

Exercices Pratiques de Python

Ali ZAINOUL

Exercices Pratiques de Python

1. **Gestion des versions de Python** : Écrivez un script Python qui vérifie la version actuelle de Python installée sur le système et comparez-la avec une version spécifiée. Si la version est plus ancienne, le script devrait suggérer une mise à jour.
2. **Exploration de la bibliothèque standard** : Écrivez un programme qui utilise le module `os` pour lister tous les fichiers et dossiers dans le répertoire courant et les sous-répertoires. Le programme doit afficher la taille de chaque fichier.
3. **Utilisation de `help()`** : Créez un script Python qui utilise la fonction `help()` pour afficher la documentation de trois fonctions intégrées de Python (par exemple `len()`, `input()`, `print()`) et enregistre cette documentation dans un fichier texte.
4. **Indentation et commentaires** : Écrivez un programme Python qui lit un fichier contenant du code Python mal indenté et commente les lignes de code. Le programme doit réindenter correctement le code et enregistrer la version corrigée dans un nouveau fichier.
5. **Programme autonome** : Développez un programme autonome qui génère un nombre aléatoire entre 1 et 100 et demande à l'utilisateur de deviner ce nombre. Le programme doit fournir des indices (plus haut ou plus bas) et compter le nombre de tentatives jusqu'à ce que l'utilisateur devine correctement.
6. **Utilisation de fonctions intégrées** : Écrivez un programme qui utilise les fonctions `len()`, `input()`, `type()` et `print()` pour demander à l'utilisateur d'entrer une liste de mots, afficher la longueur de chaque mot, et le type de chaque élément dans la liste.

7. **Manipulation de booléens** : Créez un script Python qui demande à l'utilisateur d'entrer une série de valeurs. Le script doit convertir ces valeurs en binaire et afficher le résultat.
8. **Calculs avec des nombres** : Développez un programme qui effectue des opérations mathématiques avancées sur des nombres entiers, flottants et complexes. Le programme doit demander à l'utilisateur d'entrer deux nombres et d'afficher les résultats des opérations suivantes : addition, soustraction, multiplication, division, exponentiation, et division entière.
9. **Opérateurs avancés** : Écrivez un programme qui utilise les opérateurs bit à bit (&, |, ^, », «) pour manipuler des nombres binaires. Le programme doit demander à l'utilisateur d'entrer deux nombres entiers et afficher les résultats des opérations bit à bit.
10. **Notations numériques** : Créez un script Python qui convertit des nombres entre différentes notations (exponentielle, binaire, octale, hexadécimale). Le programme doit demander à l'utilisateur d'entrer un nombre dans une notation spécifique et afficher les équivalents dans les autres notations.
11. **Calculatrice de base** : Développez une calculatrice simple en Python qui peut effectuer les opérations de base (addition, soustraction, multiplication, division). Le programme doit offrir une interface utilisateur simple via la ligne de commande.
12. **Simulation de jeu de dés** : Écrivez un programme qui simule le lancer de deux dés. Le programme doit afficher les résultats de chaque lancer et compter combien de fois chaque valeur (2 à 12) apparaît après un certain nombre de lancers définis par l'utilisateur.
13. **Jeu du pendu** : Créez un jeu du pendu en Python. Le programme doit choisir un mot aléatoire dans une liste de mots, demander à l'utilisateur de deviner des lettres, et afficher l'état actuel du mot et le nombre d'essais restants.