

Les images du télescope James Webb



===== Principe des filtres de couleurs =====

Il est possible de télécharger les données des images capturées par le télescope James Webb sous différentes longueurs d'ondes : <https://mast.stsci.edu/portal/Mashup/Clients/Mast/Portal.html>

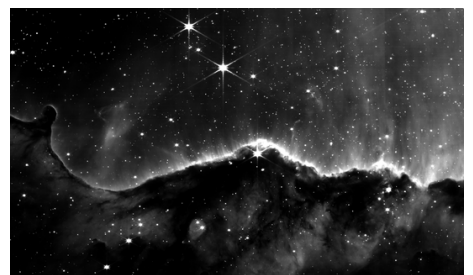
Voici, après une première retouche, les images issues de trois filtres différents :



F444 : Rouge



F335 : Vert



F187 : Bleu

===== Vers la couleur =====

```

1 import matplotlib.pyplot as plt
2 from PIL import Image
3 import numpy as np
4
5 #Import des images filtres:
6 R_img = Image.open("Piliers_RED_small_i.png")
7 G_img = Image.open("Piliers_GREEN_small_i.png")
8 B_img = Image.open("Piliers_BLUE_small_i.png")
9
10 #Création de la nouvelle image couleur vide:
11 RGB_img = Image.new("RGB", R_img.size)
12
13 #Numpy 2D array :
14 R_data = np.array(R_img.getdata())
15 G_data = np.array(G_img.getdata())
16 B_data = np.array(B_img.getdata())
17
18 #Formatage 0-1
19 R_data = R_data/max(R_data)
20 G_data = G_data/max(G_data)
21 B_data = B_data/max(B_data)
    
```

```

1 #Fonctions filtres :
2 def resserrer(x):
3     return 4*(x-0.5)**3+0.5
4 def racine(x):
5     return np.sqrt(x)
6 def carre(x):
7     return x*x
8 def contraste(x):
9     return x*(-2*x**2+3*x)
10 def renverser(x):
11     return 1-x
12
13 #Modifications :
14 B_data = contraste(racine(B_data))
15 #Remplissage des images filtres modifi es sur 0-255:
16 R_img.putdata(255*R_data)
17 G_img.putdata(255*G_data)
18 B_img.putdata(255*B_data)
19
20 #Remplissage de l'image couleur: On parcourt tous les pixels de chaque image :
21 for x in range(RGB_img.size[0]):
22     for y in range(RGB_img.size[1]):
23         #On recup re les valeurs de pixels de chaque image filtre:
24         R = R_img.getpixel((x, y))
25         G = G_img.getpixel((x, y))
26         B = B_img.getpixel((x, y))
27         if R > 4*B:
28             B = R
29         #pix = (R, G, B)
30         pix = ((6*R+2*G+B)//9, (R+3*G+B)//5, (G+2*B+G)//4)
31         #On ajoute le pixel dans la nouvelle image
32         RGB_img.putpixel((x, y), pix)
33 RGB_img.show()

```



Un exemple de production