

Algorithme de traitement des images

Christophe Viroulaud

Seconde - SNT

Phot 04



Construire
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc

Construire
l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc



Figure 1 – Modifier le rendu d'une image

Construire
l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter
l'algorithme

Environnement de travail
Étape 1
Étape 2
Étape 3

Créer d'autres
algorithmes

Filtre de couleur
Noir et blanc

Utiliser un nouveau langage informatique.

1. Construire l'algorithme

- 1.1 Découper en étapes simples
- 1.2 Détailler les étapes critiques

2. Implémenter l'algorithme

3. Créer d'autres algorithmes

Construire l'algorithme

- Découper en étapes simples
- Détailler les étapes critiques

Implémenter l'algorithme

- Environnement de travail
- Étape 1
- Étape 2
- Étape 3

Créer d'autres algorithmes

- Filtre de couleur
- Noir et blanc

Construire l'algorithme - Découper en étapes simples

Construire l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter l'algorithme

Environnement de travail
Étape 1
Étape 2
Étape 3

Créer d'autres algorithmes

Filtre de couleur
Noir et blanc

À retenir

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

Activité 1 : Quelles étapes semblent nécessaires pour transformer une image couleur en nuances de gris ?



Construire
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc

Première version peu détaillée :

Pour transformer l'image en nuances de gris, il faut :

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

Construire
l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc

1. Construire l'algorithme
 - 1.1 Découper en étapes simples
 - 1.2 Détailler les étapes critiques
2. Implémenter l'algorithme
3. Créer d'autres algorithmes

Construire
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc

Détailler les étapes critiques

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

	0	1	2	3	4
0					
1					
2					
3					

Figure 2 – Coordonnées d'un pixel

Construire
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc

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.
 - ▶ Calculer le gris correspondant.
 - ▶ Remplacer le pixel.

Construire
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc

- ▶ Parcourir la grille ligne par ligne :
 - ▶ Parcourir la ligne colonne par colonne :
 - ▶ Récupérer les couleurs du pixel.
 - ▶ Calculer le gris correspondant.
 - ▶ Remplacer le pixel.

Activité 2 : Quelle construction élémentaire de programmation reconnaît-on dans cet algorithme ?

Construire
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc

- ▶ Pour chaque ligne :
 - ▶ Pour chaque colonne :
 - ▶ Récupérer les couleurs du pixel.
 - ▶ Calculer le gris correspondant.
 - ▶ Remplacer le pixel.

Observation

On reconnaît deux boucles dans l'algorithme : la première pour aller voir chaque ligne, la seconde pour aller voir chaque colonne.

Construire
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filter de couleur

Noir et blanc

Construire
l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter
l'algorithme

Environnement de travail
Étape 1
Étape 2
Étape 3

Créer d'autres
algorithmes

Filter de couleur
Noir et blanc

Première étape : Placer l'image en mémoire.

Deuxième étape : Modifier chaque pixel.

- ▶ Pour chaque ligne :
 - ▶ Pour chaque colonne :
 - ▶ Récupérer les couleurs du pixel.
 - ▶ Calculer le gris correspondant.
 - ▶ Remplacer le pixel.

Troisième étape : Enregistrer la nouvelle image.

1. Construire l'algorithme

2. Implémenter l'algorithme

2.1 Environnement de travail

2.2 Étape 1

2.3 Étape 2

2.4 Étape 3

3. Créer d'autres algorithmes

Construire l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres algorithmes

Filtre de couleur

Noir et blanc

Implémenter l'algorithme - Environnement de travail

À retenir

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

Observation

Pour faciliter la tâche des développeurs, il existe des logiciels dédiés : un **Environnement de Développement Intégré**

Activité 3 : Ouvrir le logiciel **Thonny**, situé dans le répertoire **NSI**.

