# TP0 - Données Multimedia

Campedel - Juillet 2014
CES Data Scientist


**Objectifs**
savoir lire et écrire une image avec Python (et opencv)
savoir visualiser une image à l'aide de Python

**Ressources bibliographiques**
Environnement : Anaconda + opencv (wrapper python)
https://docs.python.org/2/tutorial/
http://docs.continuum.io/anaconda/index.html
http://opencv-python-tutroals.readthedocs.org/
http://www.packtpub.com/opencv-computer-vision-with-python/book?tag=

**Qu'est-ce qu'une image (numérique) ?**

Un tableau de pixels.
Un pixel = une position dans l'image (numéro de ligne, numéro de colonne) + valeur d'intensité

Il existe deux formats principaux de pixels pour les photographies (et les images d'une vidéo)

- Niveau de gris : 8 bits (=1 byte) par pixel pour coder l'intensité
- En couleur (BGR, Blue Green Red) : 8 bits par canal de couleur, soit au total 24 bits (3 bytes).

**Comment cela se traduit-il avec OpenCV ?**

OpenCV est une librairie C++ avec un wrapper Python (module cv2) . Elle est très complète et est devenue la référence pour le traitement des images.
La documentation en ligne est disponible ici : http://docs.opencv.org

- http://docs.opencv.org/opencv2refman
- https://opencv-python-tutroals.readthedocs.org/en/latest/py_tutorials/py_tutorials.html

Il existe également un livre (payant), contenant de nombreux exemples exploitant Python et OpenCV : http://www.packtpub.com/opencv-computer-vision-with-python/book?tag=

*Type des images*
Une image OpenCV est une numpy.array (array 2D ou 3D).
On peut accéder à la valeur d'intensité par une expression du type image[x,y] pour une image en niveau de gris. Dans le cas d'une image couleur, il faut ajouter l'indice du canal de couleur.

**Exercice 1 : manipulation simples**

Créer une i***mage en niveaux de gris***, à l'aide d'un numpy.array, dont les valeurs seraient aléatoires. Puis créer une ***image couleur*** avec le même nombre de valeurs aléatoires. Sauvegarder cette image au format jpg,

*indices (code pour la création d'image) :*

```
import numpy
import cv2

randomByteArray = bytearray(os.urandom(120000))
flatNumpyArray = numpy.array(randomByteArray)
# Convert the array to make a 400x300 grayscale image.
grayImage = flatNumpyArray.reshape(300, 400)
bgrImage = flatNumpyArray.reshape(100, 400, 3)
```

Pour les fonctions d'écriture/lecture d'image : pensez à exploiter cv2 ! Consultez l'aide en ligne (cf ressources).

Pour vérifier que vos images ont bien été créées, vous pouvez utiliser xview sous Linux, ou une visionneuse d'images quelconque (dans le menu applications).

**Exercice 2 : Visualisation avec matplotlib**

Compléter le code précédent pour visualiser vos deux images dans l'environnement Python.

*Indices :*
Pensez au module matplotlib.pyplot (fonction imshow) et attention que l'image créée en niveaux de gris ne doit pas apparaître en couleur !

```
import matplotlib.pyplot as plt

plt.figure
plt.subplot(2,1,1)
plt.imshow(bgrImage)
plt.title('image en couleur')
plt.subplot(2,1,2)
plt.imshow(grayImage) #attention il y a quelque chose à ajouter !
plt.title('image en niveau de gris')
```