

**VOIE GÉNÉRALE**

**ET TECHNOLOGIQUE**

**2<sup>DE</sup>**
**1<sup>RE</sup>**
**T<sup>LE</sup>**

*Sciences numériques et technologie*

ENSEIGNEMENT  
COMMUN

## APPRENDRE À MANIPULER UNE IMAGE NUMÉRIQUE

### *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 2 - apprendre à manipuler une image numérique

### Exercice 1 - dans l'interpréteur d'Edupython

1. Ouvrir un nouveau fichier python avec Edupython que vous appellerez *depart.py*. Sauvegarder le dans votre dossier personnel, puis exécuter le.
2. Dans l'interpréteur d'Edupython taper :

```
>>> from PIL import Image
>>> im = Image.new("RGB", (600,300), "grey")
>>> type(im)
>>> im.save("trace.png", "PNG")
```
3. Préciser à quoi sert la fonction *Image.new* et expliquer comment elle fonctionne. Ne pas hésiter à regarder ce qu'il se passe dans le dossier contenant le fichier *depart.py*.
4. Préciser à quoi sert la méthode *.save* et expliquer comment elle fonctionne.

#### Commentaires

Sauf paramétrage préalable, *edupython* n'enregistre pas le script dans le dossier de l'élève. Comme ici nous allons manipuler des fichiers sur le disque il est impératif que le module python dans lequel l'élève travaille soit dans le même dossier que les images. D'où cette petite manipulation pour que l'interpréteur « travaille » dans le bon dossier. Si ce n'est pas fait un message peut survenir stipulant que l'on n'a pas les droits d'écriture.

### Exercice 2 - toujours dans l'interpréteur d'Edupython

1. Dans l'interpréteur python taper :

```
>>> im.putpixel( (10,10), (0,0,0) )
>>> im.save("trace.png", "PNG")
```
2. Préciser à quoi sert la méthode *.putpixel* et expliquer comment elle fonctionne.
3. Indiquer pourquoi *.save* il indispensable.

Retrouvez éducol sur :



### Commentaires

Il est fondamental de faire distinguer à l'élève l'image avec laquelle python va travailler qui est un objet python de l'image sur le disque dur. Si l'on ne prend pas le temps de sauver *im*, il n'y a aucune raison que l'image du disque soit modifiée.

## Exercice 3 - premier programme

1. Dans le bloc note d'Edupython, écrire un programme qui créera sur le disque dur une image toute jaune de taille 800 par 800 pixels. Cette image devra s'appeler *premiere\_image.png*
2. Modifier le programme pour que l'image contienne un pixel noir, un bleu, un rouge, un vert et un blanc.
3. Modifier votre programme pour que votre image contienne un pixel de couleur « skyblue ».
4. Comparer le résultat si nous avons décidé d'appeler notre image *premiere\_image.jpg*. Expliquer pourquoi les résultats sont différents.

### Commentaires

Ici l'élève est invité à manipuler les commandes qu'il vient de voir. La décomposition d'une couleur selon le codage (RGB) est travaillée, les valeurs de « skyblue » se trouvant facilement sur le web.

Pour avoir une image jpg la commande doit être `im.save("trace.png", "JPEG")` et non `im.save("trace.png", "PNG")`. Dans le deuxième cas, nous aurions une image s'appelant *trace.jpg* dont Windows considèrera qu'elle est au format *.jpg* alors qu'elle sera en fait au format *.png*.

La différence de visu, si les élèves ont mis les valeurs des pixels modifiées au même endroit est flagrante.

L'image jpg est floue (format avec compression) au contraire du png (format sans compression). Une image enregistrée au format *.jpg* prenant moins de place qu'une au format *.png*. Ce que nous ne constatons malheureusement pas ici en raison de notre image très particulière (toute jaune).

## Exercice 4

À partir d'une image carrée de taille 800 par 800 dont le fond est gris, tracer un segment de droite. L'objectif est d'obtenir un résultat comme ci-dessous :



Retrouvez éduscol sur :



**Correction python**

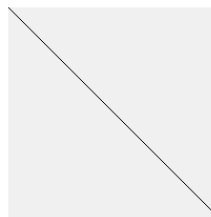
```
im = Image . new ( "RGB" , (800 ,800) , "grey" )
for k in range (800) :
    im . putpixel (( k ,400) ,(255 ,255 ,255) )
im . save ( "image.png" , "png" )
```

**Commentaires**

L'élève doit mobiliser une instruction de parcours *for*.

