Chapitre # (ANN) 1 Erreurs courantes en Python

Résumé & Plan

L'objectif de ce chapitre est le suivant : vous permettre d'analyser et de comprendre les erreurs que vous obtenez en Python. « Monsieur, ça ne marche pas » n'est à prononcer qu'après l'analyse de l'erreur.

Avec Python, lorsque le lancement d'une instruction ou d'un programme provoque une erreur, un message s'affiche en rouge. Ce message vous informe :

- de l'endroit où l'erreur s'est produite dans le programme (numéro de la ligne dans le programme),
- de la cause de l'erreur.

Concernant cette dernière information, le message paraîtra parfois obscur pour un non-anglophone. Il est donc important que vous puissiez comprendre par vous-même ce qu'il faut corriger à votre programme pour qu'il tourne correctement.

- `SyntaxError` regroupe toutes les situations où ce que vous avez demandé n'a pas de sens en Python. Cela peut venir d'une parenthèse en trop, d'un crochet au lieu d'une parenthèse, d'une virgule au lieu d'un point …. Exemple :
```
((1+2]
```
- `IndentationError: unexpected indent` signifie qu'il y a un problème d'alinéa. En général, une ou plusieurs lignes ne sont pas alignées correctement avec leur bloc. Il suffit d'un espace en trop !
```
x = 1 # BIEN
  x = 1 # MAUVAIS
```
- `IndexError: list index out of range` signifie que vous avez demandé d'extraire un élément d'une liste avec une position plus grande que la taille de la liste.
```
L = [2,3,9]
L[5]
```
- `TypeError: list indices must be integers or slices, not float` même chose mais si vous demandez une position qui n'est pas un nombre entier.
```
A = [1, 8, 4]
A[3.3]
```
- `NameError: name 'BCPST' is not defined` signifie que vous demandez à Python d'utiliser une variable alors que celle-ci n'existe pas. Souvent, ça arrive quand on a mal recopié le nom de notre variable.
```
BCPST *= 2
BCPST *= 8
```
Ca peut arriver aussi si on utilise une fonction d'une bibliothèque qu'on a oublié d'importer au préalable. Par exemple si on fait :
```
a = cos(pi)
# sans avoir fait from math import *
```
- `ZeroDivisionError: float division by zero` assez facile à comprendre … vous avez demandé de faire une division par zéro.
- `TypeError: ... object is not callable` en général signifie que vous avez pris pour une fonction une variable qui n'en était pas une. *To call a function* signifie « exécuter une fonction ». Seules les variables de type **function** peuvent être exécutées, ceci avec la syntaxe `f(x)`. Par conséquent si `f` n'est pas une fonction ça fait une erreur.
```
a = 1
a(3)
```
Cela arrive aussi si vous avez oublié un signe * pour multiplier une parenthèse. Exemple :
```
2(3+8)
```
- `TypeError: ... object is not subscriptable ...` Même chose avec les crochets. La syntaxe `L[2]` sert à extraire l'élément en position 2 de `L`. Quand `L` n'est pas une liste ou une chaîne de caractère cela posera problème. Exemple :
```
L = 3
L[2]
```
- `OSError: [Errno 2] No such file or directory: 'C:/photos/vacances_Blanquer/ibiza.jpg'`

ceci arrive quand vous cherchez à importer un fichier qui n'existe pas. Vérifiez

alors que votre fichier est bien situé là où vous croyez, et qu'il n'y a pas d'erreur dans l'orthographe des dossiers, sous-dossiers, du fichier.