

# cours\_\_3

February 1, 2024

## 1 Cours 3

### Notions

Dans cette partie, on explorera: - les fonctions imbriquées, - les variables locales et globales - lecture et création de pseudocode

## 2 Fonctions imbriquées

L'appel de fonction peut être multiple et imbriqué. On peut appeler une fonction à l'intérieur d'une autre et en utiliser le retour.

Mise en contexte: Les jeux Olympiques Les jeux Olympiques, en temps normal, sont alternés entre les jeux d'été et les jeux d'hiver à chaque 2 ans. Si on est dans une année bissextile, il y aura des jeux d'été et sinon, ce sera des jeux d'hiver. On peut faire une fonction imbriquée pour fournir une année à la fonction `jeux_olympiques(annee)` et celle-ci me retournera "été" ou "hiver".

Voici un exemple:

```
def annee_bissextile(annee):
    if annee % 4 == 0:
        return True;
    return False;

def jeux_olympiques(annee):
    bissextile = annee_bissextile(annee)
    if bissextile == True:
        return "été"
    elif annee % 2 == 0:
        return "hiver"
```

### 2.0.1 Pourquoi faire du traitement de données dans des fonctions séparées selon vous?

Voici un autre exemple avec un seul appel qui fait plusieurs sous-appels. Faisons la TRACE de cet algorithme.

```
def fonction_a():
    print('fonction_a() début')
    fonction_b()
    fonction_d()
```

```

    print('fonction_a() fin')

def fonction_b():
    print('fonction_b() début')
    fonction_c()
    print('fonction_b() fin')

def fonction_c():
    print('fonction_c() début')
    print('fonction_c() fin')

def fonction_d():
    print('fonction_d() début')
    print('fonction_d() fin')

#Appel de la fonction a
fonction_a()

```

### 3 Variables globales et variables locales

En programmation, les variables sont des espaces de stockage permettant de stocker des données. En Python, il existe deux types de variables importantes : les variables locales et les variables globales. Comprendre la différence entre ces deux types de variables est essentiel pour écrire des programmes efficaces et éviter les erreurs.

#### 3.1 Variables locales

Une variable locale est déclarée à l'intérieur d'une fonction et n'est accessible qu'à l'intérieur de cette fonction. Elle est créée lorsque la fonction est appelée et détruite lorsque la fonction se termine. Les variables locales sont utilisées pour stocker des données temporaires nécessaires uniquement à l'intérieur d'une fonction.

**Exemple :** Calcul de la densité

```

def calculer_densite(masse, volume):
    densite = masse / volume
    return densite

# Utilisation de la variable locale "densite" uniquement à l'intérieur de la fonction
resultat_densite = calculer_densite(150, 30)
print("La densité est :", resultat_densite)

```

Ici, la variable `densite` est locale à la fonction `calculer_densite`. Elle n'a aucune signification en dehors de cette fonction.

#### 3.2 Variables globales

Une variable globale est déclarée en dehors de toutes les fonctions et est accessible à l'ensemble du programme. Les variables globales conservent leur valeur même après l'exécution de la fonction. Elles sont utilisées lorsque la donnée doit être partagée entre plusieurs parties du programme.

```

constante_acceleration_gravite = 9.8

def calculer_energie_potentielle(masse, hauteur):
    energie_potentielle = masse * constante_acceleration_gravite * hauteur
    return energie_potentielle

# Utilisation de la variable globale "constante_acceleration_gravite" dans la fonction
resultat_energie = calculer_energie_potentielle(2, 5)
print("L'énergie potentielle est :", resultat_energie)

```

Ici, la variable `constante_acceleration_gravite` est globale et peut être utilisée dans toute partie du programme. C'est l'une des utilités des variables globales.

Une variable globale peut être **modifiée**, même à l'extérieur d'une fonction. Par exemple, si on utilise une variable globale appelée `compteur`, on pourra incrémenter le compteur à partir de fonctions distinctes. Il y a toutefois un élément à ajouter pour modifier une variable globale.

### 3.2.1 Exemple:

on veut garder la trace du nombre de fois qu'un échantillon sera manipulé lors d'une expérience.

```

compteur_global = 0
echantillon_test = 5

def prise_temperature(temperature, num_echantillon):
    global compteur_global
    if (num_echantillon == echantillon_test):
        compteur_global += 1
    if (temperature > 37):
        return "chaud"
    else:
        return "froid"

def prise_pH(pH, num_echantillon):
    global compteur_global
    if (num_echantillon == echantillon_test):
        compteur_global += 1
    if (pH < 7):
        return "acide"
    elif (pH == 7):
        return "neutre"
    else:
        return "basique"

print(compteur_global)
prise_pH(8, 6)
prise_pH(8.2, 5)
prise_temperature(28, 5)

```

```
prise_temperature(39, 9)
prise_pH(7.8, 5)
print(compteur_global)
```

Que vaudra le compteur\_global à la fin?

Que se passe-t-il si on ferme le programme et qu'on l'ouvre? Que vaudra alors le compteur\_global?

## 4 Quelques **exercices** pour se pratiquer

[ ]: