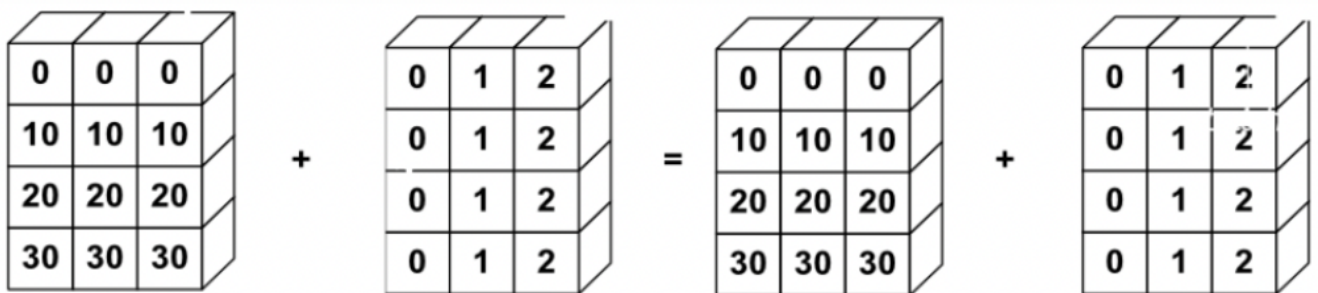


Colab Link: <https://colab.research.google.com/drive/1tdrSpxnAYtORZ1i8AJsqhzX24KceNpJY?usp=sharing>

▼ Broadcasting

```
import numpy as np
np.arange(12).reshape(3, 4) + 2
```

```
↳ array([[ 2,  3,  4,  5],
        [ 6,  7,  8,  9],
        [10, 11, 12, 13]])
```



```
b = np.tile(np.arange(0, 3), (4, 1))
```

```
np.tile(np.arange(0, 3), (4, 2))
```

```
array([[0, 1, 2, 0, 1, 2],
       [0, 1, 2, 0, 1, 2],
       [0, 1, 2, 0, 1, 2],
       [0, 1, 2, 0, 1, 2]])
```

```
a = np.tile(np.arange(0, 40, 10), (3, 1)).T
```

a

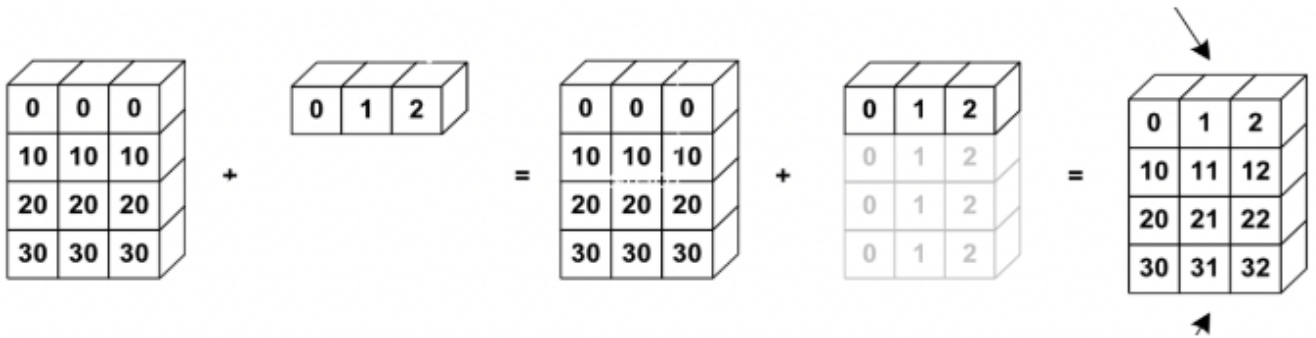
```
array([[ 0,  0,  0],
       [10, 10, 10],
       [20, 20, 20],
       [30, 30, 30]])
```

b

```
array([[0, 1, 2],
       [0, 1, 2],
       [0, 1, 2],
       [0, 1, 2]])
```

a + b

```
array([[ 0,  1,  2],
       [10, 11, 12],
       [20, 21, 22],
       [30, 31, 32]])
```



a

```
array([[ 0,  0,  0],
       [10, 10, 10],
       [20, 20, 20],
       [30, 30, 30]])
```