Construire
l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc

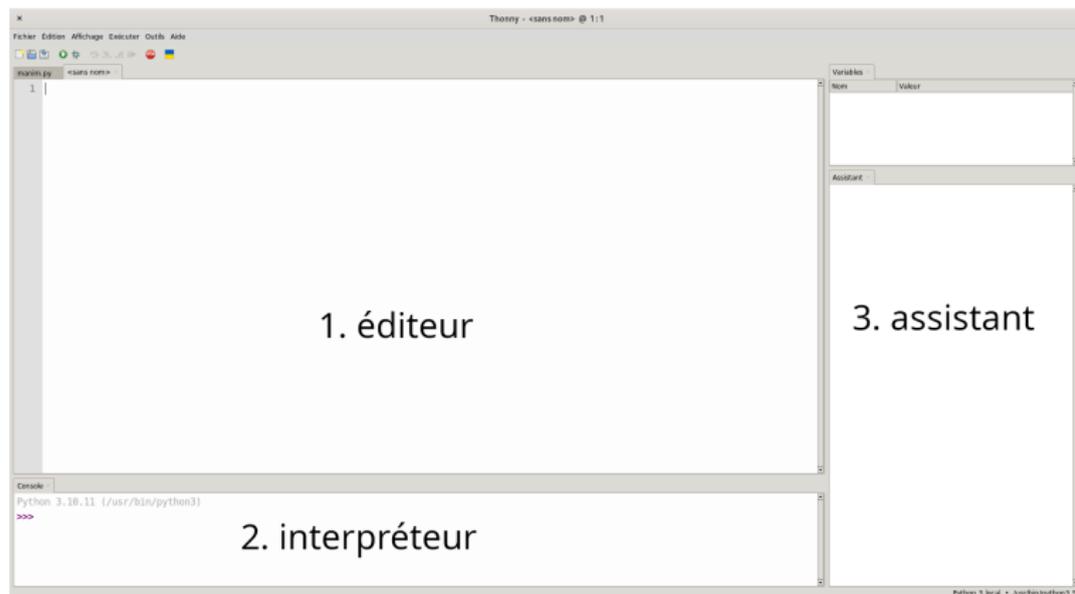


Figure 4 – L'Environnement de Développement Intégré permet d'écrire et d'exécuter des programmes Python.

Construire
l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1
Étape 2
Étape 3

Créer d'autres
algorithmes

Filtre de couleur
Noir et blanc

1. Construire l'algorithme

2. Implémenter l'algorithme

2.1 Environnement de travail

2.2 Étape 1

2.3 Étape 2

2.4 Étape 3

3. Créer d'autres algorithmes

Construire
l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc

Activité 4 :

1. Télécharger le dossier compressé *traitement-image.zip* sur le site <https://cviroulaud.gitlab.io> .
2. Extraire le dossier **traitement-image** dans le dossier **SNT/photographie**.
3. Depuis le logiciel **Thonny**, ouvrir le fichier **niveaux-gris.py**.
4. Observer le programme et repérer les trois étapes de l'algorithme.

Construire
l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc

À retenir

Les lignes qui commencent par # sont des commentaires du programmeur. Elles ne sont pas lues par l'interpréteur Python.

```
1 # Placer l'image en mémoire
```

Code 1 – Un commentaire pour aider à comprendre le code.

Construire
l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc

```

1 # Placer l'image en mémoire
2 from PIL import Image
3 mon_image = Image.open("maisons-colorees.bmp")
4 colonne, ligne = mon_image.size
    
```

