

In [1]:

```
1 import numpy as np
```

In [2]:

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

In [3]:

```
1 A
```

Out[3]:

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

In [4]:

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

In [5]:

```
1 v
```

Out[5]:

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

In [6]:

```
1 v.ndim
```

Out[6]:

```
1
```

In [7]:

```
1 v.shape
```

Out[7]:

```
(3,)
```

In [8]:

```
1 v * A
```

ValueError

Traceback (most recent call

last)

<ipython-input-8-2a1ae6ae7fd0> in <module>

----> 1 v * A

ValueError: operands could not be broadcast together with shapes (3,) (3,4)

In [9]:

```
1 A * v
```

```
-----  
-----  
ValueError                                Traceback (most recent call  
last)  
<ipython-input-9-42a27ee68a2a> in <module>  
----> 1 A * v  
  
ValueError: operands could not be broadcast together with shapes (3,4)  
(3,)
```

In [16]:

```
1 v2 = np.array([1, 2, 3, 4]).reshape(4, 1)  
2 A = np.arange(12).reshape(4, 3)  
3 print(v2.shape)  
4 print(A.shape)
```

```
(4, 1)
```

```
(4, 3)
```

In [19]:

```
1 A
```

Out[19]:

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

In [18]:

```
1 A * v2
```

Out[18]:

```
array([[ 0,  1,  2],  
       [ 6,  8, 10],  
       [18, 21, 24],  
       [36, 40, 44]])
```

In [20]:

```
1 v2 * A
```

Out[20]:

```
array([[ 0,  1,  2],  
       [ 6,  8, 10],  
       [18, 21, 24],  
       [36, 40, 44]])
```

In [22]:

```
1 A = np.arange(9).reshape(3, 3)
2 A
```

Out[22]:

```
array([[0, 1, 2],
       [3, 4, 5],
       [6, 7, 8]])
```

In [25]:

```
1 B = np.arange(3).reshape(3, 1)
2 B
```

Out[25]:

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

In [24]:

```
1 A * B
```

Out[24]:

```
array([[ 0,  1,  4],
       [ 0,  4, 10],
       [ 0,  7, 16]])
```

In [26]:

```
1 A * B
```

Out[26]:

```
array([[ 0,  0,  0],
       [ 3,  4,  5],
       [12, 14, 16]])
```

In [28]:

```
1 A = np.arange(9).reshape(3,3)
2 B = np.array([-1, 0, 1])
3 A * B
```

Out[28]:

```
array([[ 0,  0,  2],
       [-3,  0,  5],
       [-6,  0,  8]])
```

In [29]:

```
1 import random
```

In [30]:

```
1 random.randint(1, 100)
```

Out[30]:

63

In [31]:

```
1 np.random.randint(1, 100)
```

Out[31]:

41

In [32]:

```
1 np.random.randint(1, 100, 5) # integer
```

Out[32]:

```
array([69, 63, 11, 30, 37])
```

In [33]:

```
1 np.random.rand() # [0, 1]
```

Out[33]:

0.03801727439132363

In [34]:

```
1 # [0, 25], 5 floating point numbers
```

In [37]:

```
1 np.random.rand(5)*25
```

Out[37]:

```
array([12.61429    ,  9.19884903,  4.71438008, 20.31109788, 22.4263305
 9])
```

In [38]:

```
1 # 1 floating number - [50, 75]
```

In [40]:

```
1 50 + np.random.rand()*25
```

Out[40]:

73.36790840512512

In [48]:

```
1 mu = 100
2 std = 15
3 s = np.random.normal(mu, std, 1000000)
```

In [49]:

```
1 s.mean()
```

Out[49]:

```
99.9969743017827
```

In [50]:

```
1 s.std()
```

Out[50]:

```
14.998104472649917
```

In [51]:

```
1 # shallow and deep copy in Numpy
```

In [52]:

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

Out[52]:

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

In [69]:

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

Out[69]:

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

In [68]:

```
1 a
```

Out[68]:

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

In [70]:

```
1 b.flags
```

Out[70]:

```
C_CONTIGUOUS : True
F_CONTIGUOUS : False
OWNDATA : False
WRITEABLE : True
ALIGNED : True
WRITEBACKIFCOPY : False
UPDATEIFCOPY : False
```

