

Déterminer les
étapes :
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Traduire l'algorithme : le
programme

Créer d'autres
algorithmes

Algorithme de traitement des images

Christophe Viroulaud

Seconde - SNT

Phot 03

Déterminer les
étapes :
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Traduire l'algorithme : le
programme

Créer d'autres
algorithmes



FIGURE 1 – Modifier le rendu d'une image

Déterminer les
étapes :
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Traduire l'algorithme : le
programme

Créer d'autres
algorithmes

Comment élaborer un programme informatique ?

1. Déterminer les étapes : l'algorithme
 - 1.1 Découper en étapes simples
 - 1.2 Détailler les étapes critiques
 - 1.3 Traduire l'algorithme : le programme
2. Créer d'autres algorithmes

Déterminer les
étapes :
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Traduire l'algorithme : le
programme

Créer d'autres
algorithmes

Découper en étapes simples

Déterminer les
étapes :
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Traduire l'algorithme : le
programme

Créer d'autres
algorithmes

À retenir

Pour faire exécuter une tâche à la machine, il faut lui détailler toutes les étapes à réaliser.

Déterminer les
étapes :
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Traduire l'algorithme : le
programme

Créer d'autres
algorithmes

Pour transformer l'image en noir et blanc, il faut :

- ▶ **Première étape** : Stocker l'image en mémoire.
- ▶ **Deuxième étape** : Modifier chaque pixel.
- ▶ **Troisième étape** : Enregistrer la nouvelle image.

1. Déterminer les étapes : l'algorithme
 - 1.1 Découper en étapes simples
 - 1.2 Détailler les étapes critiques
 - 1.3 Traduire l'algorithme : le programme
2. Créer d'autres algorithmes

Déterminer les
étapes :
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Traduire l'algorithme : le
programme

Créer d'autres
algorithmes

Détailler les étapes critiques

Une image est une grille composée de pixels. Pour transformer l'image couleur, en noir et blanc, il faut effectuer une opération sur chaque pixel.

	0	1	2	3	4
0					
1					
2					
3					

FIGURE 2 – Coordonnées d'un pixel

Déterminer les
étapes :
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Traduire l'algorithme : le
programme

Créer d'autres
algorithmes

Deuxième étape : Modifier chaque pixel.

	0	1	2	3	4
0					
1					
2					
3					

FIGURE 3 – Coordonnées d'un pixel

- Parcourir la grille ligne par ligne.
 - Parcourir la ligne colonne par colonne.
 - Récupérer les couleurs du pixel.
 - Le transformer en noir et blanc.

Déterminer les
étapes :
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Traduire l'algorithme : le
programme

Créer d'autres
algorithmes

1. Déterminer les étapes : l'algorithme
 - 1.1 Découper en étapes simples
 - 1.2 Détailler les étapes critiques
 - 1.3 Traduire l'algorithme : le programme
2. Créer d'autres algorithmes

Déterminer les
étapes :
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Traduire l'algorithme : le
programme

Créer d'autres
algorithmes

Traduire l'algorithme : le programme

Déterminer les
étapes :
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Traduire l'algorithme : le
programme

Créer d'autres
algorithmes

À retenir

Pour que l'ordinateur puisse exécuter l'algorithme, il faut le traduire dans un langage qu'il comprend.

Activité 1 :

1. Télécharger le dossier compressé *traitement-image.zip* sur le site <https://cviroulaud.github.io> .
2. Extraire le dossier traitement-image.
3. Ouvrir le logiciel *Spyder*.
4. Depuis le logiciel ouvrir le fichier **noir-blanc.py**.
5. Observer le programme et repérer les trois étapes de l'algorithme.

Déterminer les
étapes :
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Traduire l'algorithme : le
programme

Créer d'autres
algorithmes

Déterminer les
étapes :
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Traduire l'algorithme : le
programme

Créer d'autres
algorithmes

```
1 from PIL import Image
2 mon_image = Image.open("maisons-colorees.bmp")
3 colonne, ligne = mon_image.size
```

Code 1 – Stocker l'image

Déterminer les
étapes :
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Traduire l'algorithme : le
programme

Créer d'autres
algorithmes

```
1  for y in range(ligne):
2      for x in range(colonne):
3          pixel = mon_image.getpixel((x,y))
4
5          moyenne = (pixel[0] + pixel[1] + pixel
6              [2]) // 3
7          if moyenne < 128:
8              # le pixel sera noir
9              r = 0
10             v = 0
11             b = 0
12         else:
13             # le pixel sera blanc
14             r = 255
15             v = 255
16             b = 255
17
18     mon_image.putpixel((x,y), (r,v,b))
```

