ECOLE MAROCAINE DES SCIENCES DE L'INGENIEUR

Membre de HONORIS UNITED UNIVERSITIES

Travaux Pratiques

Fonctions, Modules et récursivité en Python

Module : Python

Dr. ABOUABID Hamza

4 décembre 2024

1 Plan de la Série de TP

1. **TP1 : Introduction aux Fonctions**
2. **TP2 : Portée des Variables et Fonctions Avancées**
3. **TP3 : Modules et Packages**
4. **TP4 : Récursivité**
5. **TP5 : Projet Final**

2 Partie 1 : Introduction aux Fonctions

2.1 Objectifs

— Définir et appeler des fonctions en Python.

— Comprendre l'utilité des fonctions pour structurer le code.

— Utiliser des paramètres et des valeurs de retour.

2.2 Exercice 1 : Fonctions de Base

1. Créez une fonction nommée `saluer` qui affiche "Bonjour tout le monde !".
2. Appelez la fonction `saluer` pour afficher le message.
3. Créez une nouvelle fonction `saluer2` qui modifie la fonction `saluer` pour qu'elle prenne un paramètre `nom` et affiche "Bonjour, [nom] !".
4. Appelez la fonction avec les noms "Alice", "Bob" et "Charlie".

2.3 Exercice 2 : Calcul de l'énergie cinétique

Contexte : En physique, l'énergie cinétique est donnée par $E_c = \frac{1}{2}mv^2$.

1. Créez une fonction `energie_cinetique(masse, vitesse)` qui calcule et retourne l'énergie cinétique.
2. Calculez l'énergie cinétique d'un objet de masse 10 kg se déplaçant à 15 m/s et affichez le résultat avec un message explicatif.

3 Partie 2 : Portée des Variables et Fonctions Avancées

3.1 Objectifs

— Comprendre la portée des variables en Python.

— Utiliser des fonctions imbriquées et des fonctions anonymes (lambda).

3.2 Exercice 3 : Portée des Variables

1. Expliquez ce qui se passe dans le code suivant et prédisez les sorties.
   Code :

```
x = 5

def afficher():
    x = 10
    print("Dans la fonction, x =", x)

afficher()
print("En dehors de la fonction, x =", x)

```

Réponse attendue :

```
Dans la fonction, x = 10
En dehors de la fonction, x = 5
```

2. Modifiez la fonction `afficher` pour qu'elle modifie la variable globale `x` en utilisant le mot-clé `global`.
   Réponse attendue :

```
Dans la fonction, x = 10
En dehors de la fonction, x = 10
```

3.3 Exercice 4 : Fonctions Imbriquées

2.1 Création de fonctions imbriquées

- **Tâche** : Créez une fonction `externe` qui définit une fonction `interne` à l'intérieur. La fonction `interne` doit afficher "Je suis à l'intérieur de la fonction interne". La fonction `externe` doit appeler `interne`.
- **Réponse attendue** :

```
Je suis à l'intérieur de la fonction interne
```

3.4 Exercice 5 : Fonctions Lambda

5.1 Utilisation de fonctions anonymes

- **Tâche** : Utilisez une fonction lambda pour calculer la puissance électrique $P = U \times I$, où U est la tension et I est le courant.
- **Instructions** :

```
variable = lambda par1, par2: retour
```

5.2 Calcul de la puissance

- **Tâche** : Calculez la puissance lorsque la tension est de 230 V et le courant de 5 A.
- **Réponse attendue** :

```
La puissance est de 1150 Watt.
```

4 Partie 3 : Modules et Packages

4.1 Objectifs

— Créer des modules Python personnalisés.

— Importer et utiliser des modules dans un script principal.

— Comprendre la structure des packages.

4.2 Exercice 6 : Création d'un Module

— **Tâche** : Créez un fichier nommé `calculs.py` contenant les fonctions suivantes :

  — `addition(a, b)`

  — `soustraction(a, b)`

  — `multiplication(a, b)`

  — `division(a, b)`

— Dans un script principal (main.py), importez le module `calculs` et utilisez ses fonctions pour effectuer des opérations sur deux nombres saisis par l'utilisateur.

4.3 Exercice 7 : Importation Sélective

— Modifiez le script principal pour importer uniquement les fonctions `addition` et `multiplication` du module `calculs`.

— Effectuez des calculs avec les fonctions importées.

4.4 Exercice 8 : Création d'un Package Simple

— **Tâche** : Créez un dossier nommé `outil_mathematique` contenant un fichier `__init__.py` vide et le module `calculs.py`.

