```
import numpy as np
cars = np.array(['chevrolet chevelle malibu', 'buick skylark 320',
'plymouth satellite', 'amc rebel sst', 'ford torino'])
cars[1]
```

```
{"type":"string"}
```

```
car_names = ['chevrolet chevelle malibu', 'buick skylark 320',
'plymouth satellite', 'amc rebel sst', 'ford torino']
horsepower = [130, 165, 150, 150, 140]
car_arr = np.array([car_names, horsepower])
print(car_arr)
```

```
[['chevrolet chevelle malibu' 'buick skylark 320' 'plymouth satellite'
  'amc rebel sst' 'ford torino']
 ['130' '165' '150' '150' '140']]
```

```
import numpy as np
student_marks=[78, 92, 36, 64, 89]
student_marks_arr = np.array(student_marks)
x=np.sum(student_marks_arr)
print(x)
```

```
359
```

```
car_names = ['chevrolet chevelle malibu', 'buick skylark 320',
'plymouth satellite', 'amc rebel sst', 'ford torino']
horsepower = [130, 165, 150, 150, 140]
car_hp_arr = np.array([car_names, horsepower])
car_hp_arr[1,1]
```

```
{"type":"string"}
```

```
cars = np.array(['chevrolet chevelle malibu', 'buick skylark 320',
'plymouth satellite', 'amc rebel sst', 'ford torino'])
cars[1:4]
```

```
array(['buick skylark 320', 'plymouth satellite', 'amc rebel sst'],
      dtype='<U25')
```

```
car_names = ['chevrolet chevelle malibu', 'buick skylark 320',
'plymouth satellite', 'amc rebel sst']
horsepower = [130, 165, 150, 150]
acceleration = [18, 15, 18, 16]
car_hp_acc_arr = np.array([car_names, horsepower, acceleration])
car_hp_acc_arr[0:2, 3:5]
```

```
array([['amc rebel sst'],
       ['150']], dtype='<U25')
```

```python
import numpy as np
horsepower = [130, 165, 150, 150, 140]
horsepower_arr = np.array(horsepower)
print("Mean horsepower = ",np.mean(horsepower_arr))
```

```
Mean horsepower =  147.0
```

```python
import numpy as np
horsepower = [130, 165, 150, 150, 140]
horsepower_arr = np.array(horsepower)
Mean=np.mean(horsepower_arr)
print("Mean horsepower = ", Mean)
```

```
Mean horsepower =  147.0
```

```python
import numpy as np
horsepower = [130, 165, 150, 150, 140]
horsepower_arr = np.array(horsepower, dtype='int')
print("Mean horsepower = ",np.mean(horsepower_arr))
```

```
Mean horsepower =  147.0
```

```python
horsepower = [130, 165, 150, 150, 140]
horsepower_arr = np.array(horsepower)
power=[1,2,3,4,5]
power_arr = np.array(power)
power_arr
horsepower_arr
```

```
array([130, 165, 150, 150, 140])
```

```python
horsepower = [130, 165, 150, 150, 140]
horsepower_arr = np.array(horsepower)
power=[1,2,3,4,5]
power_arr = np.array(power)
print(power_arr)
horsepower_arr
```

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

```
array([130, 165, 150, 150, 140])
```

```python
horsepower = [130, 165, 150, 150, 140]
horsepower_arr = np.array(horsepower)
power=[1,2,3,4,5]
power_arr = np.array(power)
```

```
horsepower_arr
power_arr
```

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

```
horsepower = [130, 165, 150, 150, 140]
horsepower_arr = np.array(horsepower)
filter= horsepower_arr > 135
newarr = horsepower_arr[filter_arr]
print(filter_arr)
horsepower_arr[filter]
```

```
[False  True  True  True  True]
```

```
array([165, 150, 150, 140])
```

```
horsepower = [130, 165, 150, 150, 140]
horsepower_arr = np.array(horsepower)
x = np.where(horsepower_arr > 135)
print(x)
horsepower_arr[x]
```