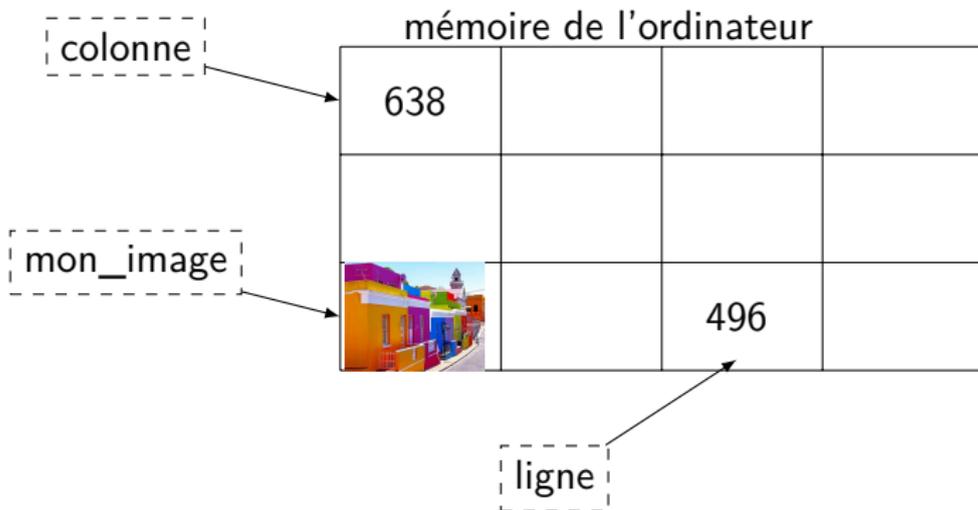


Figure 5 – Les variables sont des étiquettes placées sur des cellules de la mémoire.

1. Construire l'algorithme

2. Implémenter l'algorithme

2.1 Environnement de travail

2.2 Étape 1

2.3 Étape 2

2.4 Étape 3

3. Créer d'autres algorithmes

Construire
l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc

Étape 2

```
1 Pour chaque ligne:  
2     Pour chaque colonne:  
3         Récupérer les couleurs du pixel.  
4         Calculer le gris correspondant.  
5         Remplacer le pixel.
```

Code 2 – Étape 2 : Algorithme

```
1 for y in range(ligne):  
2     for x in range(colonne):  
3         pixel = mon_image.getpixel((x, y))  
4  
5         val = (pixel[0] + pixel[1] + pixel[2]) // 3  
6  
7         mon_image.putpixel((x, y), (val, val, val))
```

Code 3 – Étape 2 : Implémentation

Construire
l'algorithmeDécouper en étapes simples
Détailer les étapes critiquesImplémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc

Activité 5 :

1. Que signifie `pixel[0]` ?
2. Préciser le rôle de chaque ligne.

```

1 for y in range(ligne):
2     for x in range(colonne):
3         pixel = mon_image.getpixel((x, y))
4
5         val = (pixel[0] + pixel[1] + pixel[2]) // 3
6
7         mon_image.putpixel((x, y), (val, val, val))

```

Construire
l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc



Figure 6 – Quelle est la couleur de ce pixel ?

Un pixel est composé de trois couleurs :

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

Remarque

Pour obtenir du gris on utilise la même quantité de rouge, vert, et bleu.

Construire
l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc

```

1 # pour chaque ligne
2 for y in range(ligne):
3     # pour chaque colonne (de la ligne en cours)
4     for x in range(colonne):
5         # récupérer la couleur (RVB) du pixel en (x,y)
6         pixel = mon_image.getpixel((x, y))
7
8         # calculer la couleur moyenne de R, V, B
9         val = (pixel[0] + pixel[1] + pixel[2]) // 3
10
11        # modifier le pixel en (x, y)
12        # 3 valeurs identiques pour R V B
13        mon_image.putpixel((x, y), (val, val, val))

```

Construire
l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc

À retenir

L'instruction suivante est une boucle.

```
1 for i in range(10):  
2     print("numéro", i)  
3     print("encore")  
4 print("fin")
```

- ▶ La boucle effectue 10 répétitions.
- ▶ Les lignes 2 et 3 constituent le corps de la boucle. Elles sont **indentées**.
- ▶ La ligne 4 ne fait pas partie de boucle.

