Un enseignement au numérique pour notre académie : Le traitement d'une image numérique

Le traitement d'une photographie pour faire des retouches ou pour la déformer de façon à la rendre plus rigolote est un geste accessible à tous via l'utilisation de « petites » applications disponibles sur tablettes et téléphones portables. Qu'y-a-t-il derrière ces applications ? Quel est ce procédé magique qui - en une ou deux manipulations - permet d'obtenir une image modifiée ? C'est ce que nous souhaitons vous faire découvrir ici grâce à une image interpellante.

Les valeurs de notre République vivent par nos actes quotidiens et par le sens profond que nous donnons à notre devise : **Liberté – Égalité - Fraternité**. Il est intéressant d'analyser comment l'allégorie est un moyen de s'approprier un sentiment, une qualité et de voir combien l'image d'une valeur peut être un moyen d'assurer sa présence matérielle dans notre univers quotidien.

Le peintre Eugène Delacroix a été l'auteur de l'œuvre majeure « *La Liberté guidant le peuple* ». Ce tableau personnifie de façon magistrale la puissance de la Liberté dans un contexte où tout semble perdu ; la Liberté porte ainsi sans cris notre drapeau et invite – sans contraindre – à se battre contre la tyrannie.

Source : Site Wikipédia Autoportrait d'Eugène Delacroix

Cette brève introduction montre combien l'idée de faire travailler les élèves sur un pan du numérique peut aussi les inviter à partager une des valeurs centrales de notre République.

Nous allons au début de cet article revisiter l'œuvre précitée de Delacroix ; l'image que nous faisons figurer ci-contre n'est cependant pas celle du tableau d'Eugène Delacroix. Remarquons en effet que sur cette image, la Liberté porte notre drapeau de sa main gauche, que le garçon à l'écharpe est placé à sa droite et que l'homme à la carabine est posté à sa gauche. Voici ci-dessous mises côte-à-côte la photographie (placée à gauche) du tableau de Delacroix et notre image (à droite) :

Source : site Wikipédia

Si l'image de gauche est la photographie d'un tableau bien réel accroché sur un mur du musée du Louvre, l'image de droite ne correspond – en ce qui la concerne - à aucune réalité. En d'autres termes, l'image de droite ne représente aucun objet concret. C'est exactement comme lorsque l'on se place devant un miroir où face au miroir nous avons la réalité et dans le miroir une image créée de toute pièce à partir des lois de la physique et en l'occurrence de l'optique.

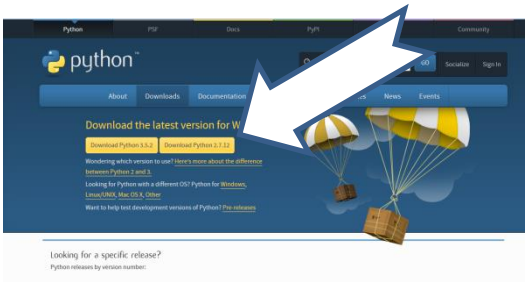
L'image de droite a été créée point par point (pixel par pixel) à partir de la photographie de gauche. Nous allons vous montrer comment réaliser de A à Z le traitement d'image qui permet à partir d'une image numérique (du type de la photographie de gauche) d'obtenir une image « symétrique » (comme l'image de droite). Vous pourrez à la fin de l'article construire une image correspondant au traitement d'une de vos photographies préférées.

1ère étape : sauvegarde sur votre bureau d'une photographie au format jpg

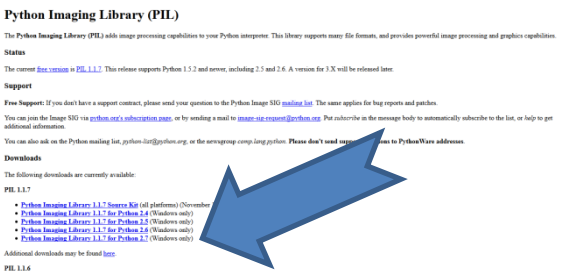
Nous prenons une photo que nous sauvegardons sur le bureau de notre ordinateur, en cliquant droit sur la photo, on obtient le lien vers cette photo que l'on « copie-colle » afin de le conserver dans le presse-papier. Ce lien est du type : C:\Users\Votre nom d'utilisateur\Bureau\Maphoto.jpg