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

```
array([165, 150, 150, 140])
```

```
student_marks=[78, 92, 36, 64, 89]
student_marks_arr = np.array(student_marks)
additional_marks = [3.4, 6.2, 8.9, 5.0, 2.5]
additional_marks_arr = np.array(additional_marks)
final_student_marks_arr=student_marks_arr+additional_marks
final_student_marks_arr
```

```
array([81.4, 98.2, 44.9, 69. , 91.5])
```

```
import numpy as np
student_marks_arr = np.array([3, 9])
additional_marks = [2, 2]
final_marks_arr = np.power(student_marks_arr, additional_marks)
final_marks_arr
```

```
array([ 9, 81])
```

```
import numpy as np
array1=np.array([5, 10, 15])
array2=np.array([5])
```

```
array3= array1+array2
array3
```

```
array([10, 15, 20])
```

```
from skimage import data
img_list=dir(data)
print(img_list)
```

```
['astronaut', 'binary_blobs', 'brain', 'brick', 'camera', 'cat',
'cell', 'cells3d', 'checkerboard', 'chelsea', 'clock', 'coffee',
'coins', 'colorwheel', 'data_dir', 'download_all', 'eagle',
'file_hash', 'grass', 'gravel', 'horse', 'hubble_deep_field',
'human_mitosis', 'immunohistochemistry', 'kidney',
'lbp_frontal_face_cascade_filename', 'lfw_subset', 'lily', 'logo',
'microaneurysms', 'moon', 'nickel_solidification', 'page',
'palisades_of_vogt', 'protein_transport', 'retina', 'rocket',
'shepp_logan_phantom', 'skin', 'stereo_motorcycle', 'text', 'vortex']
```

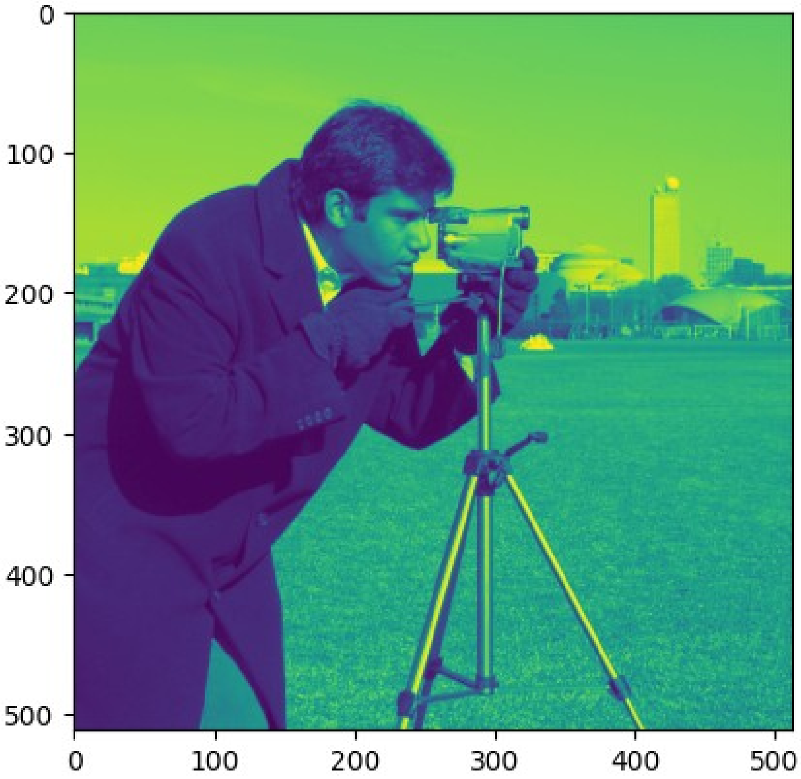
```
from skimage import data
print(dir(data))
```

```
['astronaut', 'binary_blobs', 'brain', 'brick', 'camera', 'cat',
'cell', 'cells3d', 'checkerboard', 'chelsea', 'clock', 'coffee',
'coins', 'colorwheel', 'data_dir', 'download_all', 'eagle',
'file_hash', 'grass', 'gravel', 'horse', 'hubble_deep_field',
'human_mitosis', 'immunohistochemistry', 'kidney',
'lbp_frontal_face_cascade_filename', 'lfw_subset', 'lily', 'logo',
'microaneurysms', 'moon', 'nickel_solidification', 'page',
'palisades_of_vogt', 'protein_transport', 'retina', 'rocket',
'shepp_logan_phantom', 'skin', 'stereo_motorcycle', 'text', 'vortex']
```

```
from skimage import data
import matplotlib.pyplot as plt
img = data.camera()
print('Type of image: ', type(img))
print('Dimensions of image: ', img.ndim)
print('Shape of image:', img.shape)
plt.imshow(img)
```

```
Type of image:  <class 'numpy.ndarray'>
Dimensions of image:  2
Shape of image: (512, 512)
```

```
<matplotlib.image.AxesImage at 0x7aff582c7040>
```