Construire
l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc

1. Construire l'algorithme

2. Implémenter l'algorithme

2.1 Environnement de travail

2.2 Étape 1

2.3 Étape 2

2.4 Étape 3

3. Créer d'autres algorithmes

Construire
l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc

Étape 3

Construire
l'algorithme

Découper en étapes simples

Détailler les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc

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

Code 4 – Enregistrer la nouvelle image

1. Construire l'algorithme

2. Implémenter l'algorithme

3. Créer d'autres algorithmes

3.1 Filtre de couleur

3.2 Noir et blanc

Construire
l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc

Construire
l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter
l'algorithme

Environnement de travail
Étape 1
Étape 2
Étape 3

Créer d'autres
algorithmes

Filtre de couleur
Noir et blanc

Activité 6 :

1. Déterminer un algorithme pour ne garder que la couleur rouge de chaque pixel d'une image.
2. Dans Thonny, créer un nouveau fichier `niveaux-rouge.py`.
3. Écrire le programme qui implémente l'algorithme.

Construire
l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc

Première étape : Placer l'image en mémoire.

Deuxième étape : Modifier chaque pixel.

- ▶ Pour chaque ligne :
 - ▶ Pour chaque colonne :
 - ▶ Récupérer les couleurs du pixel.
 - ▶ Garder seulement la composante rouge.
 - ▶ Remplacer le pixel.

Troisième étape : Enregistrer la nouvelle image.

```
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         # garder le rouge
7         val = pixel[0]
8
9         # on remplace le nouveau pixel
10        mon_image.putpixel((x,y), (val, 0, 0))
```

Code 5 – Niveaux de rouge

Construire
l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc



Figure 7 – filtre rouge

Construire
l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc

1. Construire l'algorithme

2. Implémenter l'algorithme

3. Créer d'autres algorithmes

3.1 Filtre de couleur

3.2 Noir et blanc

Construire
l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc

Observation

- ▶ Une image en gris est composée de 256 nuances.

Construire
l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc

Observation

- ▶ Une image en gris est composée de 256 nuances.
- ▶ Une image en rouge est composée de 256 nuances.

Construire
l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc

Observation

- ▶ Une image en gris est composée de 256 nuances.
- ▶ Une image en rouge est composée de 256 nuances.
- ▶ Plutôt que de n'avoir des nuances de gris, il est possible de ne garder que deux couleurs : noir et blanc.

Construire
l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filter de couleur

Noir et blanc

Activité 7 :

1. Ouvrir le fichier `noir-blanc.py`
2. Observer le code et déterminer le choix effectué pour transformer le pixel coloré en noir ou blanc.
3. Modifier le code pour obtenir une image plus sombre.
4. Modifier le code pour obtenir une image plus claire.

Construire
l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc

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

instruire
lgorithme
couper en étapes simples
tailler les étapes critiquesplémenter
lgorithme

vironnement de travail

ape 1

ape 2

ape 3

éer d'autres
lgorithmes

tre de couleur

oir et blanc

Code 6 – La moyenne des trois couleurs détermine le choix du noir ou blanc.

```

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

```

instruire
lgorithme

couper en étapes simples
tailler les étapes critiques

plémenter
lgorithme

vironnement de travail

ape 1

ape 2

ape 3

éer d'autres
lgorithmes

tre de couleur

ir et blanc

Code 7 – Plus sombre : la comparaison est modifiée

```

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 8 – Plus clair

instruire
lgorithme

couper en étapes simples
tailler les étapes critiques

planifier
lgorithme

environnement de travail

ape 1

ape 2

ape 3

écouter d'autres
algorithmes

traitement de couleur

noir et blanc

À retenir

L'instruction suivante est une structure conditionnelle.

```
1 x = 3
2 if x < 10:
3     print("moins de 10")
4 else:
5     print("plus de 10")
```

- ▶ La ligne 2 compare la valeur de x et 10.
- ▶ La ligne 3 est exécutée si la condition est vraie.
- ▶ La ligne 5 est exécutée si la condition est fausse.

Construire
l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc

À retenir

On retrouve dans le langage Python, les constructions élémentaires pour écrire un programme :

- ▶ les variables,
- ▶ les structures conditionnelles,
- ▶ les boucles.

Construire
l'algorithme

Découper en étapes simples
Détailer les étapes critiques

Implémenter
l'algorithme

Environnement de travail

Étape 1

Étape 2

Étape 3

Créer d'autres
algorithmes

Filtre de couleur

Noir et blanc