

PROGRAMMATION PYTHON

Chapitre 7: Exceptions



Sommaire

1. [Introduction](#)
2. [Exercices](#)

Introduction

Tous les programmes écrits dans les chapitres précédents fonctionnent uniquement dans le cas où l'utilisateur entre une valeur attendue. Si le programme va générer une erreur.

Supposons le programme suivant qui demande à l'utilisateur de saisir un nombre et calcule le carré du nombre:

```
>>> # Fonction qui demande un nombre à l'utilisateur et retourne le carré
>>> def saisieEtCalculCarre():
...     nbre = input('Entrer un nombre:')
...     nbre = int(nbre)
...     return nbre * nbre
...
>>> # Exécution avec saisie d'un nombre
>>> saisieEtCalculCarre()
Entrer un nombre:2
4
>>> # Exécution avec saisie de tout sauf un nombre
>>> saisieEtCalculCarre()
Entrer un nombre:a
Traceback (most recent call last):
  File "<pyshell#19>", line 1, in <module>
    saisieEtCalculCarre()
  File "<pyshell#15>", line 3, in saisieEtCalculCarre
    nbre = int(nbre)
ValueError: invalid literal for int() with base 10: 'a'
```

On voit que Python génère une erreur quand on ne fournit pas un nombre comme attendu. Tout code écrit après ne sera pas traité et le programme s'arrête là dès qu'une erreur est rencontrée.

Introduction

Les erreurs comme celle vue précédemment peuvent être interceptées très facilement avec Python. Quand Python rencontre une erreur lors de l'exécution du code, il lève une **exception** comme ci-dessous:

```
Traceback (most recent call last):  
  File "<pyshell#19>", line 1, in <module>  
    saisieEtCalculCarre()  
  File "<pyshell#15>", line 3, in saisieEtCalculCarre  
    nbre = int(nbre)  
ValueError: invalid literal for int() with base 10: 'a'
```

Le code précédent a échoué parce que nous n'avons pas préparé notre code à faire face à ce genre d'erreur. Quand Python tombe sur une exception, il exécute le code qui doit être exécuté lors d'une exception sinon il affiche une erreur comme précédemment.

Chaque exception est identifiée par deux informations:

- le type de l'exception dans notre cas **ValueError**
- le message qu'envoie Python pour aider à comprendre l'erreur qui vient de se produire.

Il faut noter que lors de la levée d'une exception, Python arrête l'exécution du programme si aucun code n'a été fourni en cas d'erreur. Quand un code est fourni en cas d'exception on parle d'**interception de l'exception**. Par exemple dans l'exemple précédent on aurait pu redemander un autre nombre par l'utilisateur quand il saisit autre chose qu'un nombre.

Exercices

Exercise 1

ssdsds

FIN CHAPITRE 7