



TRAITEMENT D'IMAGES

Contenus et capacités

Notions transversales de programmation

Contenus	Capacités attendues
Affectations, variables Séquences Instructions conditionnelles Boucles bornées et non bornées Définitions et appels de fonctions	Écrire et développer des programmes pour répondre à des problèmes et modéliser des phénomènes physiques, économiques et sociaux.

La photographie numérique

Contenus	Capacités attendues
Traitement d'image	Traiter par programme une image pour la transformer en agissant sur les trois composantes de ses pixels.

Note d'intention

Cette ressource propose un parcours d'apprentissage pour amener l'élève à programmer le négatif d'une image en noir et blanc (niveau de gris).

D'une durée globale de 6 heures, ce parcours se décompose en deux étapes.

La première étape d'une durée de 3 heures consiste en une activité qu'il est possible de réaliser sur les heures de mathématiques. Son objectif est, pour un élève de seconde de prendre en main le langage python et de travailler les concepts algorithmiques de base : séquence, affectation, boucle pour. Les plus rapides pourront s'initier à la programmation de la double boucle qu'il sera nécessaire de manipuler pour la retouche d'images.

La seconde étape, d'une durée de 3 heures, se découpe en deux sous parties, la première consiste à se familiariser avec la bibliothèque PIL en présentant les instructions de base tout en revisitant la double boucle. La seconde partie se focalise sur la retouche d'image : négatif en noir et blanc. Les plus rapides pouvant tout à fait aller plus loin.

Retrouvez éduscol sur :



Le choix de l'environnement de programmation pour cette fiche est Edupython. La bibliothèque PIL y étant incluse par défaut.

Régulièrement les élèves seront invités à aller « taper » directement une instruction dans la console pour en voir l'effet (pour toutes les instructions ci-après).

Volontairement, nous n'utilisons pas l'image de Lena (photo la plus téléchargée sur Internet). Elle a certes un côté historique concernant la retouche d'images, mais nous avons considéré l'enjeu de féminisation de l'informatique plus important. Nous travaillons donc principalement sur la Joconde à partir d'une photo libre de droit issue de wikimedia.

[Cliquer ici pour obtenir les fichiers annexes à cette ressource.](#)

Étape 3 - traitement d'images

Liste des fichiers attachés

- Retouche.py
- Joconde_384.jpg
- Joconde_384.png
- Papillon.png
- Joconde_10.png
- Joconde_gris.png
- Joconde.png

Exercice 1 - quelques commandes en plus

1. Dans votre espace personnel, créer un dossier contenant toutes les images et le fichier retouche.py.
2. Ouvrir le fichier retouche.py avec Edupython et enregistrer le dans le même dossier que celui qui contient les quatre images.
3. Après avoir exécuté le fichier retouche.py, déterminer ce que fait la commande `.open`
4. Préciser le type de `img`.

Commentaires

Cet exercice présente la méthode `.open` permettant d'ouvrir une image du disque dur et de l'importer dans une « ImageFile » manipulable en python.

Exercice 2 - papillon

Dans cet exercice nous allons travailler sur l'image *papillon.png* qui est en noir et blanc.

Préalable

Modifier le programme retouche.py de manière à importer le papillon dans une ImageFile que nous appellerons pap.

Activité

Exécuter le programme retouche.py.

Retrouvez éduscol sur :



Grâce à l'interpréteur, déterminer ce que font les commandes suivantes :

```
> > > pap.format
> > > pap.size
> > > pap.show()
> > > pap.getpixel((3,7))
> > > pap.putpixel((3,7),128)
> > > pap.mode
> > > pap2 = pap.copy()
```

Commentaires

Présentation de la méthode *getpixel* et de l'attribut *mode* permettant de connaître la manière dont sont encodés les pixels (L : niveau de gris, RGB pour un codage couleur (R,G,B)).

Exercice 3 - Négatif d'une image en noir et blanc

1. Effectuer le négatif de l'image papillon.png.
2. Effectuer le négatif de l'image Joconde_10.png.
3. Effectuer le négatif de l'image Joconde_gris.png.

