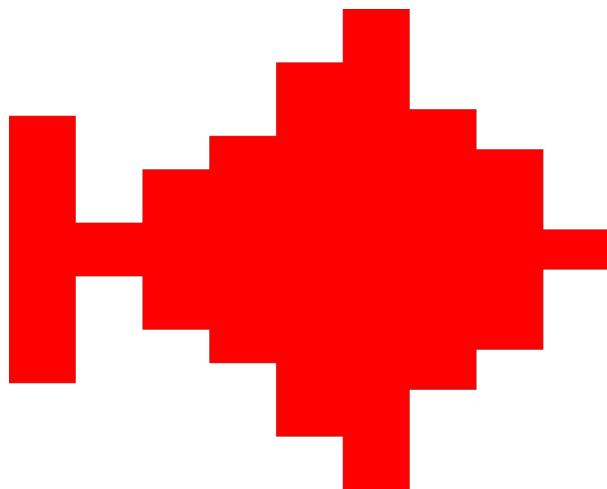


Exercices - Semaine 3

1. Poisson rouge

Complétez le code ci-dessous afin d'obtenir cette magnifique image de poisson rouge (un tantinet abstraite) :



```
from pytamaro import *

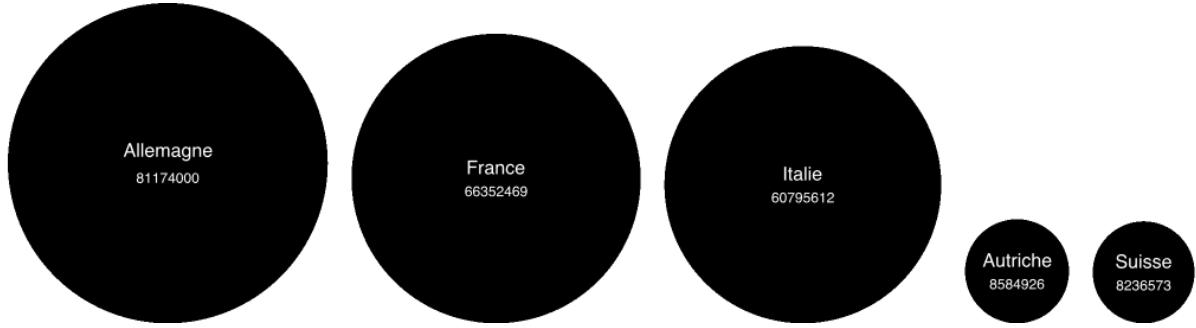
def barre(hauteur):
    return rectangle(50, hauteur, red)

hauteurs = (200, 40, 120, 170, 280, 360, 210, 150, 30)

forme = empty_graphic()
```

2. Population des pays limitrophes

Complétez le code ci-dessous afin d'obtenir la visualisation suivante :



```
from pytamaro import *

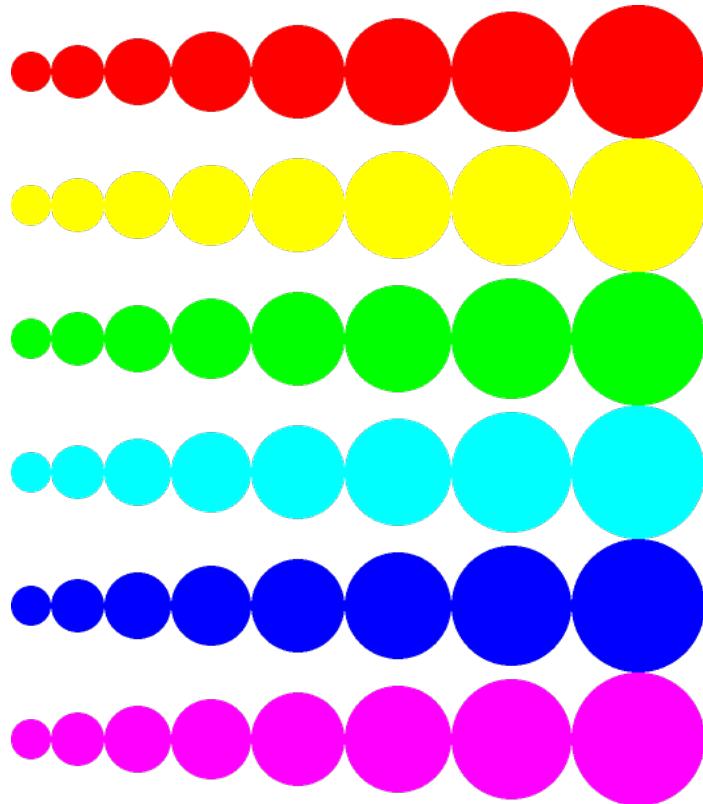
def zone(nom, valeur):
    diametre = (valeur ** 0.5) / 1000 * 30
    back = ellipse(diametre, diametre, black)
    back = overlay(back, rectangle(diametre + 20, diametre, transparent))
    label = text(nom, "Helvetica", 16, white)
    label = above(label, rectangle(0, 10, transparent))
    label = above(label, text(str(valeur), "Helvetica", 12, white))
    return overlay(label, back)

donnees = [
    ("Allemagne", 81174000),
    ("France", 66352469),
    ("Italie", 60795612),
    ("Autriche", 8584926),
    ("Suisse", 8236573)
]

def beside_low(g1, g2):
    g1 = pin(bottom_right, g1)
    g2 = pin(bottom_left, g2)
    return compose(g1, g2)
```