— **Structure du dossier** :

```
outil_mathematique/
    __init__.py
    calculs.py
```

— **Tâche** : Dans le script principal, importez le module `calculs` depuis le package `outil_mathematique`.

5 Partie 4 : Récursivité

5.1 Objectifs

— Comprendre le concept de récursivité.

— Implémenter des fonctions récursives pour résoudre des problèmes.

— Analyser la complexité et les limites de la récursivité.

5.2 Exercice 9 : Factorielle Récursive

- **Rappel** : La factorielle d'un nombre n est définie par :
  - $n! = n \times (n-1)!$ avec $0! = 1$.
- **Tâche** : Créez une fonction récursive `factorielle(n)` qui calcule la factorielle de `n`.
- **Tâche** : Calculez la factorielle de 5 et affichez le résultat. **Réponse attendue** :

```
5! = 120
```

5.3 Exercice 10 : Suite de Fibonacci

- **Rappel** : La suite de Fibonacci est définie par :
  - $F(0) = 0$, $F(1) = 1$
  - $F(n) = F(n-1) + F(n-2)$ pour $n > 1$
- Créez une fonction récursive `fibonacci(n)` qui retourne le n-ième terme de la suite de Fibonacci.
- Affichez les 10 premiers termes de la suite.
- **Réponse attendue** :

```
0
1
1
2
3
5
8
13
21
34
```

5.4 Exercice 11 : Recherche Dichotomique (Binary Search)

Contexte

- **Description** : La recherche dichotomique est une méthode efficace pour trouver un élément dans une liste triée en divisant l'espace de recherche par deux à chaque étape.

Implémentation

- Créez une fonction récursive `recherche_dichotomique(liste, element, debut, fin)` qui retourne l'indice de l'élément recherché ou `-1` s'il n'est pas trouvé.
- Testez la fonction avec une liste triée de nombres et différents éléments à rechercher.
- **Réponse attendue** :

```
L'élément 14 se trouve à l'indice 6.
```

6 Partie 5 : Projet Final

6.1 Objectifs

— Intégrer les concepts de fonctions, modules et récursivité dans un projet complet.

— Appliquer les connaissances à un problème pertinent pour le Génie Électrique ou Industriel.

6.2 Choix du Projet

6.2.1 Option 1 : Analyse de Signal en Génie Électrique

— **Description** : Développez une application qui génère des signaux électriques (sinusoïdal, carré, triangulaire) et analyse leurs caractéristiques (fréquence, amplitude).

— **Exigences** :

  — Créez un module `signaux.py` contenant des fonctions pour générer chaque type de signal.

  — Utilisez des fonctions pour calculer les caractéristiques du signal.

  — Implémentez une fonction récursive pour simuler un filtre passe-bas sur le signal.

— **Suggestions** :

  — Utilisez la bibliothèque `numpy` pour manipuler les tableaux de données.

  — Utilisez `matplotlib` pour visualiser les signaux.

6.2.2 Option 2 : Optimisation de la Chaîne Logistique en Génie Industriel

— **Description** : Développez une application qui optimise le chemin de livraison entre plusieurs points en utilisant l'algorithme du voyageur de commerce (TSP) récursif.

— **Exigences** :

  — Créez un module `logistique.py` contenant des fonctions pour calculer les distances entre points.

  — Utilisez une fonction récursive pour trouver le chemin optimal.

  — Intégrez des fonctions pour lire les coordonnées des points depuis un fichier.

— **Suggestions** :

  — Représentez les points de livraison avec leurs coordonnées (x, y).

  — Utilisez une approche récursive pour générer toutes les permutations possibles (attention à la complexité !).

6.3 Étapes du Projet

1. **Analyse et Conception**

   — Définissez les fonctionnalités de l'application.

   — Identifiez les modules et fonctions nécessaires.

2. **Développement**

   — Implémentez les modules et fonctions conformément aux bonnes pratiques.

   — Commentez le code pour expliquer le fonctionnement.

3. **Tests et Validation**
   — Testez chaque fonction individuellement.
   — Vérifiez l'intégration des modules.

4. **Documentation**
   — Rédigez un rapport expliquant l'approche, les choix techniques et les résultats obtenus.
   — Incluez des schémas ou graphiques si nécessaire.

7 Ressources Utiles

— **Documentation Python** : https://docs.python.org/3/
— **Tutoriels Python** : https://www.w3schools.com/python/
— **Bibliothèques Scientifiques** :
   — `NumPy` pour les calculs numériques : https://numpy.org/
   — `Matplotlib` pour la visualisation : https://matplotlib.org/