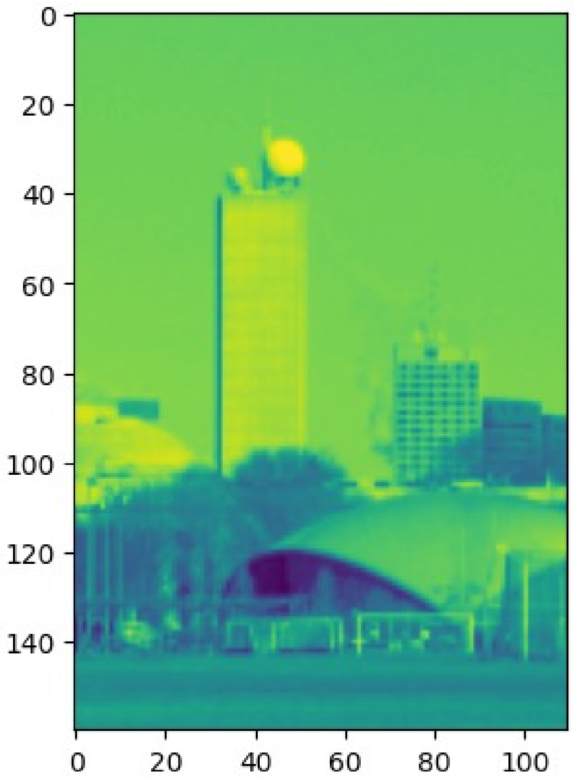


```
from skimage import data
import matplotlib.pyplot as plt
img = data.camera()
print('Type of image: ', type(img))
print('Dimensions of image: ', img.ndim)
print('Shape of image:', img.shape)
print(img)
```

```
Type of image:  <class 'numpy.ndarray'>
Dimensions of image:  2
Shape of image: (512, 512)
[[200 200 200 ... 189 190 190]
 [200 199 199 ... 190 190 190]
 [199 199 199 ... 190 190 190]
 ...
 [ 25  25  27 ... 139 122 147]
 [ 25  25  26 ... 158 141 168]
 [ 25  25  27 ... 151 152 149]]
```

```
from skimage import data
import matplotlib.pyplot as plt
img = data.camera()
new_img = img.copy()
new_img = new_img[90:250, 380:490]
plt.imshow(new_img)
```

