

Initiation à Python - leçon 4.1

s1

Dans cette séquence, nous allons découvrir les exceptions. Très généralement, une exception est un mécanisme qui permet d'anticiper les problèmes survenant au cours du déroulement d'un programme.

s2

Quand une erreur se produit dans un script - par exemple une division par zéro ou le fait que le programme ne trouve pas le fichier texte qu'il était censé ouvrir - elle provoque l'arrêt du programme et l'affichage d'un message d'erreur.

Python propose un mécanisme simple pour anticiper ce genre de situation et mettre en place un traitement spécifique. Plutôt que de parler d'erreur, on emploiera le terme "exception". Prévoir une réaction adaptée à une exception, c'est la "rattraper".

D'ores et déjà un peu de vocabulaire

- Essayer d'exécuter (try) une partie bien définie de code permet d'initialiser le mécanisme
- Capturer ou attraper une éventuelle exception associée à ce code (except) permet de rattraper une erreur

s3

Voici un exemple mettant en oeuvre la forme la plus simple de ce mécanisme.

Dans cet exemple, on demande à l'utilisateur d'entrer un nombre au clavier. Après avoir transformé ce nombre y en flottant, on effectue la division $1/y$. Le fait que l'utilisateur puisse entrer zéro est géré ici par un mécanisme d'exception qui provoque l'affichage d'un message. Bien entendu, dans cet exemple simple, on aurait pu tester y par un simple if avant de faire la division $1/y$.

Pour se convaincre que le mécanisme d'exception n'interrompt pas l'exécution, je vous invite à reproduire également l'exemple suivant. Dans cet exemple, une boucle while est lancée tant que la valeur entrée au clavier pour y n'est pas correcte.

s4

Voici un autre exemple élémentaire avec un programme cherchant à ouvrir un fichier qui n'existe pas.

s5

On peut également prévoir un traitement particulier pour des exceptions prédéfinies. Dans ce cas, on utilisera une construction un peu plus élaborée représentée ici. Python possède beaucoup d'exceptions prédéfinies. En voici juste quelques-unes.

Voici un exemple de l'utilisation de l'exception prédéfinie `SyntaxError`. Dans cet exemple on demande à l'utilisateur d'entrer une expression numérique. Grâce à la fonction intrinsèque `eval()`, Python va chercher à évaluer cette expression, transformer le résultat en flottant et l'afficher.

Ainsi, si l'on entre l'expression `2+7`, Python affichera `9` ; si l'on entre l'expression `2?7`, Python rencontrera une erreur et lèvera l'exception de Type `SyntaxError`.

s6

Voici un exemple mettant en jeu un ensemble d'exceptions prédéfinies ainsi que la clause `else`, exécutée si aucune exception n'a été levée.

s7

Pour clore cette leçon sur les exceptions, sachez qu'une exception peut être levée manuellement grâce à la fonction `raise`, comme dans l'exemple suivant.

s8

Voici pour les exceptions. Dans la leçon suivante, nous allons introduire quelques compléments très utiles lorsque l'on travaille avec des listes : les itérateurs et les générateurs.