3. Boules de couleurs

Complétez le code ci-dessous afin d'obtenir le dessin suivant :



```
from pytamaro import *

couleurs = (red, yellow, green, cyan, blue, magenta)

diametres = range(30, 110, 10)

dessin = empty_graphic()

for couleur in couleurs:
    ligne = empty_graphic()
    for valeur in diametres:
        pass
```

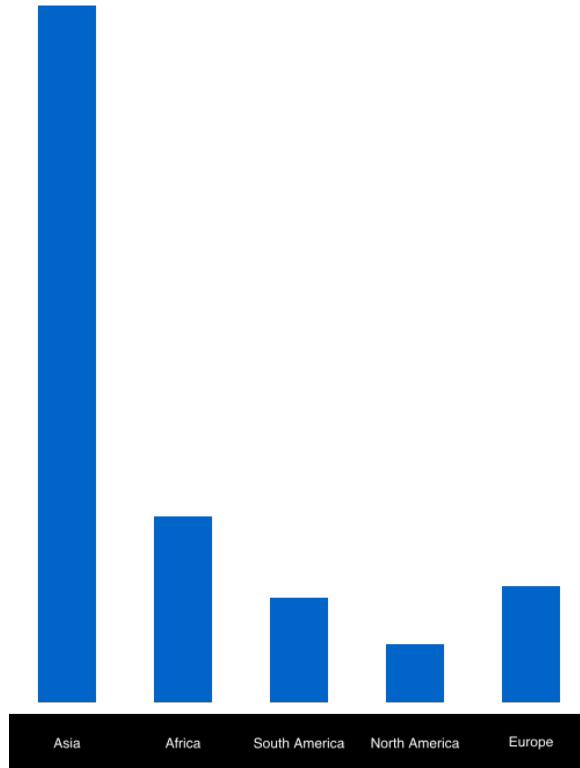
4. Dégradé rouge-jaune

Réalisez le dessin suivant, formé de 256 couches de 200 pixels de largeur pour 2 pixels de hauteur. La première couche a pour code couleur 255, 0, 0 en RGB, et la dernière, tout en bas, a pour code couleur 255, 255, 0.



5. Population mondiale

Réalisez le dessin ci-dessous à partir du code Python plus bas. Vous trouverez le fichier de données **data-ex5.csv** sur Moodle.



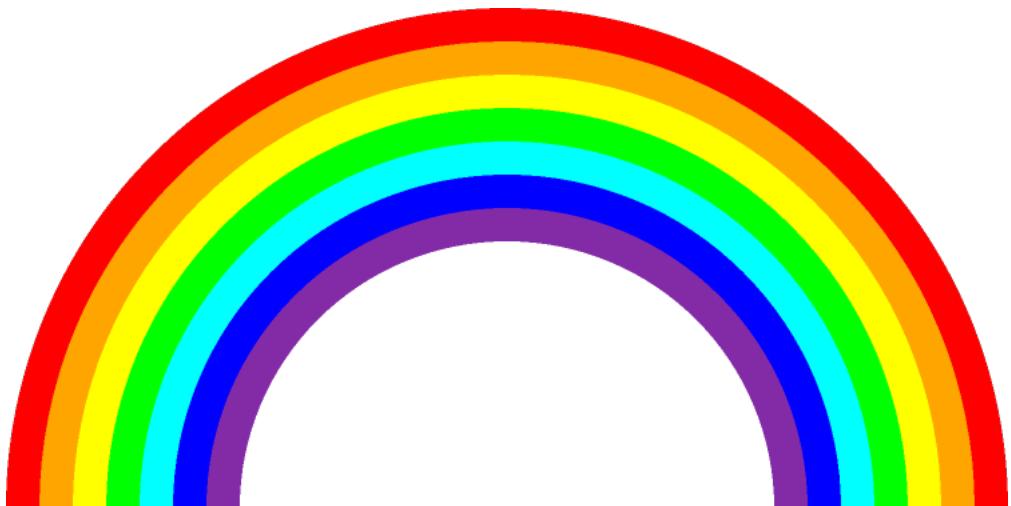
```
from pytamaro import *
from csv import reader
import os

file = "ex5-data.csv"
file = os.path.join(os.path.dirname(__file__), file)
with open(file) as f:
    data = list(reader(f))
print(data) # Pour votre information !

def barre(titre, valeur):
    bar = rectangle(50, valeur * 10, rgb_color(0, 100, 200))
    back = rectangle(70, valeur * 10 + 20, transparent)
    bar = overlay(bar, back)
    label = text(titre, "Helvetica", 12, white)
    back = rectangle(100, 50, black)
    label = overlay(label, back)
    return above(bar, label)
```