Correction python

```
# renvoie le négatif du papillon
pap = Image . open ( "papillon . png" )
for l in range (10) :
    for c in range (10) :
        valeur = pap.getpixel (( c , l ))
        if valeur == 0 :
            pap.putpixel (( c , l ),255)
        else :
            pap.putpixel (( c , l ),0)
pap . save ( "image.png" , "png" )

# renvoie le négatif de Joconde_10.png
joco = Image . open ( "Joconde_10. png" )
for l in range (10) :
    for c in range (10) :
        valeur = joco.getpixel (( c , l ))
        nouvelle_valeur = 255 - valeur
        joco.putpixel (( c , l ) , nouvelle_valeur )
joco.save ( " image.png" , "png" )

# renvoie le négatif de Joconde_gris.png
joco = Image.open ( "Joconde_gris.png" )
for l in range (384) :
    for c in range (384) :
        valeur = joco.getpixel (( c , l ))
        nouvelle_valeur = 255 - valeur
        joco.putpixel (( c , l ) , nouvelle_valeur )
joco.save ( "image.png" , "png" )
```

Retrouvez éduscol sur :



Commentaires

C'est l'exercice « cœur de séance » : construire le négatif d'une image.

La première image est une image 10 par 10 en noir et blanc sans niveau de gris (0 ou 255). Des élèves vont le faire pixel par pixel sans utiliser `getpixel` mais en regardant l'image. Il est possible dans un premier temps de les laisser faire et de les questionner par la suite pour une grosse image (question 3). D'autres élèves utiliseront `getpixel` combiné à un if (si `val = 255` alors mettre 0 sinon mettre 255). Il ne faut pas dans la question 1 s'attendre à ce que tous les élèves pensent à faire `255-val`. C'est l'objectif de la question 2.

La deuxième image est une image 10 par 10 de la Joconde en niveau de gris. Il y a ici l'obligation d'utiliser la formule `255-val`. La double boucle peut alors être rappelée à tous.

La troisième image est beaucoup plus grosse. Mais toutefois, le programme est identique.

Volontairement dans cet exercice, toutes les images sont carrées afin de ne pas perturber les élèves (pas de différenciation entre hauteur et largeur). L'objectif étant vraiment la double boucle pour et la gestion du négatif.

Exercice 4 - comprendre les images couleurs

1. Importer `Joconde_384.jpg` dans une Imagefile que nous appellerons `joco`.
2. Exécuter le fichier.
3. Tester et comprendre les commandes suivantes dans l'interpréteur :


```
> > > valeurs = joco.getpixel((3,7))
> > > valeurs
> > > valeurs[0] = 35
> > > valeurs
> > > (R,G,B) = joco.getpixel((3,7))
> > > G
```
4. Expliquer comment fonctionne `joco.getpixel`.

Exercice 5 - transformer une image couleur en noir et blanc

Commentaires

Sont introduits ici les images couleurs et le `.getpixel` les manipulant de deux manières différentes. Pour les élèves manipuler la récupération en écrivant `(R,G,B) = joco.getpixel((3,7))` est plus naturelle.

1. Traiter `Joconde_384.jpg` en remplaçant, pixel par pixel, chacune des valeurs des composantes des couleurs par la moyenne des trois.
2. Dans `Joconde_384.pgm`, toujours pixel par pixel, remplacer chaque composante des couleurs par la valeur commune :
 - $0,21 \times R + 0,71 \times G + 0,07 \times B$
3. Répéter la même opération mais en prenant cette fois-ci :
 - $0,299 \times R + 0,587 \times G + 0,114 \times B$
4. Comparer les trois images obtenues.