2ième étape : téléchargement d'un logiciel et d'une bibliothèque

Nous vous proposons à présent de télécharger un logiciel de programmation qui s'appelle Python (comme le serpent), attention à bien télécharger sa version 2.7.12.
Pour ce faire, utiliser le lien :
https://www.python.org/downloads/



Il faut ensuite aller chercher ce que l'on appelle un module Python c'est-à-dire une bibliothèque de petits programmes Python que l'on pourra ainsi utiliser à souhait. Ce module s'appelle PIL, il est accessible à l'adresse suivante :
http://www.pythonware.com/products/pil/
Attention à bien sélectionner la version de PIL correspondant à Python 2.7.



Résultat des opérations : vous avez sur votre ordinateur le logiciel Python et sa bibliothèque PIL.

3ième étape : implémentation du programme Python

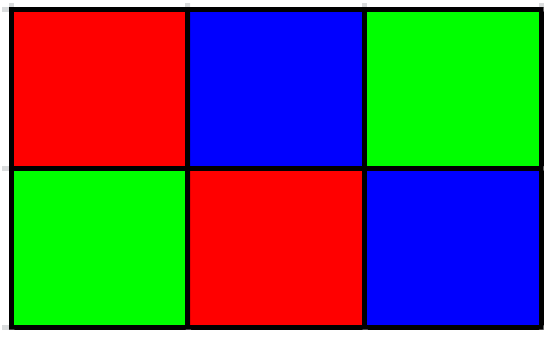
Il vous suffit à présent de lancer le logiciel Python et d'entrer les instructions du tableau suivant :

```
>>> adresse= ' C:\Users\Votre nom d'utilisateur\Bureau\Maphoto.jpg '
>>> from PIL import Image
>>> photo=Image.open(adresse)

>>> def imagesym(imagedepart):
        colonne,ligne=imagedepart.size
        imagearrivee=Image.new(imagedepart.mode,imagedepart.size)
        for i in range(ligne):
            for j in range (colonne):
                pixel=imagedepart.getpixel((j,i))
                imagearrivee.putpixel((colonne-j-1,i),pixel)
        imagearrivee.show()

>>> imagesym(photo)
```

Explications du processus

Choix d'une image plus simple

Une image numérique est formée de petits carrés que l'on appelle des pixels, ces pixels placés côte-à-côte forment à eux tous réunis une mosaïque, la photographie au format jpg du tableau de Delacroix est ainsi formée de 5946 colonnes et de 4771 lignes soit 28 368 366 pixels.



Afin d'expliquer le processus que le programme *imagesym* a mis en œuvre, nous allons considérer une image beaucoup plus simple que la photographie du tableau de Delacroix. L'image que nous prenons est celle qui figure ci-contre, elle est simple pour deux raisons :

- d'une part, elle n'est formée que de 3 colonnes et 2 lignes soit 6 pixels ;
- d'autre part ses couleurs sont des plus simples que l'on puisse construire en format RGB : Red (255,0,0), Green (0,255,0), Blue (0,0,255).

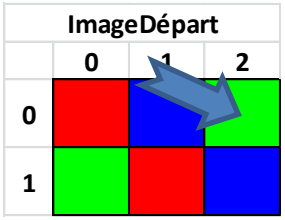
Décortication du traitement de l'image

Nous allons nommer *Image de départ* notre image 3x2 et construire pas à pas une nouvelle image que nous allons nommer *Image d'arrivée*.