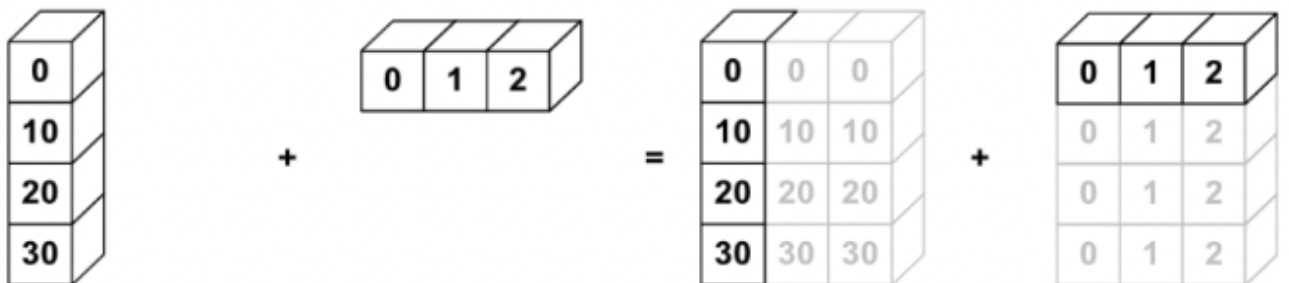
b = np.arange(0, 3)

b

```
array([0, 1, 2])
```

a + b # broadcasting

```
array([[ 0,  1,  2],
       [10, 11, 12],
       [20, 21, 22],
       [30, 31, 32]])
```



a = np.arange(0, 40, 10).reshape(4, 1)

```
b = np.arange(0, 3)
```

```
a + b
```

```
array([[ 0,  1,  2],
       [10, 11, 12],
       [20, 21, 22],
       [30, 31, 32]])
```

```
a = np.arange(8).reshape(2,4)
b = np.arange(16).reshape(4,4)
print(a*b)
```

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-29-74bfc3ae5d8e> in <module>()
      1 a = np.arange(8).reshape(2,4)
      2 b = np.arange(16).reshape(4,4)
----> 3 print(a*b)
```

ValueError: operands could not be broadcast together with shapes (2,4) (4,4)

SEARCH STACK OVERFLOW

```
A = np.arange(1,10).reshape(3,3)
B = np.array([-1, 0, 1]) # (3, ) --> (1, 3)
A * B
```

```
array([[ -1,  0,  3],
       [ -4,  0,  6],
       [ -7,  0,  9]])
```

```
A = np.arange(1,10).reshape(3,3)
B = np.arange(3, 10, 3).reshape(3,1)
C = A + B
C
```

```
array([[ 4,  5,  6],
       [10, 11, 12],
       [16, 17, 18]])
```

```
A = np.arange(12).reshape(3, 4) # (3, 4)
B = np.array([1, 2, 3]) # (3,) --> (1, 3)
A + B
```

```
ValueError                                Traceback (most recent call last)
```

```
<ipython-input-33-244aa7798641> in <module>()
```

```
1 A = np.arange(12).reshape(3, 4) # (3, 4)
```

```
# array splitting and merging
```

```
x = np.arange(1, 10)
```

```
x
```

```
array([1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
np.split(x, 3)
```

```
[array([1, 2, 3]), array([4, 5, 6]), array([7, 8, 9])]
```

```
np.split(x, [3, 5, 6])
```

```
[array([1, 2, 3]), array([4, 5]), array([6]), array([7, 8, 9])]
```

```
x = np.arange(16).reshape(4, 4) #hsplit, vsplit
```

```
x
```

```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11],
       [12, 13, 14, 15]])
```

```
np.hsplit(x, 2)
```

```
[array([[ 0,  1],
       [ 4,  5],
       [ 8,  9],
       [12, 13]]), array([[ 2,  3],
       [ 6,  7],
       [10, 11],
       [14, 15]])]
```

```
np.hsplit(x, [2, 3, 4])
```

```
[array([[ 0,  1],
       [ 4,  5],
       [ 8,  9],
       [12, 13]]), array([[ 2],
       [ 6],
       [10],
       [14]]), array([[ 3],
       [ 7],
       [11],
       [15]]), array([], shape=(4, 0), dtype=int64)]
```

```
np.vsplit(x, 2)
```

```
[array([[0, 1, 2, 3],
        [4, 5, 6, 7]]), array([[ 8,  9, 10, 11],
        [12, 13, 14, 15]])]
```

```
# stack
```

```
x = np.arange(5)
```

```
x
```

```
array([0, 1, 2, 3, 4])
```

```
np.vstack([x, x, x])
```

```
array([[0, 1, 2, 3, 4],
        [0, 1, 2, 3, 4],
        [0, 1, 2, 3, 4]])
```

