

## Série 1 (introduction à OpenCV)

Après avoir installé Python, OpenCV et numpy ([lien](#)). Réaliser les exercices avec les fonctions

- `cv2.imread(filename[,flags])` # [doc](#)
- `cv2.imshow('titre',img)` # [doc](#)
- `cv2.imwrite(filename, img[,params])` # [doc](#)
- `cv2.cvtColor(input, code[,output[,dstCn]])` # [doc](#)
- `cv2.split(src)` # [doc](#)

les propriétés d'une image et le squelette de code suivant :

```
import cv2
import numpy as np
def loadAndDisplayColorImage(imagePath):
    ...
def importColorImageAsGray(imagePath):
    ...
def importColorImageAndConvert(imagePath):
    ...
def splitImageOpenCV(image):
    ...
def displaySplitImageNumpy(image):
    ...
    print("Press any key\n")
    cv2.waitKey(0)
    cv2.destroyAllWindows()
```

### Exercice 1 (E/S et conversions)

- Ecrire la fonction `loadAndDisplayColorImage` afin qu'elle ouvre un fichier `lena.png` et l'affiche
- Compléter deux fonctions qui ouvrent la même image et la renvoie en niveau de gris:
  - 1) en ajoutant un paramètre à `cv2.imread()` pour `importColorImageAsGray`
  - 2) en utilisant la méthode `cv2.cvtColor()` pour `importColorImageAndConvert`
- Ecrire la fonction `saveGrayImage()` qui enregistre l'image de Lena en niveaux de gris dans un fichier `lenaGray.jpg`

### Exercice 2 (séparation des couleurs)

- Ecrire une fonction `splitImageOpenCV()` qui reçoit une image couleur en paramètre et affiche ses trois composantes R, G, et B avec la méthode `split` d'*OpenCV*
- Ecrire une fonction `splitImageNumpy()` () qui reçoit une image couleur en paramètre et qui affiche ses trois composantes R, G, et B avec *numpy*: `b = img[:, :, 0]`
- Discussion

[DOC]

OpenCV 4.1.0  
Open Source Computer Vision

Main Page Related Pages Modules Namespaces Classes Files Examples Java documentation

Search

### OpenCV modules

- Introduction
- OpenCV Tutorials
- OpenCV-Python Tutorials
- OpenCV.js Tutorials
- Tutorials for contrib modules
- Frequently Asked Questions
- Bibliography
- Main modules:
  - core, **Core functionality**
  - imgproc, **Image Processing**
  - imgcodecs, **Image file reading and writing**