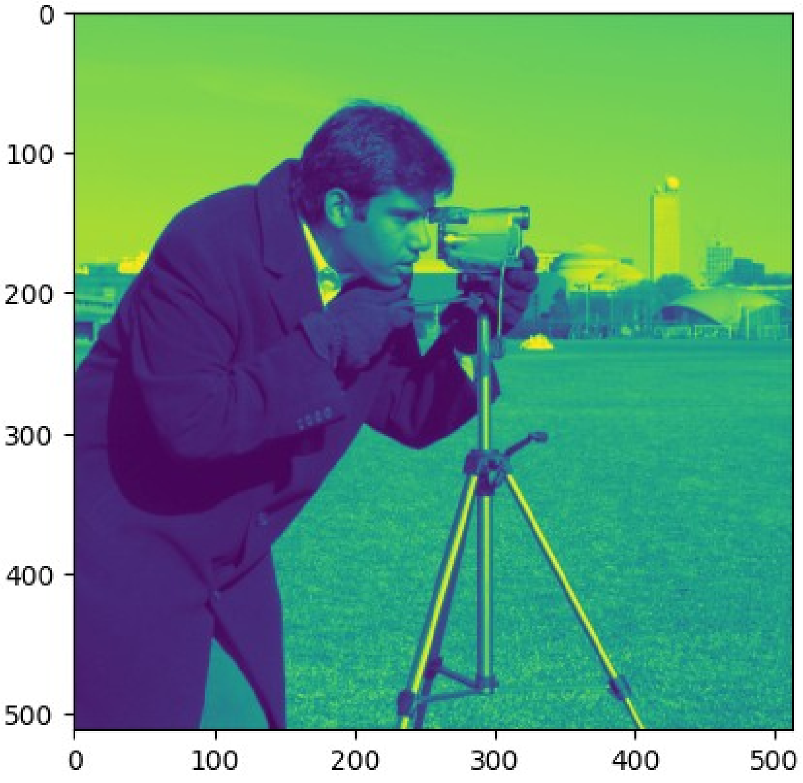
```
<matplotlib.image.AxesImage at 0x7d5447e13e80>
```

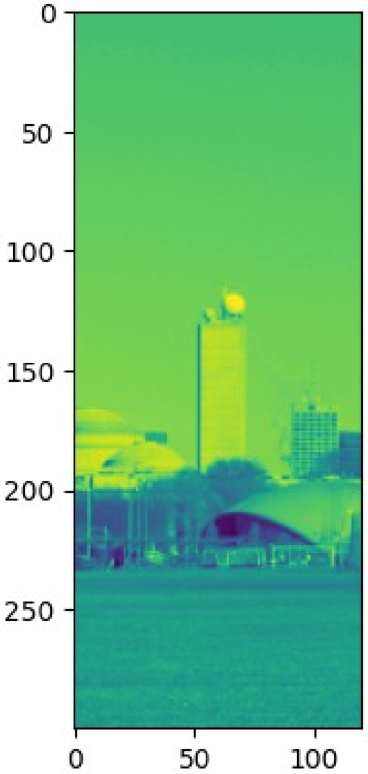


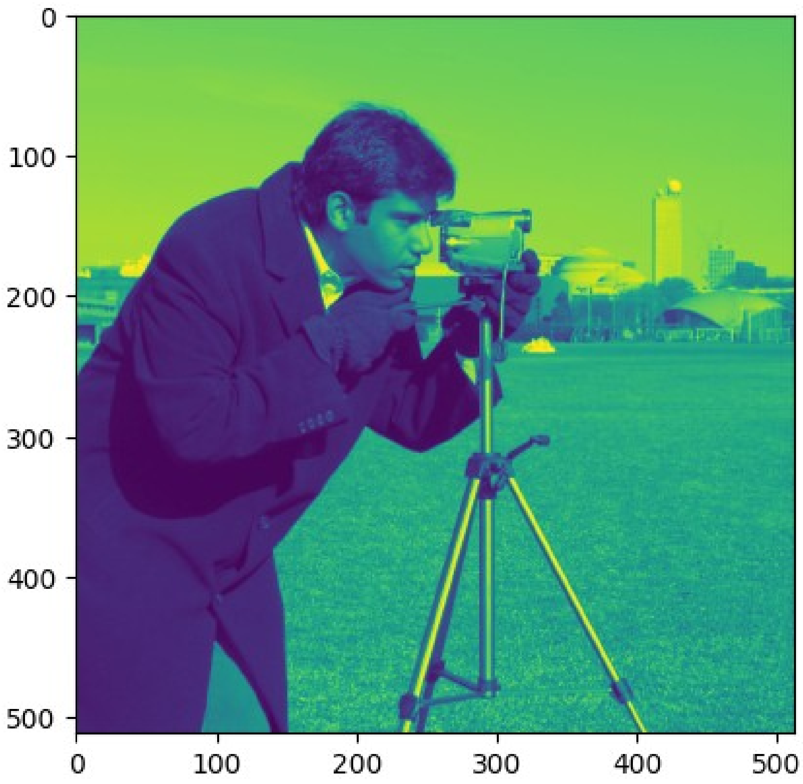
The name used to slice can be anything such as new_img, img_slice ... Also plt.figure is used to plot multiple images. If it's one image then its unnecessary

```
from skimage import data
import matplotlib.pyplot as plt
img = data.camera()
plt.imshow(img)
imgcopy= img.copy()
slice = img[0:300,360:480]
plt.figure()
plt.imshow(slice)
plt.figure()
plt.imshow(img)
```

```
<matplotlib.image.AxesImage at 0x7d5446ff7400>
```

