

Série de Travaux Pratiques N° 7

Exercice 1 : Création d'un module

Créez un module "conversion.py" qui contient des fonctions pour convertir des unités de mesure.

1. Implémentez les fonctions suivantes :
 - "dollars_to_dirhams(dollars)": Convertit des dollars en dirhams.
 - "meters_to_kilometers(meters)": Convertit des mètres en kilomètres.
2. Testez votre module dans un fichier "main.py".

Exercice 2 : Utilisation des modules

Utilisez les modules standard pour créer un script qui affiche les informations suivantes :

1. Le répertoire courant.
2. La date et l'heure actuelles.
3. La racine carrée d'un nombre donné par l'utilisateur.

Exercice 3 : Écriture de tests unitaires

Écrivez des tests unitaires pour le module "conversion.py" que vous avez créé dans l'exercice 1.

1. Créez un fichier "test_conversion.py".
2. Utilisez "unittest" pour tester toutes les fonctions de votre module.
3. Vérifiez que les fonctions retournent les résultats attendus.

Exercice 4 : Organisation du projet

Organisez un projet Python avec la structure suivante :

```
mon_projet/  
|  
├── src/  
|   ├── __init__.py  
|   ├── math_operations.py  
|   └── string_operations.py  
|  
├── tests/  
|   ├── __init__.py  
|   └── test_math_operations.py  
|  
└── main.py
```

1. Dans "math_operations.py", ajoutez des fonctions pour additionner, soustraire, multiplier et diviser.
2. Dans "string_operations.py", ajoutez des fonctions pour concaténer et mettre en majuscules des chaînes.
3. Écrivez des tests unitaires pour chaque module dans "test_math_operations.py".

Exercice 5 : Utilisation de bibliothèques tierces

Créez un script qui utilise "Pandas" pour analyser un jeu de données simple.

1. Téléchargez un fichier CSV (par exemple, un fichier contenant des informations sur des employés).
2. Utilisez "Pandas" pour lire le fichier et afficher les cinq premières lignes.
3. Calculez et affichez la moyenne d'un champ numérique (par exemple, l'âge des employés).

Exercice 6 : Documentation du code

Ajoutez des docstrings et des commentaires à toutes les fonctions de vos modules.

1. Assurez-vous que chaque fonction a une docstring décrivant son objectif, ses paramètres et sa valeur de retour.
2. Commentez les parties complexes de votre code pour améliorer sa lisibilité.