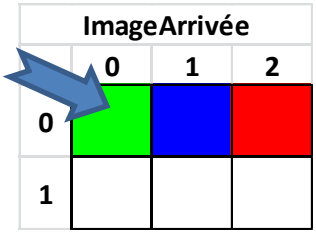
<table>
<tr><th>Traitement</th><th>Résultat</th></tr>
<tr><td>Au départ, l'image d'arrivée est vide. Celle-ci est uniquement constituée des mêmes éléments que l'image de départ :
- son format est le format jpg;
- ses dimensions sont 3x2.</td><td>ImageArrivée
<table><tr><th></th><th>0</th><th>1</th><th>2</th></tr><tr><th>0</th><td></td><td></td><td></td></tr><tr><th>1</th><td></td><td></td><td></td></tr></table></td></tr>
<tr><td colspan="2">Le programme considère la première ligne de l'image de départ numérotée 0 (i=0) :</td></tr>
<tr><td>1) Il considère tout d'abord la première colonne numérotée 0 de cette ligne (j=0) et note sa couleur rouge (255, 0 ,0) ;
ImageDépart
<table><tr><th></th><th>0</th><th>1</th><th>2</th></tr><tr><th>0</th><td></td><td></td><td></td></tr><tr><th>1</th><td></td><td></td><td></td></tr></table></td><td>2) Il reporte cette couleur au pixel de coordonnées (3-0-1,0), c'est-à-dire au pixel de coordonnées (2,0). Le couple (2,0) signifie que ce pixel se trouve sur la colonne numérotée 2 et sur la ligne numérotée 0.
ImageArrivée
<table><tr><th></th><th>0</th><th>1</th><th>2</th></tr><tr><th>0</th><td></td><td></td><td></td></tr><tr><th>1</th><td></td><td></td><td></td></tr></table></td></tr>
<tr><td colspan="2">Toujours sur la première ligne (i=0) :</td></tr>
<tr><td>1) Le programme considère la deuxième colonne numérotée 1 (j=1) de cette ligne et note sa couleur bleue (0, 0 , 255) ;
ImageDépart
<table><tr><th></th><th>0</th><th>1</th><th>2</th></tr><tr><th>0</th><td></td><td></td><td></td></tr><tr><th>1</th><td></td><td></td><td></td></tr></table></td><td>2) Il reporte cette couleur au pixel de coordonnées (3-1-1,0), c'est-à-dire au pixel de coordonnées (1,0).
ImageArrivée
<table><tr><th></th><th>0</th><th>1</th><th>2</th></tr><tr><th>0</th><td></td><td></td><td></td></tr><tr><th>1</th><td></td><td></td><td></td></tr></table></td></tr>
</table>

Toujours sur la première ligne (**i=0**) :

1) Le programme considère la troisième colonne numérotée **2** (**j=2**) de cette ligne et note sa couleur verte (0,255,0) ;

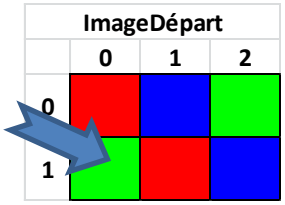


2) Il reporte cette couleur au pixel de coordonnées (3-**2**-1,**0**), c'est-à-dire au pixel de coordonnées (0,0).

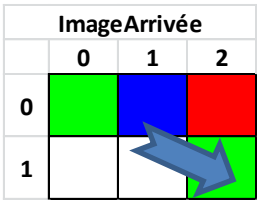


Le programme considère à présent la deuxième ligne numérotée **1** (**i=1**) de l'image de départ :

1) Il considère tout d'abord la première colonne numérotée **0** (**j=0**) de cette ligne et note sa couleur verte (0,255,0);

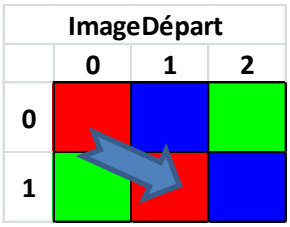


2) Il reporte cette couleur au pixel de coordonnées (3-**0**-1,**1**), c'est-à-dire au pixel de coordonnées (2,1).

