

Python – TP2 : Débogage – 2h

I Les outils de base

Exercice 1 : Exercice 1 : Fonction carré

Téléchargez et ouvrez dans Spyder le fichier “exo1-carre.py”. Il ne contient aucun bug et fonctionne très bien. Le but de l’exercice est simplement de prendre en main les principaux outils du débogueur que nous allons utiliser.

Lancez le programme et constatez qu’il produit le résultat attendu, sans renvoyer d’erreur.

Lancez le programme en mode debug (Ctrl + F5). Affichez, à droite, le volet Explorateur de variables. Exécutez les lignes à l’aide des différents “pas-à-pas” proposés. Essayez de bien déceler les différences et intérêts de chacune des méthodes. Par la suite, il ne sera plus précisé lequel utiliser. A vous de voir ce qui vous semblera judicieux.

Placez un “point d’arrêt” en double cliquant dans la marge (ou depuis le menu “Déboguer”) en face d’une ligne contenant une instruction, par exemple la ligne 2. Démarrez de nouveau en mode debug. Vous constatez que l’exécution se lance jusqu’à atteindre le point d’arrêt (cela est bien visible dans la console). Vous pouvez ensuite continuer en pas-à-pas ou en utilisant “Continuer” (Ctrl + F12) qui vous emmènera au prochain point d’arrêt rencontré.

Les points d’arrêt permettent de déboguer en se concentrant uniquement sur les passages intéressants. On peut placer autant de points d’arrêts que l’on souhaite.

Lors de la consultation des variables pendant un debug, prenez une capture d’écran (de tout l’écran) que vous enregistrez dans un fichier image (JPG ou PNG) dont vous connaissez l’emplacement. Assurez-vous de savoir faire cette manipulation car elle sera utilisée lors de l’examen de TP pour vérifier que vous savez effectivement utiliser le débogueur. Il vous sera demandé de faire cette capture lors d’un état bien précis du déroulement du programme en mode debug afin que l’on observe les valeurs de variables.

On peut aussi placer maintenant un point d’arrêt conditionnel. Il suffit d’y écrire une condition logique et le débogage ne s’arrête à ce point que si la condition est vérifiée. Les points d’arrêts conditionnels sont extrêmement pratiques lorsqu’il s’agit de déboguer à la fin d’une boucle ayant beaucoup d’itérations.

II Cas pratiques

Dans tous les exercices qui suivent, l’énoncé de l’exercice ne fait que décrire le comportement attendu ou le rôle du programme.

Le but est de chasser le ou les bugs. Il faut les trouver et les corriger. Il n’est pas

demandé de réécrire entièrement le programme pour obtenir le bon résultat mais plutôt de corriger de manière propre et économique.

Il est fortement conseillé (même si ce n'est pas utile pour que le code fonctionne) de renommer les variables dont le nom est insuffisamment explicite. Cela vous aidera dans la compréhension. La fonctionnalité de remplacement, dans le menu "Recherche" vous aidera dans cette tâche. Attention, cependant, quand il s'agit de renommer "a" en "aire"... Tous les caractères "a" de votre code seront remplacés. Dans ce genre de cas, il vaut mieux éviter d'utiliser l'option "Replacer tout".

Exercice 2 : Factorielle

Le programme doit renvoyer la valeur de $5!$, qui vaut 120.

Exercice 3 : Noms complets

A partir d'une liste de prénoms et d'une liste de noms, le programme crée une liste de "prénom nom".

Exercice 4 : Aire minimale

On dispose d'une liste de rectangles dont les côtés sont connus. Ils se trouvent dans une liste de longueurs et une liste des largeurs. L'objectif est la recherche de l'aire minimale.

Exercice 5 : Liste croisée

Écrivez une fonction qui prend en entrée deux listes L1 et L2 et qui renvoie en sortie une nouvelle liste composée de la manière suivante :

- Le premier terme est le premier de L1. Le deuxième est le premier de L2. Le troisième est le deuxième de L2. Etc.
- On continue d'alterner les termes jusqu'à l'épuisement d'une des deux listes.
- On complète en ajoutant les termes restant de l'autre liste.

Exercice 6 : Voyelles

Écrivez une fonction qui prend en entrée une chaîne de caractères et qui renvoie en sortie cette chaîne où toutes les voyelles (a, e, i, o, u, y) sont remplacées par des y.

Après avoir trouvé l'erreur, réécrire ce programme de manière plus efficace et plus facile à entretenir (imaginons que l'on veuille par la suite ajouter des caractères à considérer comme voyelles, tels que é et è).

Exercice 7 : Suite de Conway

La suite de Conway a pour premier terme 1. Les termes suivants sont construits en lisant le terme précédent (nombre de chiffre, quel chiffre). Les termes suivants sont donc 11 (car un 1), puis 21 (car deux 1), puis 1211 (car un 2, un 1), etc.

Le programme affiche les 11 premiers termes de la suite de Conway.

Exercice 8 : Comptage de mots

Le fichier “data.txt” contient un texte.

1. Lire le fichier et récupérer le texte dans une variable.
2. Séparer ce texte de manière à avoir une liste de mots : ['À', 'la', 'fin', 'des',...]
3. Créer un dictionnaire dont les clés sont les mots et les valeurs le nombre de fois ou chacun de ces mots apparaît dans le texte
4. Afficher le mot apparaissant le plus de fois ainsi que son nombre d'apparitions.

Exercice 9 : Récurrence à deux termes

On souhaite calculer les termes u_0, \dots, u_{12} de la suite définie par

$$u_0 = 0, \quad u_1 = 1, \quad u_n = 2u_{n-1} - \frac{4}{3}u_{n-2}.$$