

День 3. Операції з растровою графікою

Робота з кольірними каналами зображення

- 1) ^{1 бал} Модуль `matplotlib.image` надає функції для завантаження, зміни масштабу та відображення зображення. Зчитування зображення по шляху (тут – `mtplogo.png`) до нього відбувається за допомогою функції `imread()`:

```
import matplotlib.image as mpimg  
img = mpimg.imread('mtplogo.png')
```

Утворений об'єкт є `ndarray`-масивом із значеннями типу `float32 dtype`. Виведіть тип цього об'єкта на екран та додайте відповідний код і скріншот до звіту.

Бібліотека `matplotlib` підтримує завантаження тільки `png`-файлів, тому поширеним інструментом роботи з іншими форматами зображень є бібліотека `imageio`.

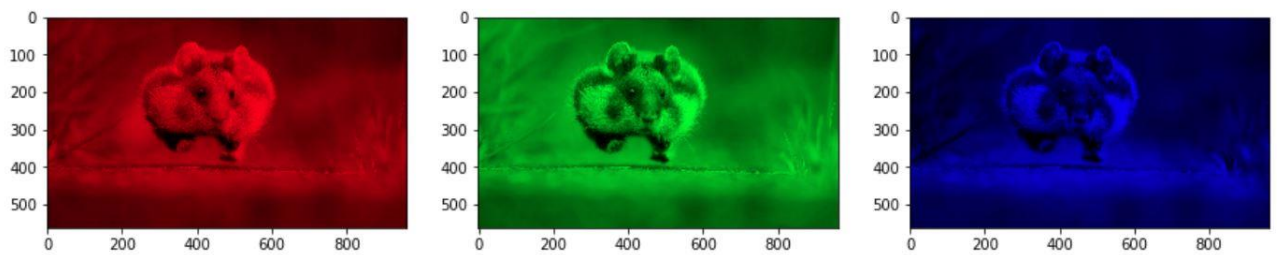
```
import imageio  
import matplotlib.pyplot as plt  
  
pic = imageio.imread('demo_2.jpg')  
plt.imshow(pic)  
plt.show()
```



Показ зображення відбувається за допомогою функції `imshow()`, куди передається відповідний об'єкт-зображення, а також вже відомої функції `show()`.

Оберіть доречне растрове зображення та виведіть у консоль його базові властивості: тип, лінійні розміри (за допомогою `shape`), мінімальне та максимальне `RGB`-значення. Також зверніться до довільного пікселя зображення та окремо виведіть значення його каналів `RGB`.

Продовжуючи роботу з `RGB`-простором, розбийте растрове зображення на три окремі складові, як це показано на рисунку



- 2) ^{1 бал} Доповніть результат задачі 1 чорно-білим (grayscale) зображенням, застосувавши формулу для перетворення кольорового пікселя у піксель в сірих тонах:

$$Y' = 0.299 R + 0.587 G + 0.114 B$$

Для вище згаданого прикладу отримаємо:



Розгляньте також [методи знебарвлення зображення](#) для графічного редактору GIMP. Застосуйте три представлені формули разом з вище згаданим стандартним перетворенням. Порівняйте результати на одному графіку. Додайте відповідний код та скріншоти до звіту.