In [58]:

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

In [59]:

```
1 a
```

Out[59]:

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

In [60]:

```
1 b
```

Out[60]:

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

In [61]:

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

In [62]:

```
1 a
```

Out[62]:

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

In [63]:

```
1 c = a + 2
```

In [64]:

```
1 c
```

Out[64]:

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

In [66]:

```
1 c
```

Out[66]:

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

In [67]:

```
1 c.flags
```

Out[67]:

```
C_CONTIGUOUS : True
F_CONTIGUOUS : True
OWNDATA : True
WRITEABLE : True
ALIGNED : True
WRITEBACKIFCOPY : False
UPDATEIFCOPY : False
```

In [71]:

```
1 a = np.arange(12).reshape(3, 4)
2 a
```

Out[71]:

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

In [72]:

```
1 a.mean()
```

Out[72]:

5.5

In [73]:

```
1 np.mean(a, axis=0)
```

Out[73]:

```
array([4., 5., 6., 7.])
```

In [74]:

```
1 np.sum(a, axis=0)
```

Out[74]:

```
array([12, 15, 18, 21])
```

In [75]:

```
1 #np.median()
```

In [82]:

```
1 a = np.array([0, 2, 0, 4])
```

In [86]:

```
1 a != 0
```

Out[86]:

```
array([False,  True, False,  True])
```

In [84]:

```
1 def any_non_zero(a):  
2     for element in a:  
3         if element != 0:  
4             return True
```

In [85]:

```
1 any_non_zero(a)
```

Out[85]:

```
True
```

In [87]:

```
1 np.any(a != 0)
```

Out[87]:

```
True
```

In [88]:

```
1 np.any(a)
```

Out[88]:

```
True
```

In [89]:

```
1 np.all(a)
```

Out[89]:

```
False
```

In [90]:

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

In [91]:

```
1 np.all(a)
```

Out[91]:

```
True
```


In [93]:

```
1 np.all(a <= 3)
```

Out[93]:

False

In [99]:

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

Out[99]:

```
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]]])
```

In [98]:

```
1 a[0, 0, 0]
```

Out[98]:

0

In [100]:

```
1 a[1, 0, 2]
```

Out[100]:

14

In [101]:

```
1 np.sum(a, axis=0) # 2, 3, 4 ---> 1, 3, 4
```

Out[101]:

```
array([[12, 14, 16, 18],
       [20, 22, 24, 26],
       [28, 30, 32, 34]])
```

In [102]:

```
1 np.sum(a, axis=1) # 2, 3, 4 --> 2, 1, 4
```

Out[102]:

```
array([[12, 15, 18, 21],
       [48, 51, 54, 57]])
```

In [103]:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
```

In [104]:

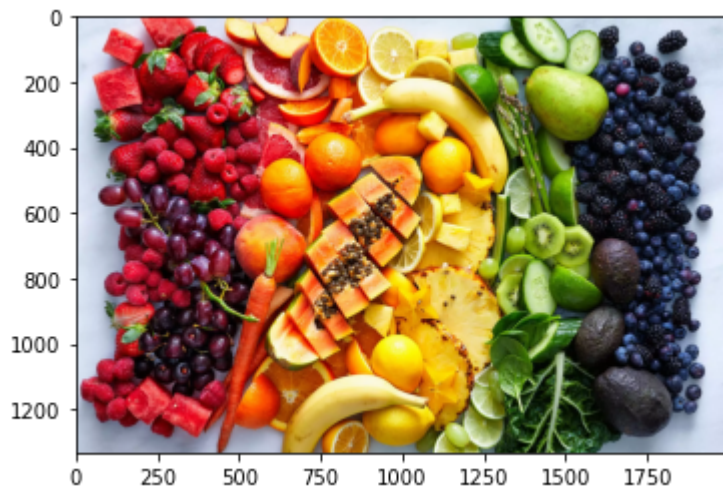
```
1 # imread
2 # imshow
3 # imsave
```

In [107]:

```
1 img = np.array(plt.imread("fruits.png"))
2 plt.imshow(img)
```

Out[107]:

<matplotlib.image.AxesImage at 0x7fc048558370>



In [108]:

```
1 img.ndim
```

Out[108]:

3

In [109]:

```
1 img.shape
```

Out[109]:

(1333, 2000, 3)

In []:

```
1 # [0, 225] - int
```

In []:

```
1 # [255, 0, 0] - RED
```

In [110]:

```
1 # [0, 255, 0] - GREEN
```

In [111]:

```
1 # [0, 0, 255] - BLUE
```

In [112]:

```
1 # [255, 255, 0] - YELLOW  
2 # 0-255=256 2^8=256
```

In [115]:

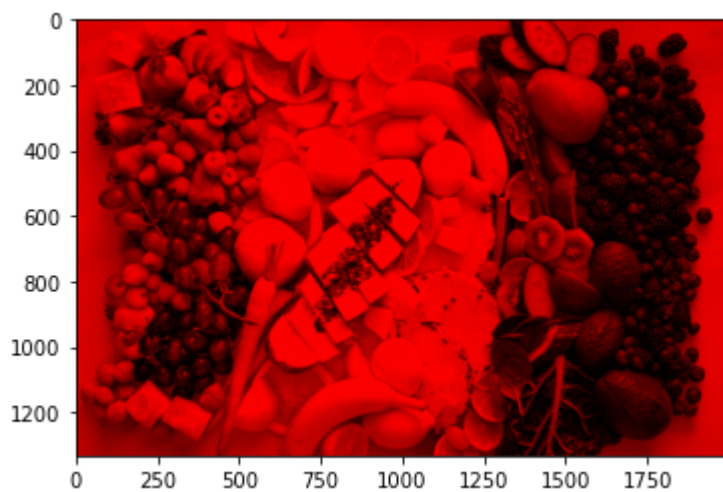
```
1 img_r = img.copy()  
2 img_r[:, :, (1, 2)] = 0
```

In [116]:

```
1 plt.imshow(img_r)
```

Out[116]:

<matplotlib.image.AxesImage at 0x7fc0485cf9d0>

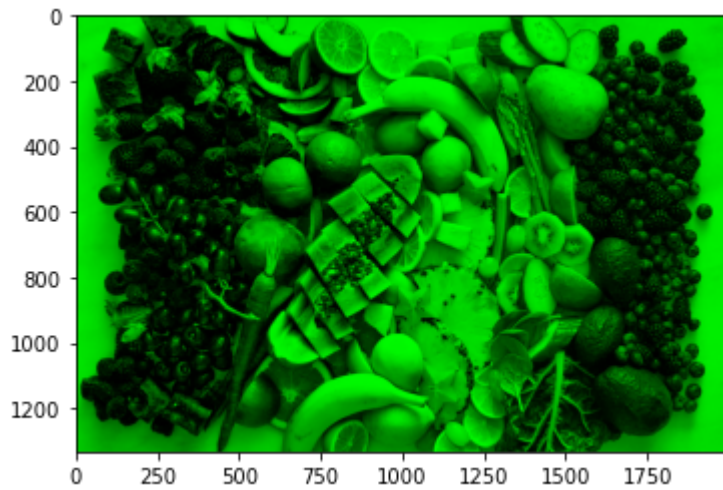


In [117]:

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

Out[117]:

<matplotlib.image.AxesImage at 0x7fc0182439a0>

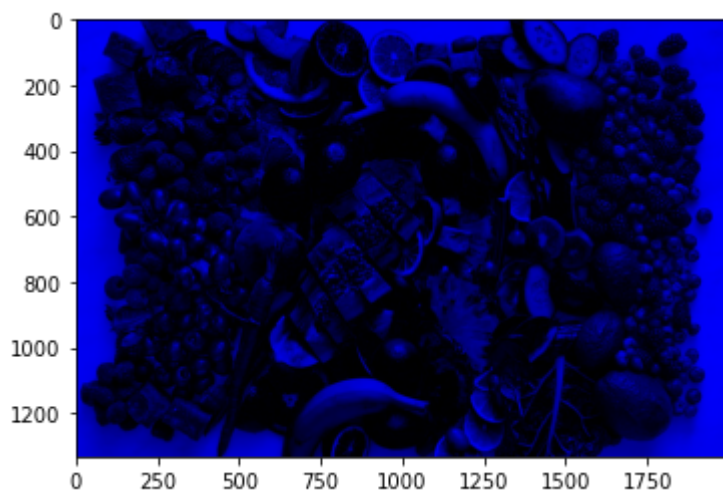


In [118]:

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

Out[118]:

<matplotlib.image.AxesImage at 0x7fc0182880d0>

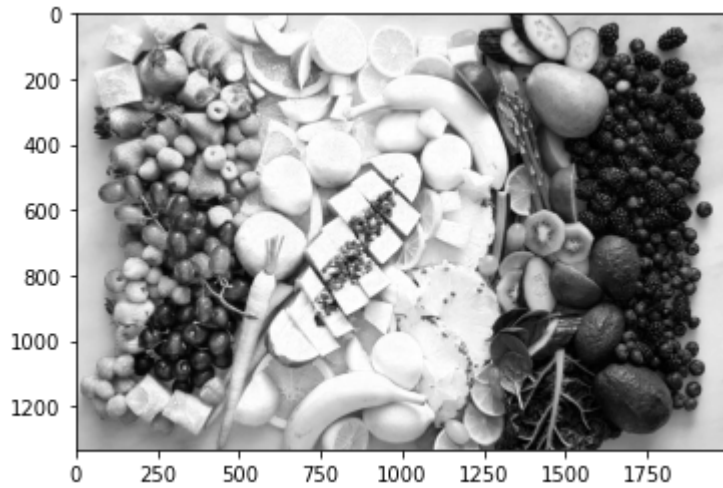


In [126]:

