

TP_05 NSI -- lycée de l'Elorn 2023 --

Les bases en Python, seconde partie

Si vous avez terminé le Notebook précédent ! Voici une série d'activités que vous me restituerez en téléchargeant le Notebook.

<https://capytale2.ac-paris.fr/web/c/30dc-761773/ttt> (ou par le code 30dc-761773)

I - Le type list (liste)

Nous étudierons plus en détail ce type de variable très intéressant. Une liste peut contenir une série de valeurs. Une liste est déclarée par une série de valeurs séparées par des virgules.

```
In [1]: jours = ["lundi", "mardi", "mercredi", "jeudi", "vendredi"]
print(jours)
#On peut appeler les éléments d'une liste par la position (leur indice)
print(jours[0]) #affiche le 1er élément
print(jours[2]) #affiche le 3ème élément
print(len(jours)) #affiche la longueur de la liste (nb d'éléments)
```

```
['lundi', 'mardi', 'mercredi', 'jeudi', 'vendredi']
lundi
mercredi
5
```

```
In [2]: #L'indexage peut être négatif
jours = ["lundi", "mardi", "mercredi", "jeudi", "vendredi"]
print(jours[-1]) #affiche le dernier élément
print(jours[-2]) #affiche l'avant dernier élément
```

```
vendredi
jeudi
```

```
In [3]: #Ajout d'un élément de la liste
jours = ["lundi", "mardi", "mercredi", "jeudi", "vendredi"]
jours.append("saturday")
```

```
jours=jours+["sunday"]  
print(jours)
```

```
['lundi', 'mardi', 'mercredi', 'jeudi', 'vendredi', 'saturday', 'sunday']
```

In [4]:

```
#On peut donc afficher l'ensemble de la liste en utilisant une boucle FOR.  
#On dit alors que la liste est itérable  
jours = ["lundi", "mardi", "mercredi", "jeudi", "vendredi"]  
  
for element in jours:  
    print(element)  
  
#ce qui est équivalent à ceci  
for i in range(len(jours)):  
    print(jours[i])
```

```
lundi  
mardi  
mercredi  
jeudi  
vendredi  
lundi  
mardi  
mercredi  
jeudi  
vendredi
```

In [5]:

```
#On peut tester si un élément est contenu dans une liste  
jours = ["lundi", "mardi", "mercredi", "jeudi", "vendredi"]  
print("lundi" in jours)  
print("samedimanche" in jours)  
if "mardi" in jours : print("mardi est dans la liste")
```

```
True  
False  
mardi est dans la liste
```

II - Le module Random

Le langage Python dispose d'un module random qui contient plusieurs fonctions qui nous seront utiles pour générer du hasard, par exemple simuler le jet d'un dé, mélanger des cartes ...

In [6]:

```
# random est un module permettant de générer des nombres aléatoires
import random #importation du module
x=random.random() #valeur de type float entre 0 et 1, 1 exclus
y=random.randint(0,3) #valeur entière entre 0 et 3 inclus
z=random.choices(["Pile","Face"]) #renvoie un élément de cette liste au hasard.
print("x=",x)
print("y=",y)
print("z=",z)
print("z=",z[0])

#Exécuter plusieurs fois ce script et vérifier le comportement aléatoire des affectations
```

```
x= 0.5646104743741605
y= 3
z= ['Pile']
z= Pile
```

Programme 1 : Loto

Réaliser un programme capable de proposer une grille de Loto®. Pour jouer vous devez définir 6 numéros : 5 numéros sur une grille de 49 numéros et 1 numéro chance sur une grille de 10 numéros. Vous prendrez soin d'optimiser le code par l'usage de boucles et de personnaliser votre compte-rendu.

Si vous êtes en forme : Utiliser une liste afin d'éviter de proposer plus d'une fois le même numéro.

In []:

Programme 2 : Pile ou face

Écrire un programme qui simule 10 000 lancers d'une pièce de monnaie. Vous afficherez le nombre ainsi que le pourcentage de piles et de faces obtenus. Écrire une version avec une boucle for et une autre avec une boucle while.

