

Python : gérer les exceptions

Présentation

Lors de l'**exécution** d'un programme Python, certaines instructions ou expressions qui sont **syntactiquement correcte**, peuvent **générer une erreur**: ce sont des **exceptions**. En effet, ces erreurs ne sont pas toujours fatales et le programmeur peut anticiper leur éventuelle survenue en prévoyant comment les traiter.

Si vous **ne prévoyez pas** de gérer les exceptions, cela entraînera l'**arrêt immédiat** du programme. La gestion des exceptions permet d'avoir un programme mieux fini.

Générer une exception

Voici un exemple de gestion d'une exception.

- Saisie d'un nombre par l'utilisateur : `<code python> # saisie d'une chaîne de caractère nombre = input("Saisissez un nombre :") # conversion de la chaîne de caractères en nombre nombre = int(nombre) </code>`
- Ces deux instructions s'exécutent sans **aucun problème** si l'utilisateur saisit un **nombre**.
- Si l'utilisateur saisit autre chose qu'un nombre, des **lettres** par exemple, la deuxième instruction, celle qui permet de convertir la saisie en un nombre, **génère une exception**. `<code> Saisissez un nombre : aaaaa Traceback (most recent call last): File "D:\Development\python\scripts\exception.py", line 2, in <module> nombre = int(nombre) ValueError: invalid literal for int() with base 10: 'aaaaa'</code>`

```
<
/code>
```

Gérer une exception

L'instruction qui a généré une exception est l'instruction **int()**. Voici comment gérer l'exception :

```
nombre = input("Saisissez un nombre : ")
try:
    nombre = int(nombre)
except:
    print("Erreur, ce n'est pas un nombre.")
```

- voici ce que donne le 'exécution du programme si on saisit une chaîne de caractères

```
Saisissez un nombre : aaaa
Erreur, ce n'est pas un nombre.
>>>
```

Explications :

- le bloc d'instructions qui peut générer une exception se met après le **mot-clé try** suivi de : (on indique un bloc d'instruction)
- le bloc d'instruction à exécuter en cas d'exception est mis après le **mot-clé except** suivi de :

De cette manière on **capture** toutes les exceptions possibles qui peuvent être générées par l'instruction **int()**. Il est possible de préciser l'exception que l'on souhaite gérer c'est à dire celle qui est liée à la conversion de la manière suivante :

```
nombre = input("Saisissez un nombre : ")
try:
    nombre = int(nombre)
except ValueError:
    print("Erreur, ce n'est pas un nombre.")
```

Else

Le bloc **except:** peut aussi être complété par un bloc **else:** s'il aucune exception ne se produit :

```
nombre = input("Saisissez un nombre : ")
try:
    nombre = int(nombre)
except ValueError:
    print("Erreur, ce n'est pas un nombre.")
else:
    print("Bien vous avez saisi un entier")
```

Finally

finally est un bloc qui est exécuté **après** que tous les autres blocs aient été exécutés, peu importe qu'il y ait eu une exception ou non.

```
nombre = input("Saisissez un nombre : ")
try:
    nombre = int(nombre)
except ValueError:
    print("Erreur, ce n'est pas un nombre.")
else:
    print("Bien vous avez saisi un entier")
finally:
    print("Nous continuons le programme")
```

L'affichage **"Nous continuons le programme"** sera toujours affiché.

Retour au cours : Les instructions du langage Python

- [Cours : Les instructions du langage Python](#)

From:

/ - Les cours du BTS SIO

Permanent link:

/doku.php/icn/facultatif/c_langage_python_exception

Last update: 2018/04/08 14:20

