

## Gérer les exceptions

### Exercice 1 Vérification de l'âge

Écrire un programme qui demande à l'utilisateur d'écrire son âge et qui gère le cas où celui-ci n'entre pas un nombre.

### Exercice 2 Fonction définie par morceau

Écrire une fonction `sinc` qui prend en entrée un nombre  $x$  et renvoie la valeur  $\frac{\sin(x)}{x}$  si  $x \neq 0$  et 1 sinon.

Le programme doit gérer une exception. La fonction `sin` est disponible dans le module `math` de Python.

### Exercice 3 Ouverture de fichier

Lorsque l'on essaie d'ouvrir un fichier qui n'existe pas avec la fonction `open` de Python, l'interpréteur nous renvoie une exception (mal définie en Python < 3 et `IOError` en Python3) Écrire un programme qui demande à l'utilisateur le nom d'un fichier et essaie de le lire. Si le fichier n'existe pas, le programme affiche qu'il n'existe pas et redemande à l'utilisateur un autre nom de fichier.

### Exercice 4 Un jeu

Voici le début d'un jeu très simple consistant à arrêter une boucle sur un nombre précisé à l'avance :

```
1#!/usr/bin/python
2#coding: utf-8
3import random
4import time
5x = random.randint(0,100)
6print 'Vous devez arrêter le programme sur %d'%x
7print 'Pour arrêter le programme, il faut faire Ctrl+C.'
8print 'Appuyer sur Entrée pour commencer ...'
9b = raw_input()
10i=0
11while (i<100) :
12    i+=1
13    print "%d\x0d"%i ,
14    time.sleep(0.2)
```

En gérant l'exception `KeyboardInterrupt`, compléter le programme pour qu'il réponde aux spécifications.

**Remarque :** Le code ASCII « `\x0d` » est le caractère retour chariot donné comme code hexadécimal.

## Lever des exceptions

### Exercice 5 Pente d'une droite

Nous désirons écrire un programme qui calcule la pente d'une droite passant par deux points dont nous connaissons les coordonnées.

- 1 ▶ Écrire une fonction `pente(xA, yA, xB, yB)` qui utilise la formule mathématique et renvoie la pente de la droite (AB).
- 2 ▶ Modifier le programme dans le cas où l'utilisateur commet une division par zéro en gérant l'exception associée.

### Exercice 6 Racines réelles

Nous désirons écrire un programme qui calcule les racines réelles d'un polynôme et qui lève une exception «`NameError`» avec un message adapté lorsque le discriminant est strictement négatif.

- 1 ▶ Écrire une fonction `racines(a, b, c)` qui utilise la formule mathématique pour renvoyer une valeur approchée des racines réelles.
- 2 ▶ Tester ce programme avec un discriminant négatif puis avec un argument  $a$  égal à 0. Gérer en conséquence les exceptions (et les changements de formules!)

### Exercice 7 Addition en temps limité

- 1 ▶ Écrire un programme qui demande à l'utilisateur de calculer la somme de deux nombres (pris aléatoirement entre 0 et 100).
- 2 ▶ En utilisant le module `time`, lever une exception lorsque l'utilisateur ne répond pas après 10 secondes.  
*Conseil :* on utilisera une boucle `while`.

### Exercice 8 Fonctions récursives ★

Voici une fonction récursive qui

```
def fibonacci(n):
2     if n == 0 or n == 1:
3         return 1
4     else:
5         return fibonacci(n-1)+fibonacci(n-2)
```

- 1 ▶ Modifier le programme pour qu'il renvoie le nombre d'appel récursifs nécessaires au calcul de `fibonacci(n)`.  
*Conseil :* on pourra ajouter un argument à `fibonacci`.
- 2 ▶ Modifier le programme pour qu'il lève une exception lorsque le niveau de récursivité dépasse 100.