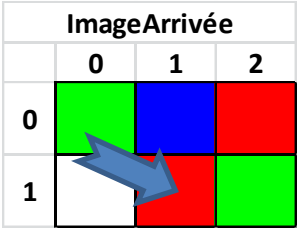


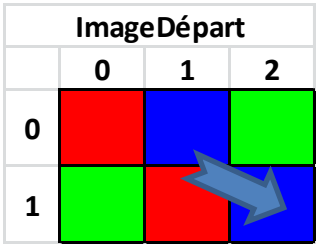
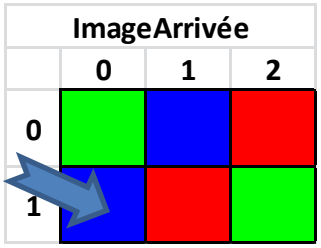
Toujours sur la deuxième ligne (**i=1**) :

1) Le programme considère la deuxième colonne de cette ligne (**j=1**) et note sa couleur rouge (255,0,0) ;

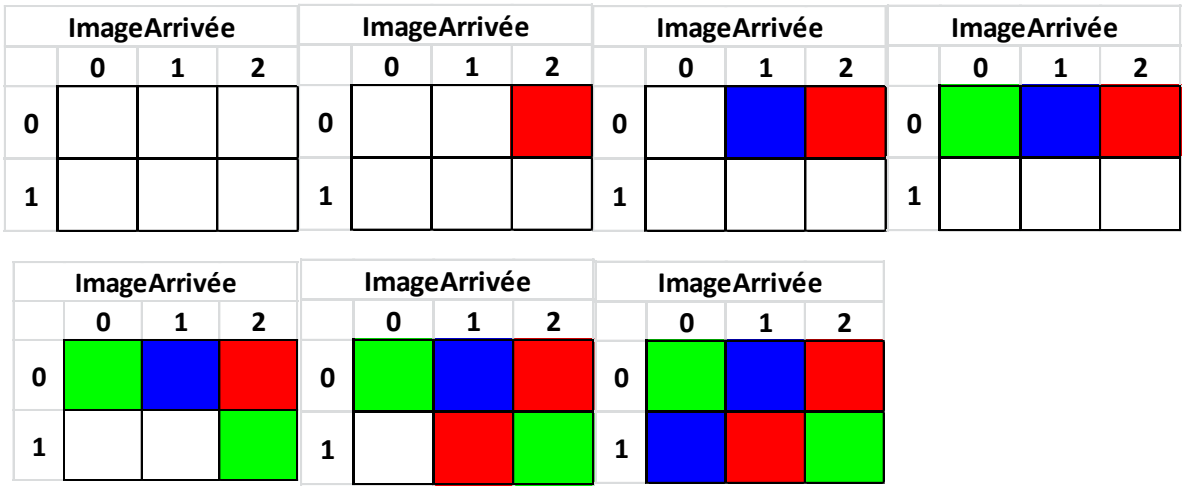


2) Il reporte cette couleur au pixel de coordonnées (3-**0**-1,**1**), c'est-à-dire au pixel de coordonnées (1,1).



<table>
<tr><td colspan="2">Toujours sur la deuxième ligne (i=1) :</td></tr>
<tr><td>1) Le programme considère la troisième colonne de cette ligne (j=2) et note sa couleur bleue (0,0,255).
 
</td><td>2) Il reporte cette couleur au pixel de coordonnées (3-2-1,1), c'est-à-dire au pixel de coordonnées (0,1).
 
</td></tr>
</table>

Pour chaque ligne (allant du numéro 0 au numéro 2), toutes les colonnes (allant du numéro 0 au numéro 2) ont été traitées. Le programme peut donc renvoyer l'image d'arrivée. Si nous refaisons le film de la construction de l'image d'arrivée, voici ce que nous obtenons :



Cette opération - très laborieuse lorsqu'elle est effectuée à la main - est réalisée en deux temps trois mouvements par l'ordinateur. C'est donc ici que l'on voit deux côtés pratiques du numérique : d'une part, on peut confier à l'ordinateur une tâche pour le moins rébarbative qui consiste à effectuer sans relâche une action répétitive qui va être réalisée en boucle et d'autre part, la vitesse de traitement par l'ordinateur de la tâche qui lui est confiée permet d'obtenir l'image d'arrivée en au plus quelques minutes (ce temps d'attente dépend entre autres de la vitesse de votre microprocesseur).

Explication dans le détail de toutes les instructions demandées à l'ordinateur

Dans le tableau suivant nous allons dans la colonne de gauche reprendre toutes les instructions fournies précédemment et dans la colonne de droite expliquer l'une après l'autre ces instructions.