6. Arc-en-ciel

Réalisez le dessin ci-dessous d'un arc-en-ciel.



Indice : Aidez-vous de la fonction `circular_sector` de PyTamaro.

7. Rebonds infinis

Pour cet exercice, vous allez travailler non plus sur une unique image, mais sur une séquence d'images qui forment une animation ! Complétez le code ci-dessous afin de terminer l'animation d'une balle qui rebondit sur le sol de manière totalement élastique (sans perte d'énergie).

```
from pytamaro import *

def frame(progression, couleur_boule, couleur_sol):
    boule = ellipse(50, 50, couleur_boule)
    hauteur_dessus = 500 * progression
    hauteur_dessous = 500 - hauteur_dessus
    forme = above(rectangle(50, hauteur_dessus, transparent), boule)
    forme = above(forme, rectangle(50, hauteur_dessous, transparent))
    forme = above(forme, rectangle(200, 50, couleur_sol))
    return forme

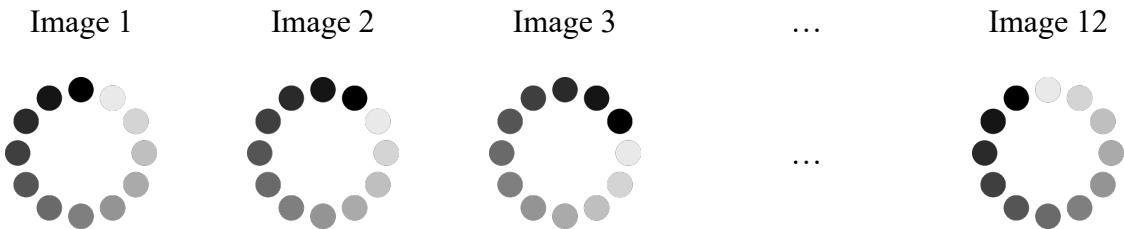
frames = []
for i in range(0, 101):
    p = (i / 100) ** 2
    frames.append(frame(p, red, blue))

save_animation("rebonds.gif", frames, 20)
```

Bonus : Faites en sorte qu'à chaque impact les couleurs de la boule et du sol s'intervertissent.

8. Chargement

Réalisez une animation de chargement basée sur les images suivantes, qu'il vous faudra reproduire :



Indice : Utilisez la fonction `rotate` afin de faire tourner un graphique dans PyTamaro.

9. Demande de rançon

Complétez le code plus bas afin de terminer le générateur de texte façon « demande de rançon ».



Le programme doit inviter l'utilisateur à saisir un texte, puis lui retourner une image de ce même texte formé de lettres comme découpées dans des magazines.

```
from pytamaro import *
from random import choice

couleurs = []
couleurs_fond = []

for i in range(0, 360, 30):
    couleurs.append(hsl_color(i, 1, 0.3))
    couleurs_fond.append(hsl_color(i, 0.5, 0.9))

tailles = (32, 36, 42, 48, 56)

polices = ("Helvetica", "Times", "Courier")

tilts = (0, -2, 3, 5, -5)

entree = input("Entrez un texte : ")
```

Indice : La fonction `choice` du module `random` permet d'obtenir un élément pseudo-aléatoire d'une séquence.

10. Étudiants à l'EPFL

Pour ce dernier exercice, vous êtes amenés à produire un graphique ou une animation afin de visualiser l'évolution de nombre d'étudiants dans les différentes sections de l'EPFL.

Cet exercice est volontairement très libre. Laissez libre court à votre créativité !
Si toutefois vous le souhaitez, vous pouvez démarrer avec le programme suivant :

```
from pytamaro import *
from csv import reader
import os

file = "ex10-effectifs-EPFL.csv"
file = os.path.join(os.path.dirname(__file__), file)
with open(file) as f:
    data = list(reader(f, delimiter=";"))

sections = data[0][1:]
annees = []
effectifs = []
for row in data[1:]:
    annees.append(int(row[0]))
    ligne = []
    for value in row[1:]:
        ligne.append(int(value))
    effectifs.append(ligne)

print(sections)
print(annees)
print(effectifs)
```