Retrouvez éduscol sur :



Correction python

transforme la Joconde couleur en noir et blanc par moyenne des pixels

```
joco = Image.open ("Joconde_384.jpg")
```

```
for l in range (384) :
```

```
    for c in range (384) :
```

```
        valeurs = joco.getpixel (( c , l ) )
```

```
        nouveaux = ( valeurs [0]+ valeurs [1]+ valeurs [2]) //3
```

```
        joco.putpixel (( c , l ) ,( nouveaux , nouveaux , nouveaux ) )
```

```
joco.save ("image.png" ,"png")
```

#transforme la jonconde couleur en noir et blanc avec 0,21 R + 0,71 G + 0,07 B

```
joco = Image.open ("Joconde_384.jpg")
```

```
for l in range (384) :
```

```
    for c in range (384) :
```

```
        valeurs = joco.getpixel (( c , l ) )
```

```
        nouveaux = 0.21* valeurs [0]+0.71* valeurs [1]+0.07* valeurs [2]
```

```
        nouveaux = int ( nouveaux )
```

```
        joco.putpixel (( c , l ) ,( nouveaux , nouveaux , nouveaux ) )
```

```
joco.save ("image.png" ,"png")
```

#transforme la Joconde couleur en noir et blanc avec 0,299 R + 0,587 G + 0,114 B

```
joco = Image.open ("Joconde_384.jpg")
```

```
for l in range (384) :
```

```
    for c in range (384) :
```

```
        (R,G , B ) = joco.getpixel (( c , l ) )
```

```
        nouveaux = 0.21* R +0.71* G +0.07* B
```

```
        nouveaux = int ( nouveaux )
```

```
        joco.putpixel (( c , l ) ,( nouveaux , nouveaux , nouveaux ) )
```

```
joco.save ("image.png" ,"png")
```

Commentaires

Exercice classique et basique de manipulation des images couleurs.

Exercice 6 - négatif d'une image en couleurs

Effectuer le négatif de l'image Joconde_384.jpg.

Correction python

```
#Négatif de la Joconde en couleurs
joco = Image.open ( "Joconde_384. jpg" )
for l in range (384) :
    for c in range (384) :
        (R ,G , B ) = joco.getpixel (( c , l ))
        nouveaux = (255 - R ,255 - G ,255 - B )
        joco.putpixel (( c , l ) , nouveaux )
joco.save ( "image.png" , "png")
```

Commentaires

Exercice peu difficile car pas de nouvelles réflexions algorithmiques. Les problèmes pour les élèves viendront de la manipulation de la syntaxe en python.

Exercice 7 - miroir

1. Effectuer le miroir de l'image papillon.png.
2. Effectuer le miroir de l'image Joconde_gris.png.
3. Effectuer le miroir de l'image Joconde_384.jpg.

Correction python

```
#miroir du papillon
pap = Image . open ( "papillon.png" )
pap2 = pap.copy ()
for l in range (10) :
    for c in range (10) :
        valeur = pap.getpixel (( c , l ))
        pap2.putpixel ((9 - c , l ) , valeur )
    pap2.save ( "image.png" , "png")

#miroir Joconde noir et blanc
joco = Image.open ( "Joconde_gris.png" )
joco2 = joco.copy ()
for l in range (384) :
    for c in range (384) :
        valeur = joco.getpixel (( c , l ))
        joco2.putpixel ((383 - c , l ) , valeur )
joco2.save ( "image.png" , "png" )
```

Retrouvez éduscol sur :



```
#miroir Joconde couleur
joco = Image.open ( " Joconde_384. jpg" )
joco2 = joco.copy ()
for l in range (384) :
    for c in range (384) :
        valeurs = joco.getpixel (( c , l ) )
        joco2.putpixel ((383 - c , l ) , valeurs )
joco2.save ( "image.png" , "png")
```

Commentaires

La difficulté de cet exercice vient du fait que les élèves modifient leur image de départ. Ainsi, arrivés à la moitié, ils ont perdu l'information de la partie gauche. Il est alors nécessaire de faire une copie de l'image de départ, ce qui constitue une réelle difficulté pour eux, au même titre que travailler avec les indices.

Exercice 8 - images non carrées

4. Effectuer le négatif de joconde.png.
5. Effectuer le miroir de joconde.png.