Instructions	Explications

>>> adresse= ' C:\Users\Votre nom d'utilisateur\Bureau\Maphoto.jpg '
>>> from PIL import Image
>>> photo=Image.open(adresse)
	Ces trois instructions permettent successivement :
	- d'aller chercher la photographie à son emplacement ;
	- d'activer la fonction Image de la bibliothèque PIL ;
	- de définir l'image nommée Photo.

>>> def imagesym(imagedepart):
 	
colonne,ligne=imagedepart.size
 	
imagearrivee=Image.new(imagedepart.mode,imagedepart.size)
 	
for i in range(ligne):
for j in range (colonne):
pixel=imagedepart.getpixel((j,i))
imagearrivee.putpixel((colonne-j-1,i),pixel)
	Ces huit lignes constituent le programme central de notre article, c'est-à-dire un lot d'instructions regroupées qui seront activées successivement lorsqu'on fera exécuter ce programme sur une image (c'est pour cela que l'on parle plutôt d'une fonction qui appliquée à une image va renvoyer en l'occurrence une autre image).
	
	On définit le nom de la fonction imagesym et la variable à laquelle cette fonction s'applique imagedepart.
	
	On rapatrie les dimensions de notre variable (ce sont les dimensions de l'image à laquelle on appliquera la fonction).
	
	On définit une image nommée imagearrivee qui a le même format que notre variable et les mêmes dimensions.
	
	Le programme effectue **une boucle**, c'est-à-dire qu'il effectue de façon répétitive une succession d'instructions identiques. On parcourt ainsi tous les pixels de l'image de départ et pour chaque pixel :
	- on lit le code couleur de ce pixel ;
	- on reporte ce code sur son pixel symétrique.

imagearrivee.show()	On demande au programme d'afficher l'image d'arrivée, c'est-à-dire l'image que nous venons de construire point par point.
>>> imagesym(photo)	Le lancement du programme se fait en prenant le nom du programme *imagesym* et en mettant comme argument le nom de la photographie sur laquelle nous souhaitons faire opérer notre fonction ; ici la photographie s'appelle *photo*.

La gestuelle opaque du numérique

Le numérique peut nous aider à être créatif avec des résultats qui frisent le réalisme et qui ressemblent - à s'y méprendre - à la réalité. À l'image de ce que réalisait l'homme préhistorique sur les parois des cavernes, tout à chacun d'entre nous peut ainsi produire aujourd'hui des œuvres « intéressantes ». Une simple différence est cependant à prendre en compte : si pour l'homme préhistorique, la main était le pinceau de l'artiste et en ce sens l'artiste maîtrisait parfaitement tous les gestes effectuées par cette main, il n'en va pas de même du numérique où l'on

peut constater qu'il est souvent assez délicat de décortiquer - comme nous l'avons fait - la gestuelle opaque du numérique, gestuelle réduite à quelques clics qui laisse penser à tort à l'artiste qu'il n'y a pas grand-chose derrière ces clics.

Retour sur l'allégorie de la Liberté

Le 7 janvier 2015, au nom d'une idéologie que nous refusons catégoriquement deux individus ont voulu mettre à mal notre démocratie. Rappelons que lors de cet événement dramatique, le clermontois **Michel Renaud** a perdu la vie au siège du journal **Charlie Hebdo**, haut lieu de notre Liberté d'expression.

Source : site du Journal La Montagne
*Auteur : **Franck Boileau***
Autorisation du 12 décembre 2016

Source : site du Journal Le Monde
*Auteur : **Jean Plantureux dit Plantu***
Autorisation du 10 décembre 2016

Dans le numéro du journal Le Monde du 9 janvier 2015, Jean Plantureux dit Plantu a repris le tableau de Delacroix pour nous fournir cette image remarquable intitulée : ***« La Liberté sera toujours la plus forte »***. C'est sur cette note optimiste - qu'il nous appartient à tous de défendre - que nous concluons cet article.

Jean-Alain Roddier
IA-IPR de mathématiques
Référent du dossier sur la formation
au numérique des élèves
Académie de Clermont-Ferrand