

Cours 4 : Micro-projet sur la boucle for

🔊 Compétences évaluées :

- C1.1 [Utiliser des outils numériques pour traiter et produire de l'information]
- C2.2 [Communiquer par écrit, de façon claire et non ambiguë, en langage Python]
- C8.1 [Dialoguer au sein d'une équipe]
- C9.3 [Apprendre en remettant en question son code]
- C10.5 [Veiller au respect des échéances]
- C11.2 [Réaliser un projet en langage Python]

Objectif de la séance

Le but de cette séance est de convertir une image couleur en noir et blanc. Pour cela, nous allons apprendre à utiliser quelques commandes pour pouvoir les insérer dans le code final.

1 Fonction int

Question 1 : Saisir dans la console les commandes `int(3.0)`, `int(3.3)`, `int(3.7)` et `int(4.1)`. Que fait la fonction `int` ?

2 Tableaux et images

On rappelle qu'une image matricielle est un tableau (en anglais : *array*) de pixels notamment défini par le nombre de pixels horizontaux et le nombre de pixels verticaux.

Question 2 : À partir de l'espace des classes (MBN > Espace des classes > DNMADE 2 > Dossiers partagés > Algorithmique > Projet boucle for), télécharger le code `première_image.py` ainsi que l'image `chat.jpg`. Les mettre dans un dossier commun, par exemple intitulé « Lecture image ».

Le code est le suivant :

```
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image #la librairie PILlow permet de traiter des images

image_de_chat = Image.open('chat.jpg') #importe l'image chat.jpg dans Python et l'
    enregistrer dans la variable image_de_chat
TABLEAU=np.array(image_de_chat) #transforme l'image en un tableau de pixels nommé
TABLEAU
```

Question 3 : Afin de travailler sur ce bout de code, faire un clic droit sur le script et sélectionner « Run in Interactive Window > Exécuter le fichier actif dans la fenêtre interactive ».

Cette fenêtre interactive permet, comme son nom l'indique, d'interagir avec le code que l'on vient de compiler. Afin d'exécuter des commandes, il suffit de les taper dans la fenêtre interactive puis d'appuyer sur Maj+Enter.

Question 4 : Taper `np.size(TABLEAU,0)` et `np.size(TABLEAU,1)`. À quoi correspondent ces deux valeurs ?

Indices d'un tableau

Soit un tableau bidimensionnel T . La commande $T[i, j]$ renvoie l'élément situé dans sa $(i + 1)$ ^{ème} ligne et $(j + 1)$ ^{ème} colonne.

Exemple : $T[25, 72]$ renvoie l'élément situé dans la 26^{ème} ligne et 73^{ème} colonne du tableau T .

Question 5 : Taper `TABLEAU[18,105]`. À quoi correspondent les trois nombres affichés, selon vous ?

Question 6 : Taper `TABLEAU[18,105][0]`, `TABLEAU[18,105][1]` et `TABLEAU[18,105][2]`. Commenter.

3 Boucles imbriquées

Boucles imbriquées

Soit un tableau T possédant NL lignes et NC colonnes. Si l'on souhaite effectuer une action sur chacun des éléments du tableau T , on utilisera des boucles l'une dans l'autre (voir exemple de code ci-dessous).

```
for L in NL:
    for C in NC:
        #instructions
```

4 Passer un pixel en niveau de gris

Soit un pixel désigné par le triplet décimal (r, v, b) (r , v et b sont donc trois nombres compris entre 0 et 255, représentant la part de rouge, de vert et de bleu dans le pixel).

On peut passer en niveau de gris (c'est-à-dire : « en noir et blanc ») le pixel en donnant comme nouvelle valeur à r , v et b la valeur :

$$G = \left\lfloor \frac{r + v + b}{3} \right\rfloor$$

où $\lfloor x \rfloor$ représente la partie entière de x (`int(x)` en Python).

5 Projet

Vous avez à présent tous les outils en main pour convertir une image en niveaux de gris.

À partir de l'espace des classes (MBN > Espace des classes > DNMADE 2 > Dossiers partagés > Algorithmique > Projet boucle for), télécharger le code `code_élève.py` ainsi que l'image `chien.jpg`. Les mettre dans un dossier commun, par exemple intitulé « Projet boucle for ».

Ce fichier Python a été prérempli par mes soins, mais certaines lignes sont manquantes. Il faut donc réussir à le compléter pour obtenir l'image du chien en niveau de gris.

Pour information, le code à compléter est présent ci-dessous :

```
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image #la librairie PILlow permet de traiter des images

image_de_chien = Image.open('chien.jpg') #importe l'image chien.jpg dans Python et
l'enregistrer dans la variable myimg
tableau_chien=np.array(image_de_chien) #transforme l'image en un tableau de pixels
nommé img

N_vertical=... #nombre de pixels verticaux de l'image
N_horizontal=... #nombre de pixels horizontaux de l'image
for i in ...: #on fait une boucle imbriquée pour passer sur tous les pixels de l'
image
    for j ...:
        pixel=... #on désigne le pixel qui nous intéresse
        rouge=... #on donne son niveau de rouge
        vert=... #on donne son niveau de vert
        bleu=... #on donne son niveau de bleu
        gris=... #on calcule le niveau de gris
        nouveau_rouge=... #on donne la nouvelle valeur du rouge
        nouveau_vert=... #on donne la nouvelle valeur du vert
        nouveau_bleu=... #on donne la nouvelle valeur du bleu
        tableau_chien[i,j]=... #on remplace le pixel précédent par le nouveau
        triplet de couleurs

plt.imshow(tableau_chien)
plt.show()
```