**Partie E : Traitement de l’image par programmation ( Langage Python)**



#### Peut-on toujours faire confiance aux photos que l’on voit sur les médias ou les réseaux sociaux ?

Vous allez travailler sur les pixels d'une image **“pomme.jpeg”**en utilisant le langage de programmation **Python.**

**Préparation**

- Le **fichier “pomme.jpeg”** est à télécharger sur le site capneuronal. Ce fichier est à enregistrer dans votre espace en créant un dossier **“SNT-prog-python”**

Quelle est la définition de cette image ? ............................

Quelle est la résolution de cette image ? ............................



**EduPython** est un **environnement python**: un éditeur et un interpréteur dans lequel il est possible de **coder** (écrire un programme) et d’**interpréter** le code (exécuter le programme).

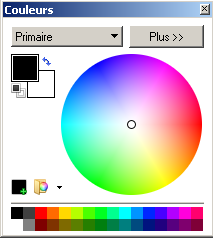
Vous trouverez l’icone sur le bureau pour lancer EduPython. **Tous les fichiers/programmes que vous allez réaliser sont à enregistrer dans le même répertoire que la photo.**

**Pré-requis:**

|  |  |
| --- | --- |
| Avant de commencer à écrire un programme qui nous permettra de travailler sur les pixels d'une image, il est nécessaire de préciser que chaque pixel a des coordonnées (x,y).  Comme vous pouvez le constater sur le schéma ci-dessus, le pixel de coordonnées (0,0) se trouve en haut à gauche de l'image. Si l'image fait 800 pixels de large et 600 pixels de haut, le pixel ayant pour coordonnées (400,300) sera au milieu de l'image. |  |

**Activité 1:** Saisir le code suivant dans l’éditeur, le commenter dans le tableau. Enregistrer le fichier sous le nom “rvbpixel-1.py” dans le même répertoire que le fichier pomme.jpeg et lancer son exécution

|  |  |
| --- | --- |
| from PIL import Image | Une **bibliothèque**, ici PIL, est un ensemble de fonctions regroupées et mises à disposition afin de pouvoir être utilisées sans avoir à les réécrire.  Celles-ci permettent de faire : du calcul numérique, du graphisme, de la programmation internet ou réseau, du formatage de texte, de la génération de documents.. |
| img = Image.open("pomme.jpg")  img.show() | Image.open : ………………………………………………………..  img  : ………………………………………………………..  … .show() : ……………………………………………………….. |
| largeur\_image=img.width  hauteur\_image=img.height | ... .width : ………………………………………………………..  ... .height: ……………………………………………………….. |
| r,v,b=img.getpixel((100,250)) | getpixel : ……………………………………………………….. |
| print("rouge : ",r,"vert :",v,"bleu : ",b)  print("Largeur x : ", largeur\_image, "hauteur y:", hauteur\_image) | print : ……………………………………………………….. |



- Qu'est-ce qui s'affiche dans la fenêtre console ? ……………………………………

- Ouvrir le fichier « pomme.jpeg » avec le logiciel paint.net :

Cliquez sur plus de la fenêtre ci-contre et saisir les valeurs du rouge, vert et bleu. A quoi correspond la couleur proposée ? …………………

......................................................................................................................................

- Modifiez le programme pour qu'il affiche les valeurs du rouge, du vert et du bleu du pixel de coordonnées (250,300). Vérifiez en vous déplaçant sur la photo sous paint.net

**Activité 2 :** Saisir le code suivant, le commenter dans le tableau, l’enregistrer sous le nom

«pomme-1pixel.py »  et lancer son exécution

|  |  |
| --- | --- |
| from PIL import Image  img = Image.open("pomme.jpg")  largeur\_image=img.width  hauteur\_image=img.height |  |
| img.putpixel((250,250),(255,0,0)) | … .putpixel : ……………………………………………………….. |
| img.save("pomme-pixelr.jpg") | ... .save : ……………………………………………………….. |
| img.show() | … .show(): ……………………………………………………….. |

Regardez attentivement le centre de l'image, agrandir l’image si

De quelle couleur est le pixel modifié ? …………………..

Pour diviser un nombre n

par 3et obtenir la partie entière

**n // 3**

|  |  |
| --- | --- |
| Modifiez le programme afin de colorier en noir un carré de 3 pixels noirs de coté au centre de la photo et enregistrez le sous le nom  **«pomme-pourrie.py »** |  |

**Attention,** ce programme doit fonctionner quelque soit la largeur et la hauteur de l’image choisie !

Réécrire les lignes modifiées ou ajouter

|  |  |
| --- | --- |
|  |  |

Tester votre programme sur l’image « orange.jpg » en le modifiant et en l’enregistrant sous le nom

Réécrire les lignes modifiées ou ajouter

|  |  |
| --- | --- |
|  |  |

En utilisant le code suivant, construire le favicon « mario »

|  |  |
| --- | --- |
| from PIL import Image  #Creation d'une image blanche RVB dimension 8x8  Mario =Image.new("RGB",(8,8),(255,255,255))  #definition d'une couleur  rouge=(255,0,0)  #remplissage de l'image pixel par pixel  Mario.putpixel((0,2),rouge)  Mario.putpixel((7,7),(0,255,0))  Mario.show()  Mario.save("Mario.jpg") |  |
|  |