```
y = np.arange(5).reshape(5, 1)
```

```
y
```

```
array([[0],
        [1],
        [2],
        [3],
        [4]])
```

```
np.hstack([y, y, y])
```

```
array([[0, 0, 0],
        [1, 1, 1],
        [2, 2, 2],
        [3, 3, 3],
        [4, 4, 4]])
```

```
y = np.arange(3)
```

```
y
```

```
array([0, 1, 2])
```

```
np.hstack([y, y, y])
```

```
array([0, 1, 2, 0, 1, 2, 0, 1, 2])
```

```
a = np.array([[1], [2], [3]])
```

```
b = np.array([[4], [5], [6]])
```

```
np.hstack((a, b))
```

```
# np.concatenate

z = np.array([[2, 4]])
z

array([[2, 4]])

np.concatenate([z, z], axis=0)

array([[2, 4],
       [2, 4]])

np.concatenate([z, z], axis=1)

array([[2, 4, 2, 4]])

a = np.array([[1, 2], [3, 4]])
a

array([[1, 2],
       [3, 4]])

b = np.array([[5, 6]])
b

array([[5, 6]])

np.concatenate([a, b], axis=0)

array([[1, 2],
       [3, 4],
       [5, 6]])

# View and Copies (Shallow VS Deep Copy)

a = np.arange(4)
a

array([0, 1, 2, 3])

b = a.reshape(2, 2)
b

array([[0, 1],
       [2, 3]])

a[0] = 100
```

a

```
array([100,  1,  2,  3])
```

b

```
array([[100,  1],
       [ 2,  3]])
```

```
a = np.arange(4)
```

a

```
array([0, 1, 2, 3])
```

```
c = a + 2
```

c

```
array([2, 3, 4, 5])
```

```
a[0] = 100
```

a

```
array([100,  1,  2,  3])
```

c

```
array([2, 3, 4, 5])
```

```
np.shares_memory(a, c)
```

```
False
```

```
np.shares_memory(a, b)
```

```
True
```

```
d = a + 0
```

d

```
array([0, 1, 2, 3])
```

```
np.shares_memory(a, d)
```

```
False
```

```
a = np.arange(4)
```

```
b = a.reshape(2, 2).copy()
```

```
np.shares_memory(a, b)
```

```
False
```

```
a = np.arange(6)
print(a)
```

```
[0 1 2 3 4 5]
```

```
b = a[a%1 == 0]
print(b)
```

```
[0 1 2 3 4 5]
```

```
b[0] = 10
print(b)
```

```
[10  1  2  3  4  5]
```

```
a
```

```
array([0, 1, 2, 3, 4, 5])
```

```
np.shares_memory(a, b)
```

```
False
```

```
# 3-Dimensional Arrays
```

```
np.arange(24).reshape(2, 3, 4)
```

```
array([[[ 0,  1,  2,  3],
        [ 4,  5,  6,  7],
        [ 8,  9, 10, 11]],
       [[12, 13, 14, 15],
        [16, 17, 18, 19],
        [20, 21, 22, 23]]])
```


Handwritten pink annotations on the screenshot:

- $P(R,G,B)$ with arrows pointing to the dimensions of the array.
- $2D$ with an arrow pointing to a box labeled *Image*.
- 3×4 next to the array dimensions.
- Diagrams of 2D and 3D grids with axes labeled R, G, B and dimensions $0, 1, 2, 3, 4$.

```
import numpy as np
import matplotlib.pyplot as plt
# imread - image read
# imshow - shows the image
```

```
!gdown 17tYTDPU5hpby9t0kGd7w_-zBsby7sEd
```

Downloading...

From: https://drive.google.com/uc?id=17tYTDPU5hpby9t0kGd7w_-zBsby7sEd

To: /content/fruits.png

100% 4.71M/4.71M [00:00<00:00, 218MB/s]

```
!gdown 1o-8yqdTM7cfz_mAaNCi2nH0urFu7pcqI
```

Downloading...

From: https://drive.google.com/uc?id=1o-8yqdTM7cfz_mAaNCi2nH0urFu7pcqI

To: /content/emma_stone.jpeg

100% 80.3k/80.3k [00:00<00:00, 52.5MB/s]