In []:

Programme 3 : Le juste prix

Le juste prix consiste à deviner le prix d'un objet (un entier rond, inférieur ou égal à 100 euros) en faisant des propositions. L'inconnue (le prix) est déterminé aléatoirement, chaque proposition du joueur devra amener une des réponses suivantes :

- C'est moins
- C'est plus
- C'est le bon prix

--> Modifier le programme pour proposer au joueur une nouvelle partie :

```
"C'est le bon prix. Souhaitez-vous rejouer (O/N) ?"
```

--> Modifier ce programme afin de rappeler le nombre de propositions ayant été faites par le joueur :

```
"C'est le bon prix, vous avez réalisé X tentatives pendant le jeu. Souhaitez-vous rejouer (O/N) ?"
```

In []:

III - Manipulation d'image

Charger le fichier "Hermione_Brest.jpg" sur votre serveur NoteBook. Cette image est au format 600x400 pixels.

Si l'image n'est pas disponible :

--> Menu Capytale --> Fichiers annexes --> Disponibles le temps de la session --> + Ajouter



Le programme ci-dessous est capable de faire la copie d'une image.

In [7]:

```
from PIL import Image
image_1=Image.open('Hermione_Brest.jpg') #image source
image_2=Image.new('RGB',(600,400)) #on crée une image du même format mais vide

#On parcourt chacun des pixels constituant l'image 1
#Remarque : Il y a des méthodes plus simples et plus rapides pour copier des images...
for x in range(600):
    for y in range(400):
        R=image_1.getpixel((x,y))[0] #donne le codage du rouge
        V=image_1.getpixel((x,y))[1] #donne le codage du vert
        B=image_1.getpixel((x,y))[2] #donne le codage du bleu
        image_2.putpixel((x,y),(R, V, B))    #affecte la couleur du pixel à la nouvelle image

image_2    #On affiche l'image 2
```

Out[7]:



In []:

Le programme ci-dessous est capable de copier une image en l'inversant horizontalement. l'image 2 est créée à partir de l'image 1 en inversant la position de chaque pixel : pixel(0,0)->pixel(599,399), pixel(0,399)->pixel(599,0) ...

In [8]:

```
from PIL import Image
image_1=Image.open('Hermione_Brest.jpg')
image_2=Image.new('RGB',(600,400)) #on crée une image du même format mais vide

for x in range(600):                #On parcourt chacun des pixels constituant l'image 1
    for y in range(400):
        R=image_1.getpixel((599-x,399-y))[0] #donne le codage du rouge
        V=image_1.getpixel((599-x,399-y))[1] #donne le codage du vert
        B=image_1.getpixel((599-x,399-y))[2] #donne le codage du bleu
        image_2.putpixel((x,y),(R, V, B))    #affecte la couleur du pixel à la nouvelle image

image_2    #On affiche l'image 2
```

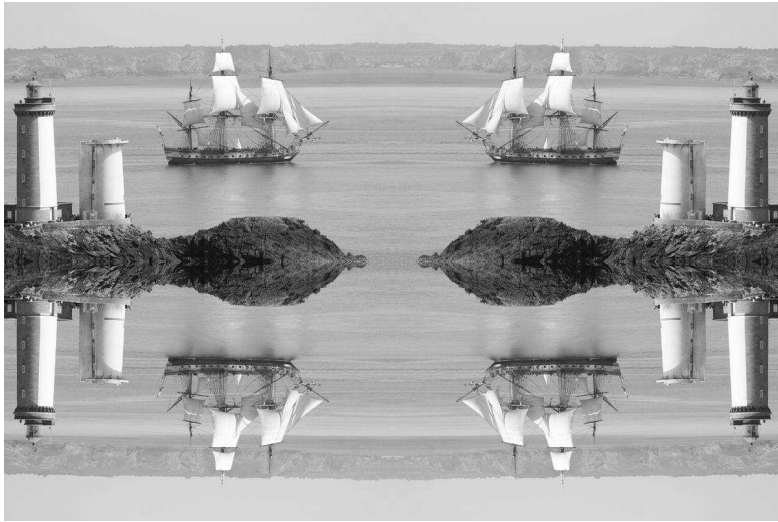
Out[8]:



In []:

Travail attendu :

- Proposer un programme capable de recopier 2 fois une image sur une autre image (de taille 2 fois plus grande évidemment).
- Proposer un programme capable de générer le résultat suivant :



In []: