

Atelier_3

N'oubliez pas d'utiliser des commentaires pour expliquer le fonctionnement de chaque partie de votre code.

Les exceptions en Python

Activité 1

Écrivez un programme qui demande à l'utilisateur de saisir deux nombres. Le programme doit afficher le résultat de la division du premier nombre par le deuxième nombre. Cependant, le programme doit également gérer les exceptions possibles :

Si l'utilisateur entre des valeurs qui ne sont pas des nombres, le programme doit afficher un message d'erreur et redemander les valeurs.

Si l'utilisateur entre 0 comme deuxième nombre, le programme doit afficher un message d'erreur indiquant que la division par zéro n'est pas autorisée et redemander le deuxième nombre.

Si une autre erreur se produit lors du calcul de la division, le programme doit afficher un message d'erreur générique

La gestion des fichiers

Activité 1

Créez un fichier texte nommé "input.txt" contenant une liste de nombres séparés par des virgules.

Écrivez un programme Python qui lit les nombres à partir du fichier "input.txt", les additionne, soustrait 5 à chaque nombre et calcule le carré de chaque résultat.

Sauvegardez les résultats des opérations dans un nouveau fichier texte nommé "output.txt", chaque résultat étant sur une nouvelle ligne.

Activité 2

Écrivez un programme qui demande à l'utilisateur de saisir le nom d'un fichier texte. Le programme doit ouvrir le fichier, lire son contenu et afficher le nombre de lignes qu'il contient. Cependant, le programme doit également gérer les exceptions possibles :

Si le fichier spécifié n'existe pas, le programme doit afficher un message d'erreur indiquant que le fichier n'a pas été trouvé.

Si une autre erreur se produit lors de la lecture du fichier, le programme doit afficher un message d'erreur générique.

Les procédures et fonctions

Activité 1

Écrivez une fonction appelée "somme_pairs" qui prend en entrée une liste d'entiers et renvoie la somme des nombres pairs présents dans cette liste.

Activité 2

Écrivez deux fonctions en Python pour calculer l'aire et le périmètre d'un rectangle en fonction de sa longueur et de sa largeur. Voici les étapes à suivre :

Créez une fonction appelée `calculer_aire_rectangle` qui prend en entrée deux paramètres longueur et largeur et renvoie l'aire du rectangle calculée en multipliant la longueur par la largeur.

Créez une fonction appelée `calculer_perimetre_rectangle` qui prend en entrée deux paramètres longueur et largeur et renvoie le périmètre du rectangle calculé en utilisant la formule : $\text{périmètre} = 2 * (\text{longueur} + \text{largeur})$.

Testez vos fonctions en appelant chacune d'elles avec différentes valeurs de longueur et de largeur, puis affichez les résultats.

Activité 3

Ecrivez un script qui crée un mini-système de base de données fonctionnant à l'aide d'un dictionnaire, dans lequel vous mémoriserez les noms d'une série des élèves, leur âge et leur taille.

Dans le dictionnaire, le nom de l'élève servira de clé d'accès, et les valeurs seront constituées de tuples (âge, taille), dans lesquels l'âge sera exprimé en années (donnée de type entier), et la taille en mètres (donnée de type réel).

Votre script devra comporter deux fonctions :

- La fonction `remplissage()` qui permet de remplir le dictionnaire, les données du dictionnaires seront saisies par l'utilisateur.
- La fonction `consultation()`. Dans la fonction de consultation permet de consulter les information, le couple (âge, taille) relatives au nom de l'élève fournit par l'utilisateur.

Le résultat de la requête devra être une ligne de texte bien formatée.

Activité 4

Écrivez une fonction nommée "mediane" qui prend en entrée une liste de nombres et renvoie la médiane des éléments de la liste. (Indice : pour une liste de longueur impaire, la médiane est l'élément du milieu une fois la liste triée ; pour une liste de longueur paire, la médiane est la moyenne des deux éléments du milieu.)

Testez cette fonction avec différentes listes de nombres pour vous assurer qu'elles fonctionnent correctement

Activité 5

Écrivez un programme Python qui effectue les opérations suivantes sur une liste de mots:

- c. Définir une fonction nommée "inversion_mots" qui prend une liste de mots en entrée et renvoie une nouvelle liste contenant les mots inversés. Par exemple, "hello" deviendra "olleh".
- d. Définir une fonction nommée "mot_palindrome" qui prend une liste de mots en entrée et renvoie une nouvelle liste contenant uniquement les mots qui sont des palindromes (c'est-à-dire des mots qui restent les mêmes lorsqu'ils sont lus à l'envers).

Testez toutes les fonctions avec différentes listes de mots pour vous assurer qu'elles fonctionnent correctement.

Activité 6 (lambda)

Écrivez une fonction nommée "calculatrice" qui prend deux nombres et une chaîne de caractères représentant une opération mathématique ('+', '-', '*', '/'). Utilisez une expression lambda avec une série de conditions if-elif-else pour effectuer le calcul approprié en fonction de l'opération fournie. Renvoyez le résultat du calcul.

Bon travail