Découverte et utilisation des Images

Objectif de la séance :
Découvrir ce qu'est une image et comment à s'en servir sous Python.

Partie 1 : Une image numérique qu'est ce que c'est ?

C'est un ensemble de petit carré qui contiennent des couleurs : les pixels.
Plus l'image a de pixels, plus sa résolution est grande et plus l'image paraît nette (et jolie).
La résolution d'une image, ou d'un écran, est donnée sous la forme L x l, où L et l sont le nombre de pixels en Longueur et en largeur :

Image 600 x 387

Image 1280 x 826

Image 3840 x 2160

On peut donc connaître le nombre de pixels d'une image en faisant les produits de ces deux nombres. Plus il y a de pixels, plus on peut afficher l'image en grand en la gardant nette.

Détaillons maintenant une image au plus près : Image zoom

Les couleurs peuvent être définis de différentes façon : systèmes RGB (ou RVB en français), Hexadécimal, LSV, LAB, MYCK, HSL ...

Une image est donc un grand tableau dans lequel chaque « case », qui est un pixel, contient un code de couleur.

Nous utiliserons le système RGB, Red Green Blue (RVB Rouge Vert Bleu).

Partie 2 : Librairie Python

Pour utiliser une image en langage Python, des fonctions préexistantes ont été définies dans ce qui s'appelle (dans tous les langages de programmation) une librairie.

La librairie image de Python s'appelle PIL (Python Imaging Library).
En début de programme, on doit donc l'importer, comme avec le robot sur france-ioi, en tapant :

```
from PIL import Image
```

Pour aller chercher un pixel, il suffit d'utiliser ses « coordonnées » dans l'image : c'est un repérage 2D, en 2 Dimensions comme dans le plan, avec deux axes : abscisses et ordonnées.
Ces deux axes ont pour origine le pixel en haut à gauche et partent, vers la droite pour l'abscisse, vers le bas pour l'ordonnée : les coordonnées sont donc toujours positives.

Exemple : Voici les coordonnées des pixels sur une image de résolution 5 x 4 :

(0;0)	(1;0)	(2;0)	(3;0)	(4;0)
(0;1)	(1;1)	(2;1)	(3;1)	(4;1)
(0;2)	(1;2)	(2;2)	(3;2)	(4;2)
(0;3)	(1;3)	(2;3)	(3;3)	(4;3)

Voici un exemple d'utilisation avec la librairie Python :

```
from PIL import Image
```

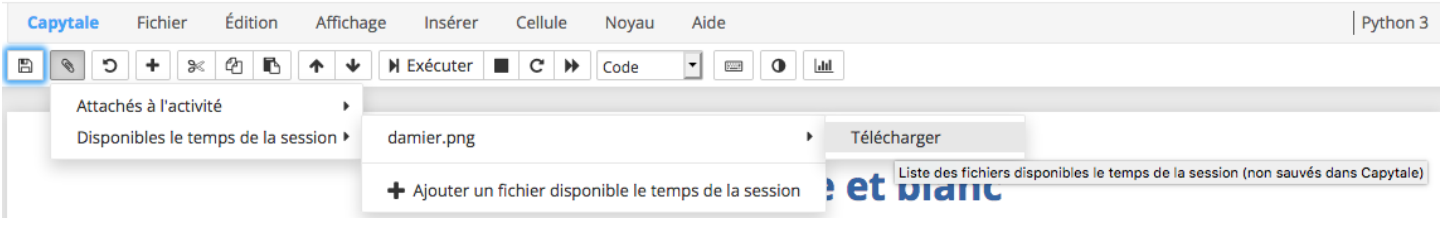
```
Nom_de_la_variable_qui_contient_l'image = Image.open(''nom_du_fichier'')
```
→ permet d'ouvrir l'image « nom_du_fichier » et de la stocker dans une variable « Nom_de_la_variable_qui_contient_l'image »

```
Nom_de_la_variable_qui_contient_l'image.show()
```
→ permet d'afficher cette image

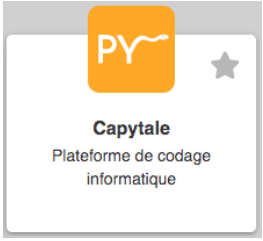
```
rouge, vert, bleu = Nom_de_la_variable_qui_contient_l'image.getpixel((0,0))
```
→ permet d'obtenir les trois nombres composant la couleur du pixel de coordonnées (0;0) en codage RGB (RVB)

```
Nom_de_la_variable_qui_contient_l'image.save(''Nom_de_la_variable_qui_contient_l'image.png'')
```
→ permet de sauvegarder, télécharger cette image puis de l'ouvrir, et de zoomer cette image.

Après une sauvegarde (imag.save()), on peut accéder au(x) fichier(s) sauvegardé(s) comme suit:



Partie 3 : Utilisation de Python sur Capytale



A partir de l'ent : mon-e-college.loiret.fr, on peut accéder à Capytale :

Exercice 1 : Decouverte d'une image

Dans cette activité, vous allez découvrir une première utilisation de la librairie PIL pour le traitement d'image

Taper le code suivant : cc90-1647855 puis GO !

ou bien

Voici le lien de l'activité : https://capytale2.ac-paris.fr/web/c/cc90-1647855

Exercice 2 : Créer une image et la modifier

Dans cette activité, vous allez pouvoir modifier les pixels d'une image puis modifier la couleur d'une ligne puis vous pourrez créer votre image, des exemples vous sont présenter à la fin du notebook.

Sur Capytale, taper ce code : 3328-1648054