

L'objectif est de comprendre quelles sont les opérations de traitement pour obtenir une image numérique

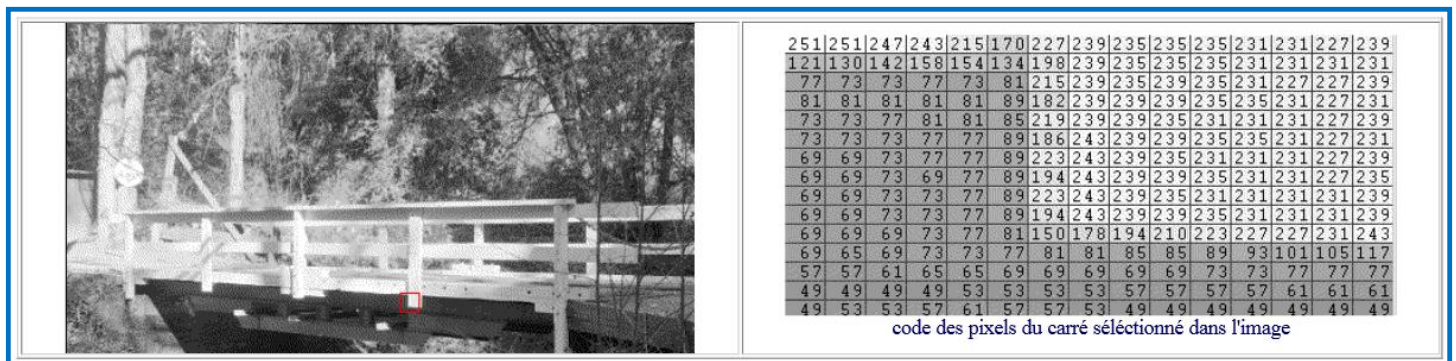
### 1. Codage d'une image en niveaux de gris

Une image en **niveau de gris** est une image dont les couleurs varient du blanc au noir.

Chaque pixel est codé par le niveau de l'intensité lumineuse, souvent sur 8 bits ce qui donne  $2^8 = 256$  **nuances de gris** possibles, codées sur un nombre entier compris entre **0** et **255**.

La valeur **0** correspond à une intensité lumineuse nulle, le **noir**, tandis que la valeur **255** représente l'intensité lumineuse maximale, le **blanc**.

|     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 000 | 008 | 016 | 024 | 032 | 040 | 048 | 056 | 064 | 072 | 080 | 088 | 096 | 104 | 112 | 120 | 128 |
|     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
|     | 255 | 248 | 240 | 232 | 224 | 216 | 208 | 200 | 192 | 184 | 176 | 168 | 160 | 152 | 144 | 136 |



A l'aide de la documentation technique « Fonctionnalités du module Image de la bibliothèque PIL (Pillow) »

1.1. Ouvrez le programme « **carre** » sous Jupyter notebook, quel est le **résultat** ?

```
from PIL import Image
carre = Image.new("L", (20,20),100)
display(carre)
```

1.2. Faites un programme sous Jupyter notebook qui permet d'afficher un rectangle de définition **20 x 10** :

- de couleur **blanche** ;
- puis de couleur **noire** ;
- puis de couleur **grise** (code de votre choix) avec les **4 pixels de coins blancs**.

Couleur blanche :

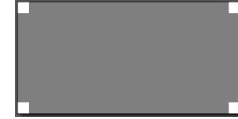
```
from PIL import Image
```

Couleur noire :

```
from PIL import Image
```

### Couleur grise avec coins blancs :

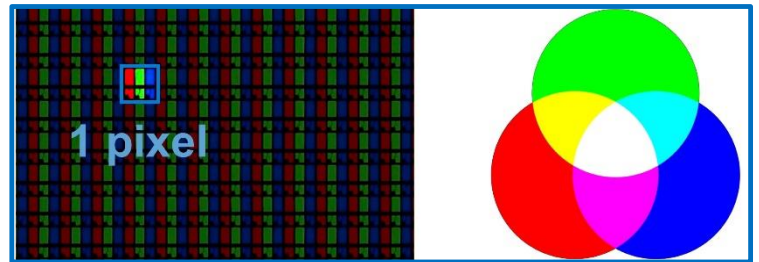
```
from PIL import Image
```



## 2. Codage d'une image en couleur

La couleur d'un pixel est représentée par trois valeurs : celle du **rouge (R)**, celle du **vert (V)** et celle du **bleu (B)** qui la composent.