**Exercice 5**

À partir d'une image carrée de taille 800 par 800 dont le fond est gris, tracer un segment de droite. L'objectif est d'obtenir un résultat comme ci-dessous :

**Correction python**

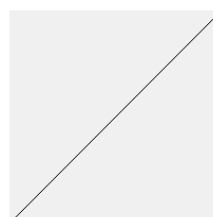
```
im = Image . new ( "RGB" , (800 ,800) , "grey" )
for k in range (800) :
    im . putpixel (( k ,k) ,(0 ,0 ,0) )
im . save ( "image.png" , "png" )
```

**Commentaires**

Avec une boucle *pour*, l'exercice est facilité par l'utilisation d'une image carrée.

**Exercice 6**

À partir d'une image carrée de taille 800 par 800 dont le fond est gris, tracer un segment de droite. L'objectif est d'obtenir un résultat comme ci-dessous :



Retrouvez éduscol sur :



**Correction python**

```
im = Image . new ( "RGB" , (800 ,800), "grey" )
for k in range (800) :
im . putpixel ((799 - k, k) ,(0 ,0 ,0) )
im . save ( "image.png", "png" )
```

**Exercice 7**

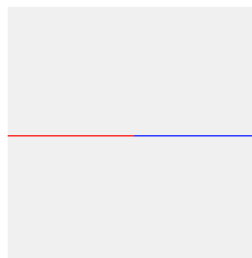
À partir d'une image carrée de taille 800 par 800 dont le fond est gris, tracer un segment de droite en pointillé. L'objectif est d'obtenir un résultat comme ci-dessous :

**Correction python**

```
im = Image . new ( "RGB" , (800 ,800), "grey" )
for k in range (0 ,800 ,2) :
    im . putpixel (( k ,400) ,(0 ,0 ,0) )
im . save ( "image.png", "png" )
```

**Exercice 8**

À partir d'une image carrée de taille 800 par 800 dont le fond est gris, tracer un segment de droite bicolore rouge/bleu. L'objectif est d'obtenir un résultat comme ci-dessous :



Retrouvez éduscol sur :



**Correction python****Méthode 1**

```

im = Image . new ( "RGB" , (800 ,800), "grey" )
for k in range (400) :
    im . putpixel (( k ,400) ,(255 ,0 ,0) )
for k in range (400 ,800) :
    im . putpixel (( k ,400) ,(0 ,0 ,255) )
im . save ( "image.png" ,"png" )

```

**Méthode 2**

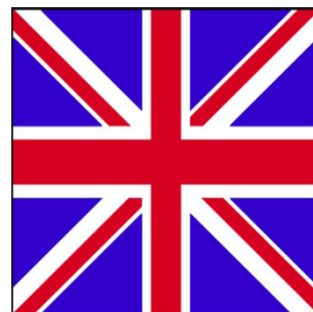
```

im = Image . new ( "RGB" , (800 ,800), "grey" )
for k in range (800) :
    if k<400 :
        im . putpixel (( k ,400) ,(255 ,0 ,0) )
    else :
        im . putpixel (( k ,400) ,(0 ,0 ,255) )
im . save ( "image.png" ,"png" )

```

**Exercice 9 : Drapeaux**

Dessiner les drapeaux suivants :

**Correction python drapeau français**

```

""" dimension et couleur : source wikipedia """
im = Image . new ( "RGB" , (900 ,450) , "grey" )
for l in range (450) :
    for c in range (900) :
        if c<300 :
            im . putpixel (( c , l) ,(5 ,20 ,64) )
        elif c<600 :
            im . putpixel (( c , l) ,(255 ,255 ,255) )
        else :
            im . putpixel (( c , l) ,(236 ,25 ,32) )
im . save ( "image.png" ,"png" )

```

Retrouvez éduscol sur :



### Commentaires

L'objectif de cet exercice est bien évidemment le drapeau français et la double boucle pour. Il est possible de préciser que le code couleur du bleu et du rouge ont été normalisés et se trouvent facilement sur le web.

Les dimensions du drapeau tchèque se trouvent sur le web également.  
Celles du drapeau britannique en revanche sont plus difficiles.

Retrouvez éduscol sur :

