache qui rit !

La pochette de l’album *Ummagumma* (Pink Floyd, 1969) : une image contenant une image, etc.

La récursivité en Python

Une fonction récursive doit toujours avoir : - Un **cas de base** qui arrête la récursion - Un **appel récursif** qui rapproche du cas de base

Factorielle

```
def factorielle(n):
if n == 0:
return 1
else:
return n * factorielle(n-1)

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

Somme des n premiers entiers

```
def somme_n(n):
if n == 0:
return 0
return n + somme_n(n-1)

print(somme_n(5)) # 15
```

Suite de Fibonacci

```
def fibonacci(n):
if n <= 1:
return n
return fibonacci(n-1) + fibonacci(n-2)
```

```
print(fibonacci(6)) # 8
```

Comptage d’éléments dans une liste

```
def compter(liste):
if liste == []:
return 0
return 1 + compter(liste[1:])

print(compter([1,2,3,4])) # 4
```

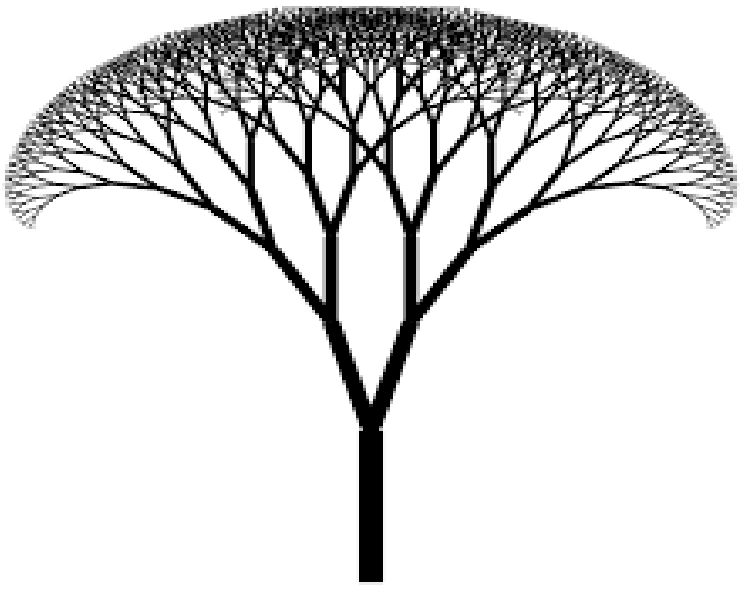
Parcours d’un dossier (simplifié)

```
import os

def lister_fichiers(dossier):
for nom in os.listdir(dossier):
chemin = os.path.join(dossier, nom)
if os.path.isdir(chemin):
lister_fichiers(chemin)
else:
print(chemin)
```

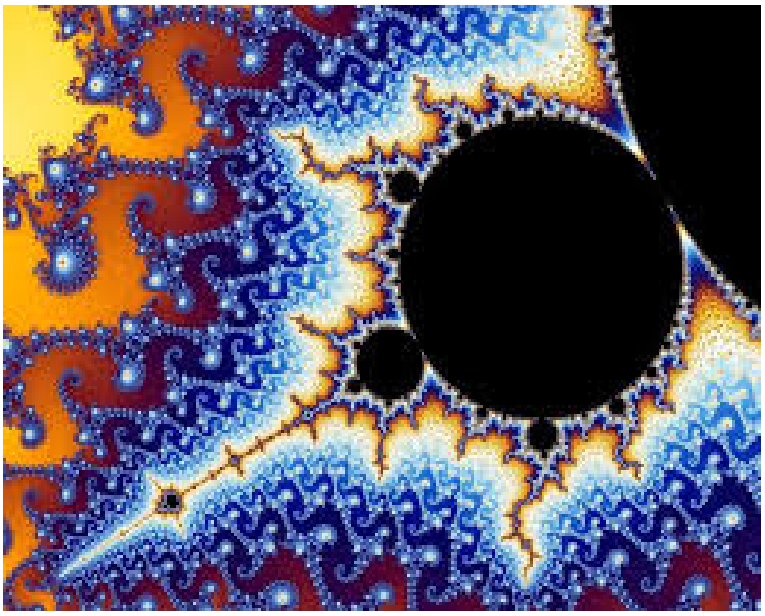
Tri rapide (QuickSort)

```
def quicksort(tab):
if len(tab) <= 1:
return tab
pivot = tab[0]
gauche = [x for x in tab[1:] if x <= pivot]
droite = [x for x in tab[1:] if x > pivot]
return quicksort(gauche) + [pivot] + quicksort(droite)

print(quicksort([5,3,8,2,1,4])) # [1,2,3,4,5,8]
```

Les fractales : récursivité visuelle



Un arbre fractal généré par récursion : chaque branche se divise en deux sous-branches.



L'ensemble de Mandelbrot : une fractale célèbre qui illustre la récursivité infinie des formes.

Exercices

Exercice 1

Écris une fonction récursive compter_voyelles(chaine) qui compte le nombre de voyelles dans une chaîne

Exercice 2

Écris une fonction récursive maximum(liste) qui renvoie le plus grand élément d·une liste sans utiliser max(

Exercice 3

Écris une fonction récursive pair(n) qui renvoie True si n est pair, et False sinon, sans utiliser l·opérateur %

Exercice 4

Écris une fonction romain(n) qui renvoie une chaine correspondant à l'écriture romaine du nombre en paramètre.

Exercice 5

Écris une fonction qui inverse une chaîne de caractères de manière récursive.

Exercice 6

Bonus : Dessine un arbre fractal en utilisant le module turtle de Python.