```
img = plt.imread("fruits.png")
plt.imshow(img)
```

```
<matplotlib.image.AxesImage at 0x7ffa28775490>
```



```
img.ndim
```

```
3
```

```
img.shape
```

```
(1333, 2000, 3)
```

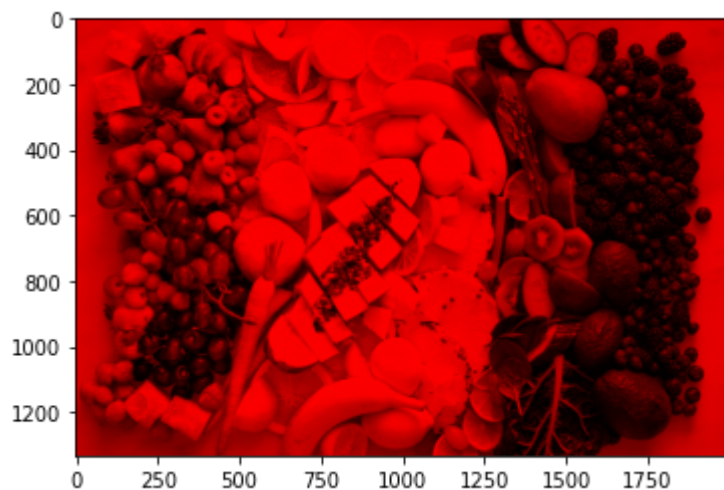
```
img = plt.imread('fruits.png')
```

```
img.shape
```

```
(1333, 2000, 3)
```

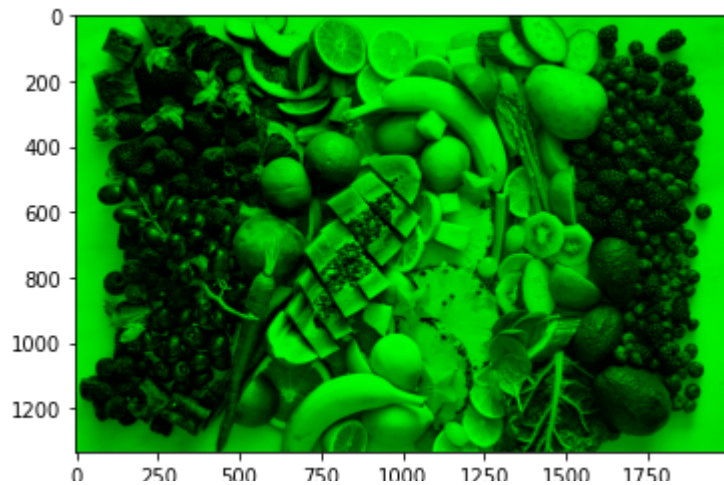
```
img_R = img.copy()  
img_R[:, :, (1, 2)] = 0  
plt.imshow(img_R)
```

```
<matplotlib.image.AxesImage at 0x7ffa273a02d0>
```



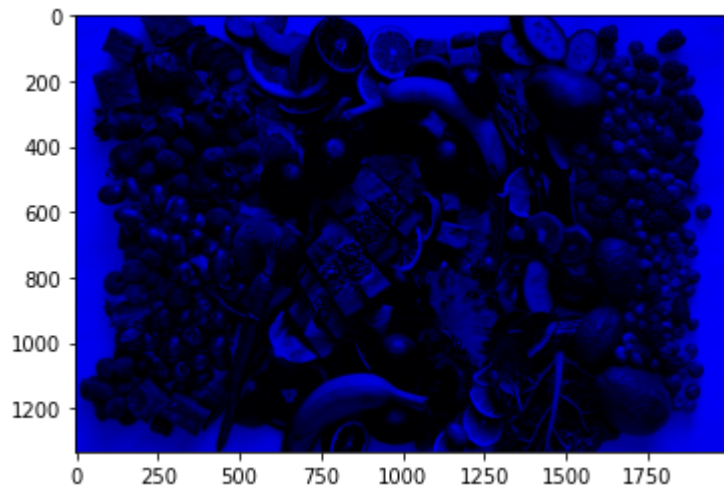
```
img_G = img.copy()  
img_G[:, :, (0, 2)] = 0  
plt.imshow(img_G)
```

<matplotlib.image.AxesImage at 0x7ffa27306fd0>



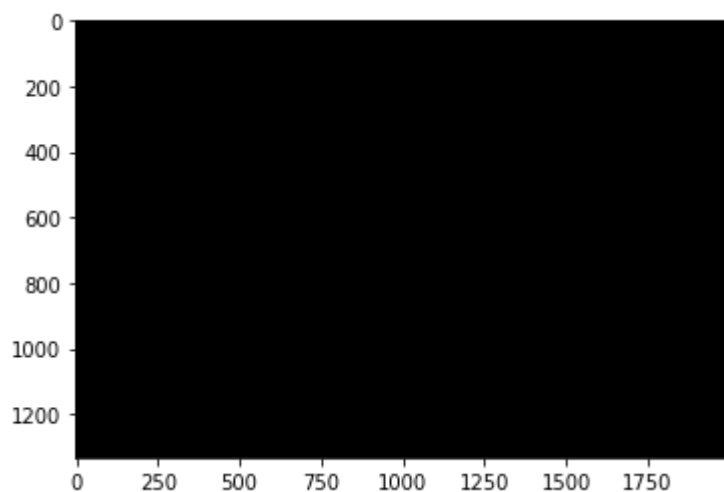
```
img_B = img.copy()  
img_B[:, :, (0, 1)] = 0  
plt.imshow(img_B)
```

<matplotlib.image.AxesImage at 0x7ffa273a8990>



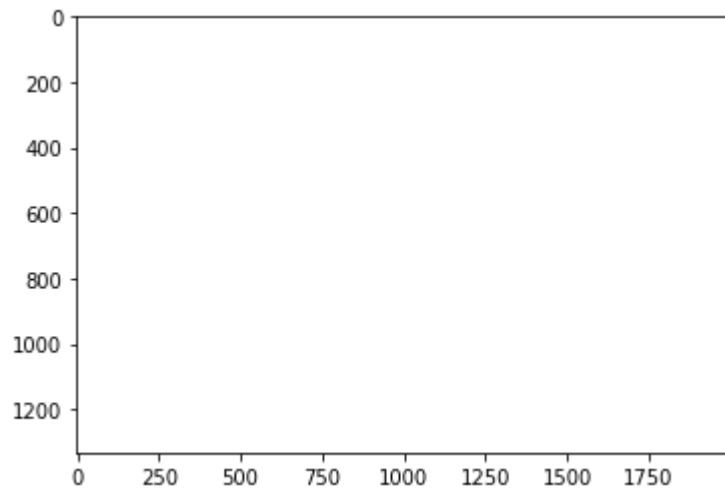
```
img_b = img.copy()  
img_b[:, :, :] = 0  
plt.imshow(img_b)
```

<matplotlib.image.AxesImage at 0x7ffa272587d0>