```
1 # red
2 img_0C = img[:, :, 0]
3 plt.imshow(img_0C, cmap="gray")
```

Out[126]:

<matplotlib.image.AxesImage at 0x7fc01825e220>

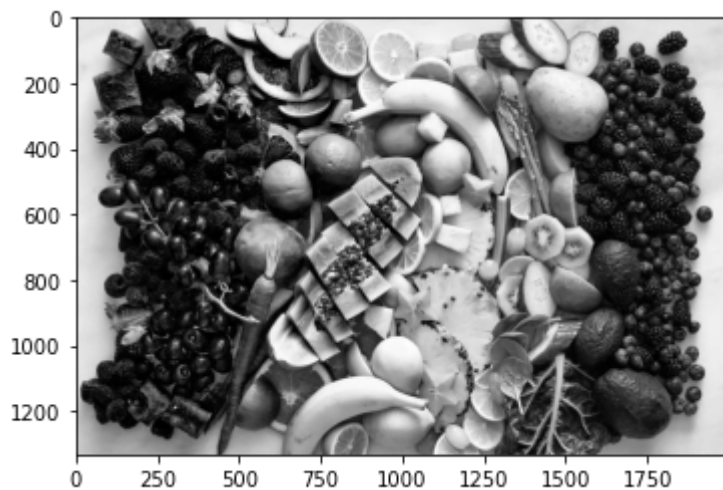


In [127]:

```
1 # green
2 img_0C = img[:, :, 1]
3 plt.imshow(img_0C, cmap="gray")
```

Out[127]:

<matplotlib.image.AxesImage at 0x7fbfe807a610>

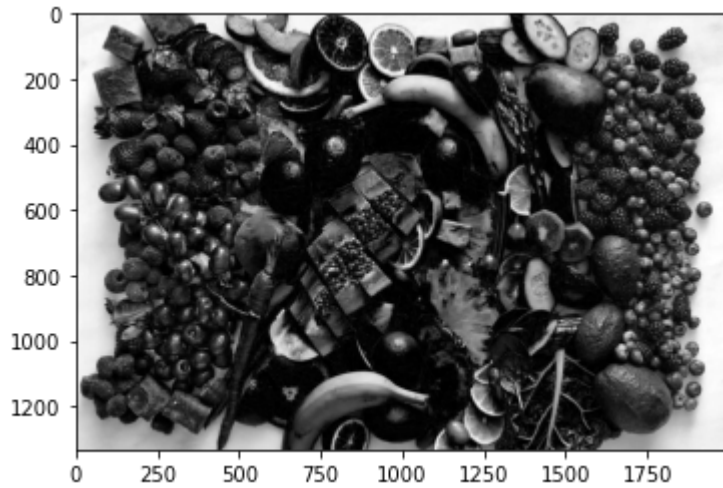


In [129]:

```
1 # blue
2 img_0C = img[:, :, 2]
3 plt.imshow(img_0C, cmap="gray")
```

Out[129]:

<matplotlib.image.AxesImage at 0x7fc00813f460>



In [132]:

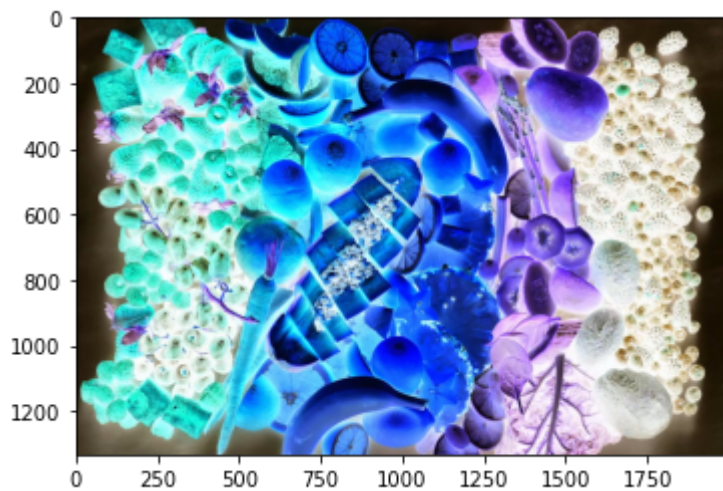
```
1 img_neg = 1 - img
```

In [133]:

```
1 plt.imshow(img_neg)
```

Out[133]:

<matplotlib.image.AxesImage at 0x7fbff802d0a0>

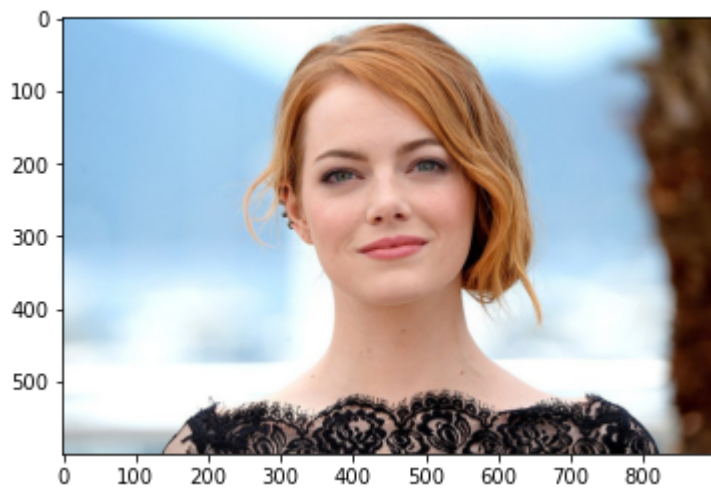


In [136]:

```
1 img = np.array(plt.imread('emma_stone.jpeg'))  
2 plt.imshow(img)
```

Out[136]:

<matplotlib.image.AxesImage at 0x7fc018602f40>



In [137]:

```
1 img.shape
```

Out[137]:

(600, 900, 3)

In [143]:

```
1 img_face = img[:,450, 225:675, :]
```

In [144]:

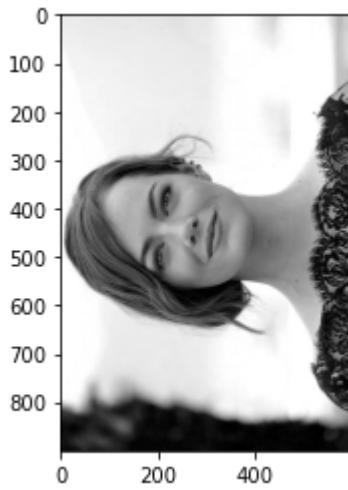
```
1 plt.imsave("crop_face.jpg", img_face)
```

In [146]:

```
1 plt.imshow(img[:, :, 2].T, cmap="gray")
```

Out[146]:

<matplotlib.image.AxesImage at 0x7fc00817fa00>



In [150]:

```
1 plt.imshow(img.T)
```

```
-----
-----
TypeError                                Traceback (most recent call
last)
<ipython-input-150-a83b1936c18a> in <module>
----> 1 plt.imshow(img.T)

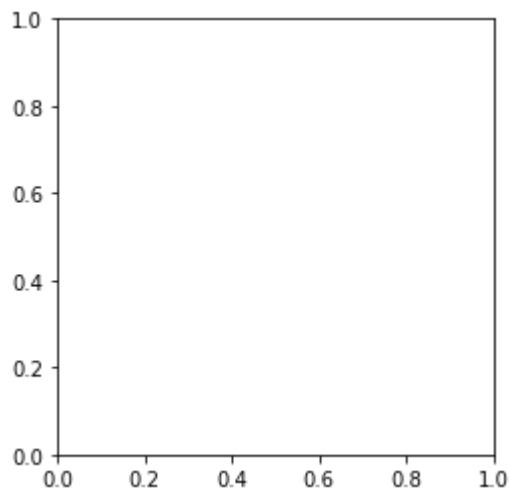
~/opt/anaconda3/lib/python3.8/site-packages/matplotlib/pyplot.py in im
show(X, cmap, norm, aspect, interpolation, alpha, vmin, vmax, origin,
extent, filternorm, filterrad, resample, url, data, **kwargs)
   2722         filternorm=True, filterrad=4.0, resample=None, url=None
e,
   2723         data=None, **kwargs):
-> 2724     __ret = gca().imshow(
   2725         X, cmap=cmap, norm=norm, aspect=aspect,
   2726         interpolation=interpolation, alpha=alpha, vmin=vmin,

~/opt/anaconda3/lib/python3.8/site-packages/matplotlib/__init__.py in
inner(ax, data, *args, **kwargs)
   1445     def inner(ax, *args, data=None, **kwargs):
   1446         if data is None:
-> 1447             return func(ax, *map(sanitize_sequence, args), **k
kwargs)
   1448
   1449         bound = new_sig.bind(ax, *args, **kwargs)

~/opt/anaconda3/lib/python3.8/site-packages/matplotlib/axes/_axes.py i
n imshow(self, X, cmap, norm, aspect, interpolation, alpha, vmin, vma
x, origin, extent, filternorm, filterrad, resample, url, **kwargs)
   5521             resample=resample, **kwargs)
   5522
-> 5523         im.set_data(X)
   5524         im.set_alpha(alpha)
   5525         if im.get_clip_path() is None:

~/opt/anaconda3/lib/python3.8/site-packages/matplotlib/image.py in set
_data(self, A)
   709         if not (self._A.ndim == 2
   710                 or self._A.ndim == 3 and self._A.shape[-1] in
[3, 4]):
-> 711             raise TypeError("Invalid shape {} for image data"
712                             .format(self._A.shape))
713

TypeError: Invalid shape (3, 900, 600) for image data
```



In [151]:

```
1 img.shape
```

Out[151]:

```
(600, 900, 3)
```

In [152]:

```
1 img.T.shape
```

Out[152]:

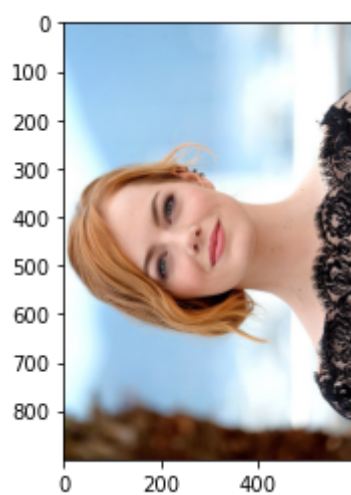
```
(3, 900, 600)
```

In [154]:

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

Out[154]:

<matplotlib.image.AxesImage at 0x7fc048a1f160>

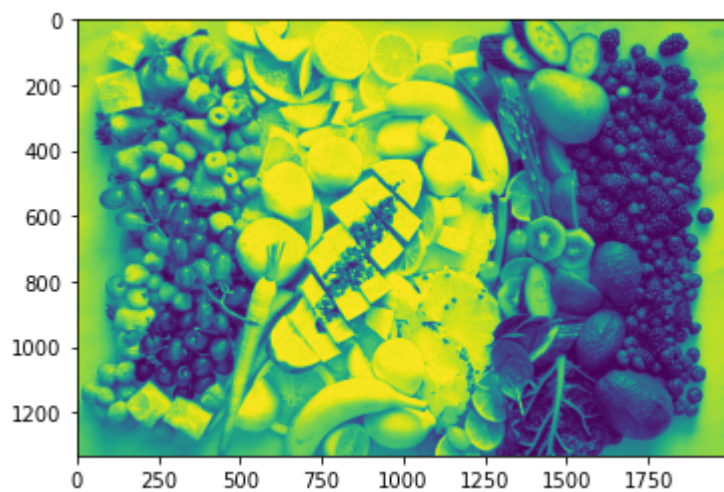


In [155]:

```
1 img = np.array(plt.imread("fruits.png"))  
2 plt.imshow(img[:, :, 0])
```

Out[155]:

<matplotlib.image.AxesImage at 0x7fc01833dd60>

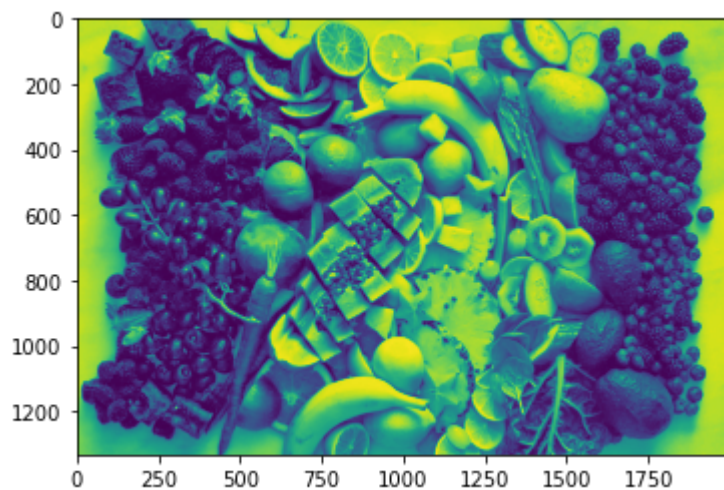


In [156]:

```
1 plt.imshow(img[:, :, 1])
```

Out[156]:

<matplotlib.image.AxesImage at 0x7fc0186ebfa0>

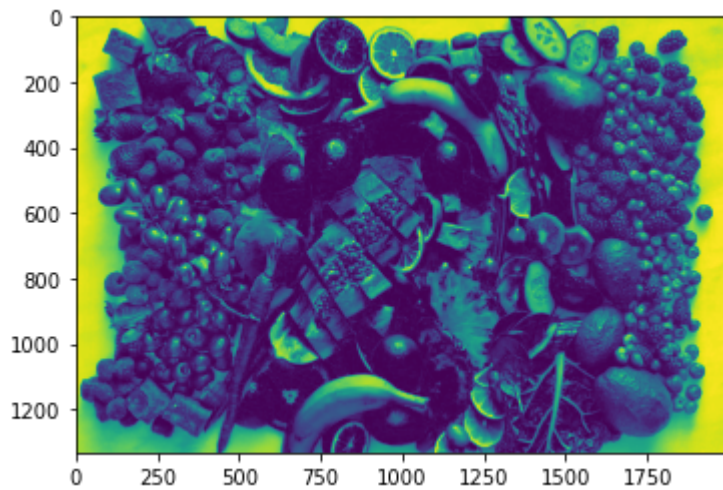


In [157]:

```
1 plt.imshow(img[:, :, 2])
```

Out[157]:

<matplotlib.image.AxesImage at 0x7fc0384f0190>



In [158]:

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

Out[158]:

```
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]])
```

In [164]:

```
1 np.split(a, (1, 2), 1)
```

Out[164]:

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