Qu’est ce qu’un favicon ? …………………………………………………………………………………………

……………………………………………………………………………………………………………………

**Activité 3 :**

Saisir, enregistrer le programme sous le nom **rvbpixel-1.py** et tester le programme suivant

* Vous devez respecter les décalages avec la touche tab à coté de la touche « a »
* ATTENTION : *l'exécution de ce programme n'est pas très intéressante en soi, vous pouvez l'arrêter à tout moment en appuyant simultanément sur la touche Ctrl et sur la touche C*

from PIL import Image

Expliquer en quelques mots ce que fait ce programme ?

img = Image.open("pomme.jpg")

largeur\_image=img.width

hauteur\_image=img.height

for y in range(hauteur\_image):

for x in range(largeur\_image):

r,v,b=img.getpixel((x,y))

print("rouge : ",r,"vert : ",v,"bleu : ",b)

print("fin")

**Activité 4 :**

Expliquer en quelques mots ce que fait ce programme ?

Saisir, enregistrer le programme sous le nom

**pomme-chg-couleur-1.py**

et tester le programme suivant

from PIL import Image

img = Image.open("pomme.jpg")

largeur\_image=img.width

hauteur\_image=img.height

for y in range(hauteur\_image):

for x in range(largeur\_image):

r,v,b=img.getpixel((x,y))

n\_r=v

n\_v=b

n\_b=r

img.putpixel((x,y),(n\_r,n\_v,n\_b))

img.show()

Modifier le programme de façon à enregistrer la nouvelle image sous le nom pomme-*couleur*.jpg Attention, remplacer *couleur* par le nom de la couleur dans le nom du fichier.

|  |  |
| --- | --- |
| - Comment obtenir une pomme bleue ?  Enregistrer la nouvelle image sous le nom  pomme-bleue.jpg | Saisir les 3 lignes modifiées  n\_r= …  n\_v=…  n\_b=… |
| - Quelle autre couleur pourrions-nous donner à la pomme ?  Enregistrer la nouvelle image sous le nom  pomme-………… .jpg | Saisir les 3 lignes modifiées  n\_r= …  n\_v=…  n\_b=… |

Appliquer ce programme, sur l’image perroquet.jpg. Recherchez la modification qui vous plait le plus puis enregistrer cette nouvelle image sous le nom perroquet-change.jpg

**Activité 5 :** À partir du programme précédent et après avoir fait quelques recherches sur le "négatif d'une image", écrire un programme qui donne le négatif d'une image.

L’appliquer sur les images pomme et perroquet

Comment avez-vous fait ? Ecrire les lignes modifiées

|  |  |
| --- | --- |
| Peut-on modifier le fond noir et le rendre blanc tout en ayant la pomme en négatif ? | if …… == … **:**  instruction  else :  **instruction** |

****

**Activité 6 :** Comme pour l’activité 5, écrire un programme qui transforme une "image couleur" en une "image en niveau de gris".

**NB :** en Python pour avoir une division entière, il faut utiliser l'opérateur // à la place de l'opérateur /

**Activité 7 :** Appliquons un filtre sur la pomme verte

* Un filtre rouge sur la pomme verte : seule la couleur rouge est visible, les autres couleurs sont noires..
* Un filtre cyan sur la pomme verte : seules les couleurs …………… et ……………….. sont visibles. La couleur ……………. est noire.

|  |  |
| --- | --- |
| Filtre rouge | **Filtre cyan** |

**Activité 8 :** Modifiez qu’une partie de l’image

|  |  |
| --- | --- |
| A partir du fichier « perroquet.jpg »  Modifiez cette image de façon à obtenir au centre de l’image un carré de dimension 200 x 200 dont l’intérieur est en négatif.  Tout le reste est inchangé !  Enregistrer l’image créée sous le nom  perroquet-centre-negatif.jpg  Enregistrez votre programme sous le nom  perroquet-centre-negatif.py | **Fonction utile :**  if …… < >== … and …… < >== … **:**  instruction  else :  instruction |

Recopiez votre programme :

|  |  |
| --- | --- |
|  |  |

**Activité 9 :** Donnez libre court à votre imagination à partir de l’image perroquet.jpg

Par exemple :

- Cadre de autour de l’image plus foncé - Une partie en négatif

- Une partie en noir et blanc …

|  |  |
| --- | --- |
| **Fonctions utiles :** | |
|  | **if** variable **<** **20:**  instruction  **elif** variable **<** **100:**  instruction2  **elif** variable **<** **250:**  instruction3  **else:**  instruction4 |
|  | **if** …… < >== … **and** …… < >== … **:**  instruction1  **else** :  instruction2 |
|  | **while** choix **<** **2** **or** choix **>** **10:**  choix **=** float**(**input**(**"Choissez un nombre entre 2 et 10 : "**))** |
|  | **while** reponse **!=** n**:**  reponse **=** int**(**input**(**"Entrez le numéro de votre réponse. "**))** |