```
img_b = img.copy()  
img_b[:, :, :] = 255  
plt.imshow(img_b)
```

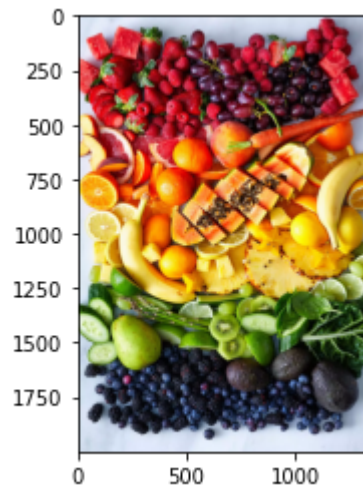
Clipping input data to the valid range for imshow with RGB data ([0..1] for fl
<matplotlib.image.AxesImage at 0x7ffa271cc510>



```
# rotate an image
```

```
plt.imshow(np.transpose(img, (1, 0, 2)))
```

<matplotlib.image.AxesImage at 0x7ffa27146e50>



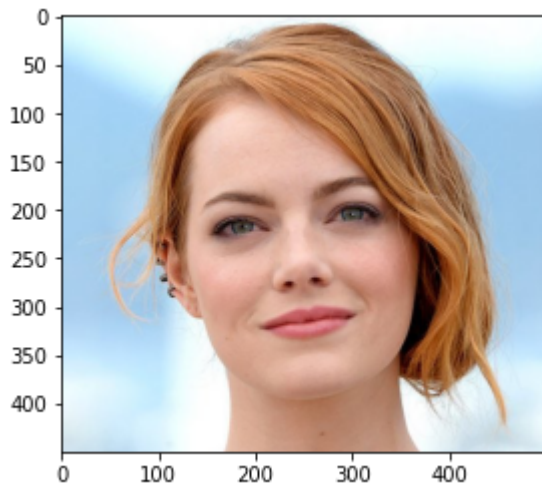
```
img = plt.imread("emma_stone.jpeg")  
plt.imshow(img)
```

```
<matplotlib.image.AxesImage at 0x7ffa26fcd910>
```



```
plt.imshow(img[:450, 200:700, :])
```

```
<matplotlib.image.AxesImage at 0x7ffa26f46710>
```



```
y = np.arange(5).reshape(5, 1)
y
```

```
array([[0],
       [1],
       [2],
       [3],
       [4]])
```

```
np.concatenate([y, y, y], axis=1).shape
```

```
(5, 3)
```

```
np.stack([y, y, y], axis=1).shape
```

```
(5, 3, 1)
```

```
a=np.arange(4)
```

```
b=a
```

✓ 0s completed at 23:27

×