Déterminer les
étapes :
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Traduire l'algorithme : le
programme

Créer d'autres
algorithmes

```
1 mon_image.save("maisons-colorees-NB.bmp")  
2 mon_image.show()
```

Code 3 – Enregistrer l'image

Déterminer les
étapes :
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Traduire l'algorithme : le
programme

Créer d'autres
algorithmes

Activité 2 :

1. Que représente `pixel[0]` ?
2. Comment fait-on le choix de transformer le pixel en noir ou blanc ?
3. Exécuter le programme.
4. Modifier le programme pour que l'image obtenue soit plus sombre. Recommencer pour qu'elle soit plus claire.

Déterminer les
étapes :
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Traduire l'algorithme : le
programme

Créer d'autres
algorithmes

- ▶ `pixel[0]` niveau de rouge
- ▶ `pixel[1]` niveau de vert
- ▶ `pixel[2]` niveau de bleu

```
1 moyenne = (pixel[0] + pixel[1] + pixel[2]) // 3
2 # si la moyenne des couleurs RVB est inférieure à
  un seuil
3 if moyenne < 128:
4     # le pixel sera noir
5     r = 0
6     v = 0
7     b = 0
8 else:
9     # le pixel sera blanc
10    r = 255
11    v = 255
12    b = 255
```

Déterminer les
étapes :
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Traduire l'algorithme : le
programme

Créer d'autres
algorithmes

Code 4 – Choix de la couleur

Déterminer les
étapes :
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Traduire l'algorithme : le
programme

Créer d'autres
algorithmes

```
1 moyenne = (pixel[0] + pixel[1] + pixel[2]) // 3
2 # si la moyenne des couleurs RVB est inférieure à
  un seuil
3 if moyenne < 200:
4     # le pixel sera noir
5     r = 0
6     v = 0
7     b = 0
8 else:
9     # le pixel sera blanc
10    r = 255
11    v = 255
12    b = 255
```

Code 5 – Plus sombre : la comparaison est modifiée

Déterminer les
étapes :
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Traduire l'algorithme : le
programme

Créer d'autres
algorithmes

```
1 moyenne = (pixel[0] + pixel[1] + pixel[2]) // 3
2 # si la moyenne des couleurs RVB est inférieure à
  un seuil
3 if moyenne < 50:
4     # le pixel sera noir
5     r = 0
6     v = 0
7     b = 0
8 else:
9     # le pixel sera blanc
10    r = 255
11    v = 255
12    b = 255
```

Code 6 – Plus clair

1. Déterminer les étapes : l'algorithme
2. Créer d'autres algorithmes

Déterminer les
étapes :
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Traduire l'algorithme : le
programme

Créer d'autres
algorithmes

Créer d'autres algorithmes

Déterminer les
étapes :
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Traduire l'algorithme : le
programme

Créer d'autres
algorithmes

À partir de l'algorithme de base, nous pouvons imaginer d'autres traitements pour la photographie.

Activité 3 :

1. Dans les cours précédents, retrouver comment obtenir la couleur grise.
2. Enregistrer le programme précédent sous le nom **niveaux-gris.py**.
3. Modifier le programme pour que l'image obtenue soit en nuances de gris.
4. Enregistrer le programme précédent sous le nom **niveaux-rouge.py**.
5. Modifier le programme pour que l'image obtenue ne contienne que les composantes rouges de chaque pixel.

Déterminer les
étapes :
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Traduire l'algorithme : le
programme

Créer d'autres
algorithmes

Déterminer les
étapes :
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Traduire l'algorithme : le
programme

Créer d'autres
algorithmes

```
1 for y in range(ligne):
2     for x in range(colonne):
3         # récupérer le pixel
4         pixel = mon_image.getpixel((x,y))
5
6         moyenne = (pixel[0] + pixel[1] + pixel
7                    [2]) // 3
8
9         # on replace le nouveau pixel
10        mon_image.putpixel((x,y), (moyenne,
11                                   moyenne, moyenne))
```

Code 7 – Niveaux de gris

Déterminer les
étapes :
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Traduire l'algorithme : le
programme

Créer d'autres
algorithmes

```
1 for y in range(ligne):
2     for x in range(colonne):
3         # récupérer le pixel
4         pixel = mon_image.getpixel((x,y))
5
6         # on replace le nouveau pixel
7         mon_image.putpixel((x,y), (pixel[0], 0,
0))
```

Code 8 – Niveaux de rouge