Il faut imaginer que l'on règle l'**intensité** de trois sources lumineuses rouge, verte et bleue qui se mélangent pour donner la **couleur résultante**.

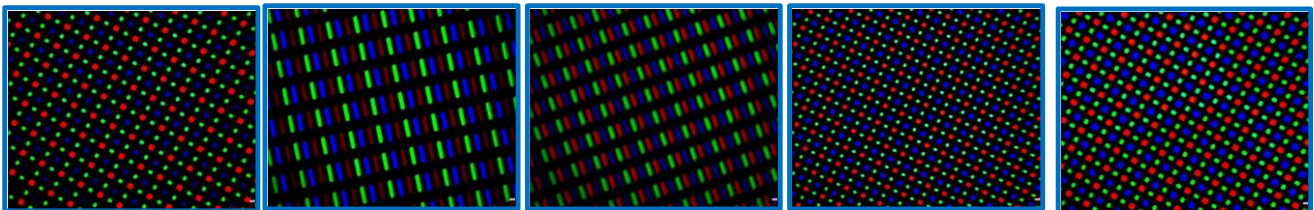


L'intensité de chacune des trois couches est codée sur **8 bits** ce qui nous donne un codage de **0 à 255**.

Comme pour le noir et blanc, pour chacune des couches, plus le code est proche de zéro moins vous envoyez de lumière et plus le code est proche de 255 plus vous envoyez de lumière.

Ce format de couleur n'est pas évident car l'aspect de la couleur résultante est parfois difficile à anticiper. On trouve néanmoins sur internet pléthore de convertisseurs RGB : <https://www.toutes-les-couleurs.com/code-couleur-rvb.php>

Lorsque vous observez un écran à la loupe, la couleur est un agencement de pixels :



Samsung GalaxyA5

Samsung GalaxyJ1

Echo Horizon

Galaxy Note 9

iPhone X

2.1. Les 3 valeurs **RGB** (**0 à 255**, **0 à 255**, **0 à 255**) permettent donc de coder une énorme variété de teintes. Combien ?

|  |   |   |   |   |   |
|--|---|---|---|---|---|
|  |  |  |  |  |  |
| 112/112/112  | 89/89/89  | 71/71/71  | 54/54/54  | 19/139/99   | 1/144/105   |
|  |  |  |  |  |  |
| 254/125/25   | 0/0/0   | 29/88/152   | 1/137/180   | 0/160/198   | 240/2/127   |
|  |  |  |  |  |  |
| 255/255/0  | 100/16/131  | 145/22/135  | 253/24/19   | 17/20/92  | 13/43/136   |
|  |  |  |  |  |  |
| 210/11/121   | 227/2/127   | 211/14/43   | 242/1/108   | 17/17/69  | 32/17/131   |

2.2. Importez l'image « moustique.jpg » dans vos documents,

- réalisez le programme sous Jupyter permettant :
  - d'ouvrir cette image ;
  - de connaître son **format** ;
  - de connaître son **mode** ;
  - de connaître sa **taille** ;
  - de connaître le **code couleur** de 3 pixels de votre choix ;
  - d'afficher l'image.

```
from IPython.display import Image
from PIL import Image
```



2.3. Puisque vous avez tout compris, une petite mise en application en utilisant des boucles for...

- Codez l'image du drapeau **Français** et du drapeau **Suisse**.

```
from IPython.display import Image
from PIL import Image
```

```
drapeau_français = Image.new(
```



```
from IPython.display import Image
from PIL import Image
```

```
drapeau_suisse = Image.new (
```

2.4. **Recouvrez** la plaque d'immatriculation de la voiture de votre professeur par un **rectangle jaune**.

Il faut mettre le fichier de **l'image tesla.jpg** dans le même dossier que votre programme dans vos documents.

```
from IPython.display import Image
from PIL import Image
```



## 2.5. Encadrez « numérique photographie » de l'image par un rectangle violet.

Il faut mettre le fichier de l'image **snt.jpg** dans le même dossier que votre programme dans vos documents.

```
from IPython.display import Image
from PIL import Image
```

