

## Programmer avec Python

### Exercices

Séance n° 5

### Les modules

**Objectif :** Réalisez chacun des exercices en utilisant le logiciel Spyder comme il a été expliqué en cours. Enregistrez bien votre travail au fur et à mesure !

## 1 Aire d'un cercle

1. Écrivez une fonction prenant en entrée le rayon d'un cercle et retourne son aire. La valeur calculée devra être arrondie à l'entier supérieur. Pour cela, utilisez la fonction `ceil(x)` du module `math`.
2. Écrivez un programme pour tester votre fonction

## 2 Simuler un lancer de dé

Le module `random` permet de générer des nombres entiers au hasard grâce à la fonction `randint`.

```
import random
nombre = random.randint(0,100) # Tire un entier au hasard entre 0
    et 100
print(nombre)
```

Écrivez un programme Python qui tire au hasard un entier compris entre 1 et 6 un grand nombre de fois (100 fois pour commencer par exemple). Le programme affichera ensuite le nombre de fois où le 1 a été tiré, le 2, le 3, le 4, et ainsi de suite pour chaque valeur de 1 à 6. Affichez également le pourcentage obtenu pour chaque face. Répétez l'exécution du programme en faisant 1000 tirages, 10000 tirages. Qu'en pensez-vous ?

Note : Il y a plusieurs façons possibles d'écrire un tel programme, pensez à une structure que vous avez étudié à la séance 2 qui vous simplifiera la tâche...

## 3 Horloge parlante

1. Écrivez des fonctions permettant de retourner :
  - La date au format `dd-mm-aaaa`
  - L'heure au format `hh:mm`
  - La date complète au format `dd-mm-aaaa hh:mm`
2. Écrivez ensuite un programme proposant à l'utilisateur un menu lui demandant de choisir s'il veut afficher :
  - la date d'aujourd'hui
  - L'heure courante
  - Les deux ensemble

## 4 Créer son module

Reprenez l'exercice 1 et créez un module comportant la fonction écrite pour calculer l'aire d'un cercle et un fichier comportant le programme principal. Testez bien !

## 5 Countdown

Pour cet exercice, vous allez utiliser un module que j'ai préparé moi-même pour créer des timers.

1. On va commencer par récupérer ce module. Téléchargez le fichier `repeat.py` disponible sur l'ENT sur la page du cours dans la partie Ressources. Ou suivez directement ce lien : <https://ent.esigelec.fr/mod/resource/view.php?id=10881>. Enregistrez le fichier dans le répertoire de la séance. Le module contient une classe `Repeat` qui permet de créer des timers. Un objet `Repeat` répètera l'exécution d'une fonction donnée en paramètre toutes les X secondes.
2. Voici un exemple. Reprenez ce code et testez-le :

---

```
from repeat import Repeat

#Une fonction qui dit bonjour
def hello():
    print("Hello, World!")

# Un timer qui répète la fonction hello toutes les secondes
t = Repeat(1.0, hello)
# On démarre le timer
t.start()
```

---

Vous pouvez arrêter le timer en tapant `t.cancel()` dans la console de Spyder (ou en la fermant avec la petite croix directement!).

3. A vous. Écrivez une fonction qui calcule la durée restante entre maintenant et le début des vacances de Noël. Vous afficherez la valeur obtenue en jours, heures, minutes, secondes (comme s'affiche un delta entre 2 dates sans ne rien faire de plus!)
4. Écrivez un programme principal qui affiche le temps restant jusqu'aux vacances de Noël, toutes les 3 secondes, en utilisant un objet `Repeat` comme dans l'exemple.

## 6 Apprendre à lire de la doc !

Vous commencez à comprendre que les modules sont très nombreux, et leur utilisation très variée. Il est impossible de connaître tous les modules Python existants, et encore moins leur utilisation. L'arme du développeur est donc : la documentation. Souvent disponible seulement en anglais, c'est elle qui vous permettra de savoir ce que permet un module et comment l'utiliser. Vous allez donc vous entraîner dans cet exercice à prendre en main un nouveau module à l'aide de sa doc : le module `csv`.

1. Vous avez à disposition un fichier de résultats de l'édition 2017 de l'UTMB dans les ressources du cours sur l'ENT : <https://ent.esigelec.fr/mod/resource/view.php?id=10886>. Téléchargez-le, ouvrez le avec Excel, regardez. Ce fichier est au format CSV, les colonnes y sont séparées par des ; ou des tabulations. Le fichier comporte à chaque ligne un coureur avec ses colonnes : son classement, son nom, son temps...
2. À l'aide de la documentation Python du module CSV (en français, vous êtes chanceux) ici : <https://docs.python.org/fr/3/library/csv.html>, écrivez un programme qui lit ce fichier CSV et affiche chaque ligne.

## 7 Apprendre à lire de la doc !\*

On continue l'exercice !

1. Complétez votre programme pour qu'il affiche le nombre de coureurs par catégorie.
2. Générez un nouveau fichier CSV ne contenant que les coureurs espagnols.

## 8 Beaucoup beaucoup de modules\*

De la même façon que dans l'exercice précédent, choisissez dans la liste suivante une tâche à accomplir avec le module donné et aidez-vous de la documentation pour réaliser le travail demandé :

- Redimensionner une image GIF (de votre choix) en taille 64x64 avec le module Pillow : <https://pillow.readthedocs.io/en/stable/>
- Demander à un utilisateur de saisir deux matrices de dimension 2x2 et lui afficher leur somme et leur produit grâce à Numpy : <https://numpy.org/devdocs/user/index.html>
- Afficher un diagramme en bâtons montrant le nombre de coureurs par catégorie dans nos résultats de l'UTMB avec Matplotlib : <https://matplotlib.org/stable/users/index.html>