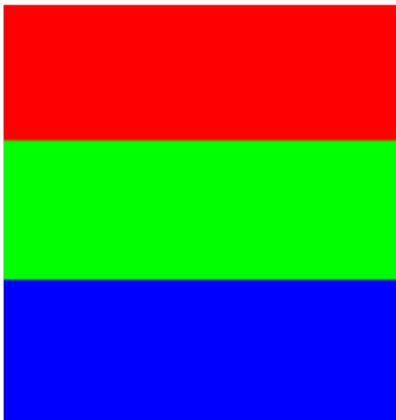


کار بر روی فضاهای رنگی:

همانطور که گفتیم در یک عکس height بیانگر تعداد سطرها و width بیانگر تعداد ستون ها است. اگر به تصویر زیر دقت کنید:



احتمالا می توانید حدس بزنید که این تصویر ماتریسی است که بخش بالایی آن کاملا قرمز است $[[0,0,255]]$ ، بخش میانی آن کاملا سبز است $[[0,255,0]]$ و در نهایت آخرین بخش آن کاملا آبی است $[[255,0,0]]$. در این تصویر هیچ ترکیب رنگی اتفاق نیفتاده است و هر کانال کاملا به صورت مجزا نمایش داده می شود. با استفاده از دستور زیر می توان هر کانال را کاملا از هم جدا نمود:

```
b,g,r = cv2.split(img)
b
```

با انجام این دستور می توانیم خروجی را به شکل زیر برای هر کدام از کانال ها به صورت مجزا داشته باشیم:

```
array([[ 0,  0,  0, ...,  0,  0,  0],
       [ 0,  0,  0, ...,  0,  0,  0],
       [ 0,  0,  0, ...,  0,  0,  0],
       ...,
       [255, 255, 255, ..., 255, 255, 255],
       [255, 255, 255, ..., 255, 255, 255],
       [255, 255, 255, ..., 255, 255, 255]], dtype=uint8)
```

در واقع می دانیم که عکس ما به صورت تعدادی سطر و ستون و همچنین 3 کانال رنگی بوده که با این روش توانسته ایم هر کدام از آنها را به صورت مجزا به دست بیاوریم.

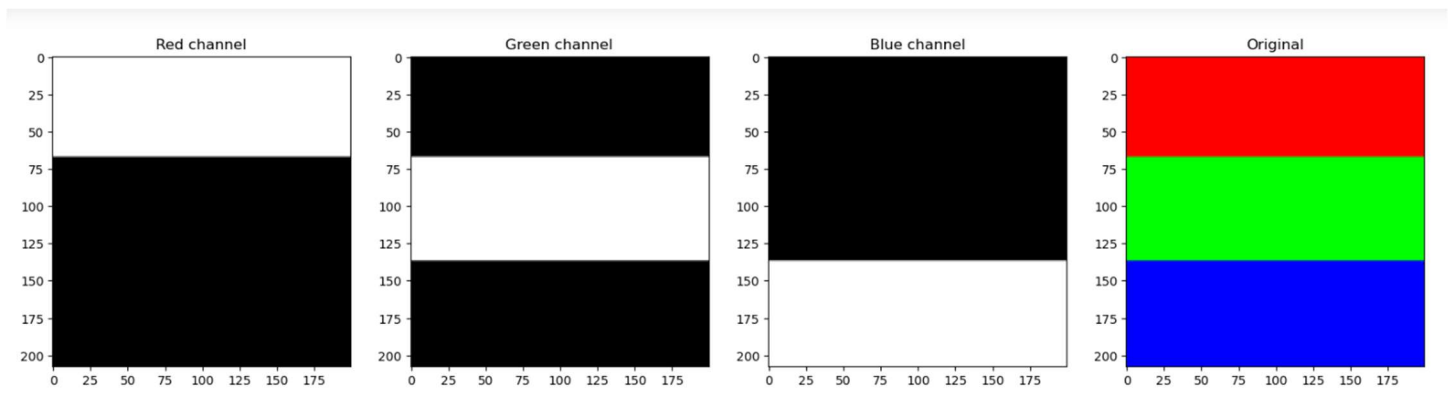
اکنون می توانیم با استفاده از کتابخانه matplotlib چند عکس را در خروجی درون یک سطر نمایش دهیم. برای این کار می توانیم از روش شی گرای در این کتابخانه استفاده کنیم.

```
# show image with matplotlib
plt.figure(figsize=[20,5]) # width * height
plt.subplot(141);plt.imshow(r, cmap= 'gray');plt.title('Red channel')
plt.subplot(142);plt.imshow(g,cmap= 'gray');plt.title('Green channel')
```

```
plt.subplot(143);plt.imshow(b,cmap= 'gray');plt.title('Blue channel')

# show the original image
plt.subplot(144);plt.imshow(img[...,:-1]);plt.title('Original')
```

همانطور که پیشتر بیان در کانال های سطح خاکستری صفر به معنی عدم وجود پیکسل و رنگ سیاه است و 255 به معنی وجود یک پیکسل و رنگ سفید است. اگر به خروجی کد بالا در نوت بوک خود دقت کنید ما هر کانال را با سطح خاکستری نمایش داده ایم. و چون می دانیم در عکس فوق هر کانال رنگی دارای شدت 255 می باشد بنابراین انتظار خواهیم داشت که این کانال ها در سطح خاکستری به رنگ سفید مطلق نمایش داده شوند.



حال که با نحوه ترکیب این کانال ها آشنا شدیم می توانیم همین کار را با سایر تصاویر نیز انجام دهیم. پیشنهاد می شود برای درک بهتر این موضوع عملیات جداسازی تصاویر را بر روی عکس های واقعی انجام داده و نتایج را با هم مقایسه نمایید.

ادغام کانال های رنگی:

بعد از جداسازی کانال های رنگی ما می توانیم همان تصاویر را دوباره و این بار با استفاده از دستور `merge` به شکل زیر با هم ادغام نماییم.

```
cv2.merge([b,g,r])
```

نکته مهمی که در زمان ادغام کانال ها باید به آن توجه کرد این است که بعد تمامی کانل ها باید دقیقا شبیه هم باشند و ما نمی توانیم کانال هایی با ابعاد متفاوت را با هم ادغام نماییم.

همچنین ما می توانیم در زمان ادغام سایر کانال ها را حذف کرده و عکس رنگی را تنها با یک کانال رنگی نمایش دهیم. یکی از راه های انجام این عملیات استفاده از ماتریس `zeros_like` می باشد. همچنین در هنگام انجام این عملیات ذکر این نکته

insta: kahkeshani_mohammad

دوره پردازش تصویر و بینایی کامپیوتر با open cv

youtube: <https://www.youtube.com/@mohammadkahkeshani>

• مدرس محمد کهکشانی (مدرس رسمی دانشگاه هاروارد)

بسیار ضروری است که هنگامی که ما می خواهیم یک ماتریس را به عنوان عکس ایجاد نماییم نوع و جنس آن حتما باید از نوع uint8 باشد.