

Informatique PCSI

TP 3 : modules et fichiers

1 Énoncé des exercices

1.1 Exercice 1

On dispose de données écrites dans un fichier nommé "data.csv". La première ligne contient le texte `largeur;longueur;hauteur`. Chaque ligne suivante du fichier contient trois valeurs, qui représentent des flottants, séparées par des virgules, par exemple `25.3, 37.5, 13.8`.

1. Écrire un code permettant de lire les lignes de données et de stocker l'ensemble des lignes dans une liste dont chaque élément est une liste constituée par les trois valeurs de type float d'une ligne.
2. Modifier le code pour ne récupérer qu'une ligne de données sur dix (la première ligne de données, la onzième, etc).

Pour garder une ligne sur dix, penser à une expression comme `i % 10`.

1.2 Exercice 2

Création d'un module

On considère un fichier `dessins.py` qui contient le code suivant :

```
from turtle import *
import figures as f

up()
goto(50, -60)
down()
f.rectangle(100, 75, 'red')

up()
goto(-80, -20)
down()
f.triangle(85, 'blue')

up()
goto(20, 60)
down()
f.cercle(60, 'green')
```

Il s'agit d'écrire le fichier `figures.py` qui est importé dans le programme ci-dessus. Lorsque le programme est exécuté, on obtient dans la fenêtre graphique du module Turtle un rectangle, un triangle équilatéral et un cercle avec des couleurs différentes.

Les principales commandes du module Turtle sont résumées à ces deux adresses :

https://fr.wikibooks.org/wiki/Programmation_Python/Turtle

<https://docs.python.org/fr/3.8/library/turtle.html>.

Procéder pas à pas en écrivant une fonction simple et en la testant pour améliorer le code avant de passer à la suite. Quand toutes les fonctions sont correctes, tester alors le module.

1.3 Exercice 3

Image bitmap.

Le format PBM est l'un des plus simples pour définir une image en noir et blanc. Il s'agit d'un fichier texte dont le contenu est :

- ▶ sur la première ligne, P1 qui définit le format ;
- ▶ sur la ligne suivante, facultative, un commentaire qui commence par le symbole # ;
- ▶ sur la ligne suivante, la largeur et le hauteur de l'image en pixel ;
- ▶ sur les lignes suivantes, des 0 et des 1 qui représentent chacun un pixel en blanc ou en noir.

Par exemple :

```
P1
# Un exemple d'image bitmap
10 12
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 0 0 0 0
0 0 0 1 1 1 1 0 0 0
0 0 0 1 1 1 1 0 0 0
0 0 1 1 0 0 1 1 0 0
0 0 1 1 0 0 1 1 0 0
0 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 0
1 1 0 0 0 0 0 0 1 1
1 1 0 0 0 0 0 0 1 1
0 0 0 0 0 0 0 0 0 0
```

Un fichier est enregistré avec l'extension `pbm`, comme `image.pbm`. Pour visualiser l'image, il suffit de l'ouvrir avec un logiciel comme XnView ou GIMP.

Une image est représentée par une matrice, une liste de listes. Pour l'exemple ci-dessus, on obtient :

```
[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
 [0, 0, 0, 0, 1, 1, 0, 0, 0, 0], [0, 0, 0, 1, 1, 1, 1, 0, 0, 0],
 [0, 0, 0, 1, 1, 1, 1, 0, 0, 0], [0, 0, 1, 1, 0, 0, 1, 1, 0, 0], ...]
```

On définit la matrice d'une image exemple avec la fonction suivante :

```
def matrice(n, p):
    m = [[0 for j in range(p)] for i in range(n)]
    for i in range(n): # n est la hauteur de l'image
        for j in range(p): # p est la largeur de l'image
            if (j // 20) % 2 == 0:
                m[i][j] = 1
    return m
```

Écrire une fonction `fichier` qui prend en paramètres une matrice comme celle définie ci-dessus, et une chaîne de caractères représentant le nom d'un fichier, sans l'extension. La fonction crée le fichier image, avec l'extension `pbm`, et y écrit les lignes définissant l'image avec les valeurs données par la matrice.

Après avoir créer une matrice avec l'instruction `m1 = matrice(240, 180)`, entrer l'instruction `fichier(m1, "image1")` et vérifier qu'un fichier `image1.pbm` a été créé. Visualiser l'image avec par exemple le logiciel XnView ou le logiciel GIMP.