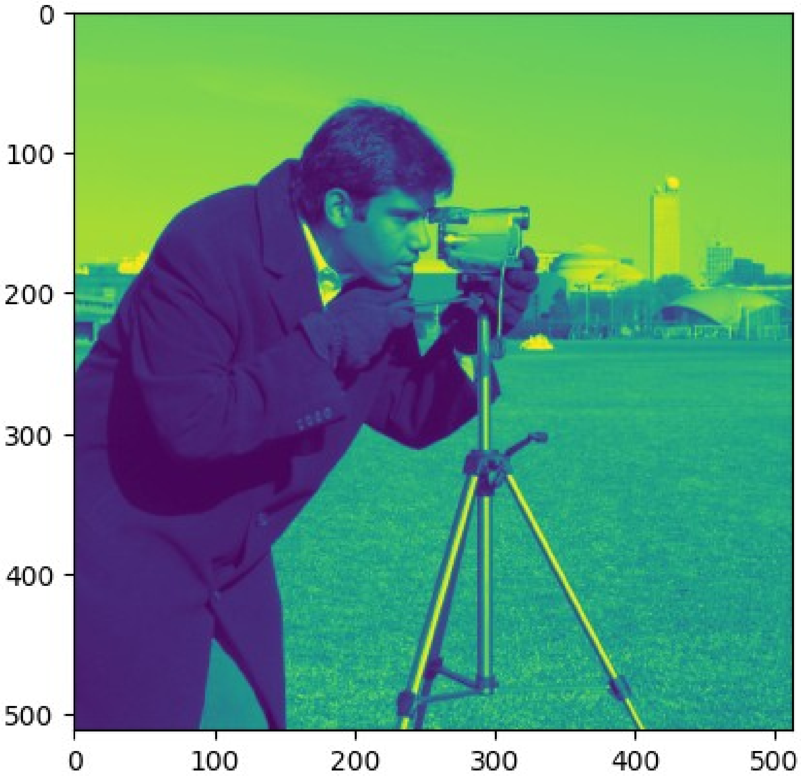

0
100
200
300
400
500
0
100
200
300
400
500

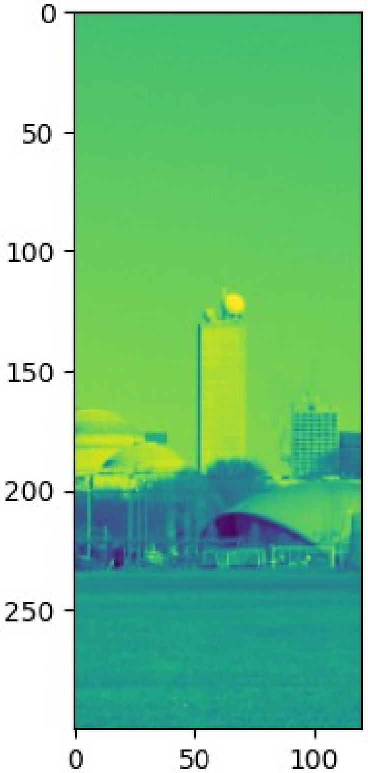

0
50
100
150
200
250
0
50
100

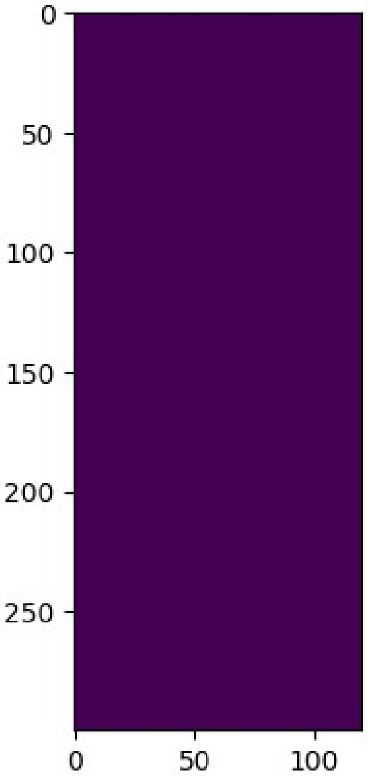


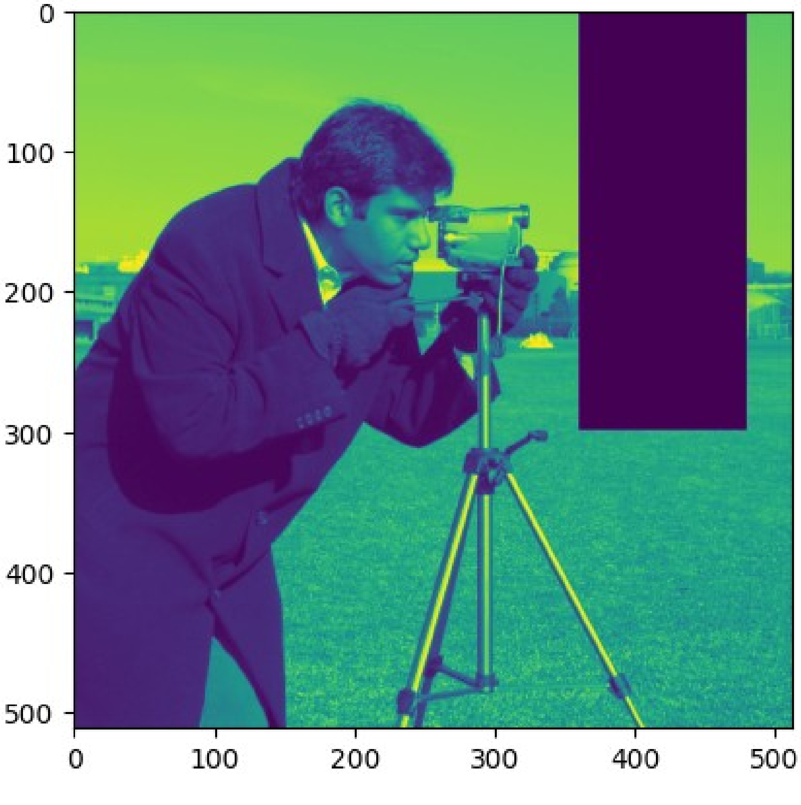
```python
from skimage import data
import matplotlib.pyplot as plt
img = data.camera()
plt.imshow(img)
imgcopy= img.copy()
slice = img[0:300,360:480]
plt.figure()
plt.imshow(slice)
img[0:300,360:480]=0
plt.figure()
plt.imshow(slice)
plt.figure()
plt.imshow(img)
plt.figure()
plt.imshow(imgcopy)
```

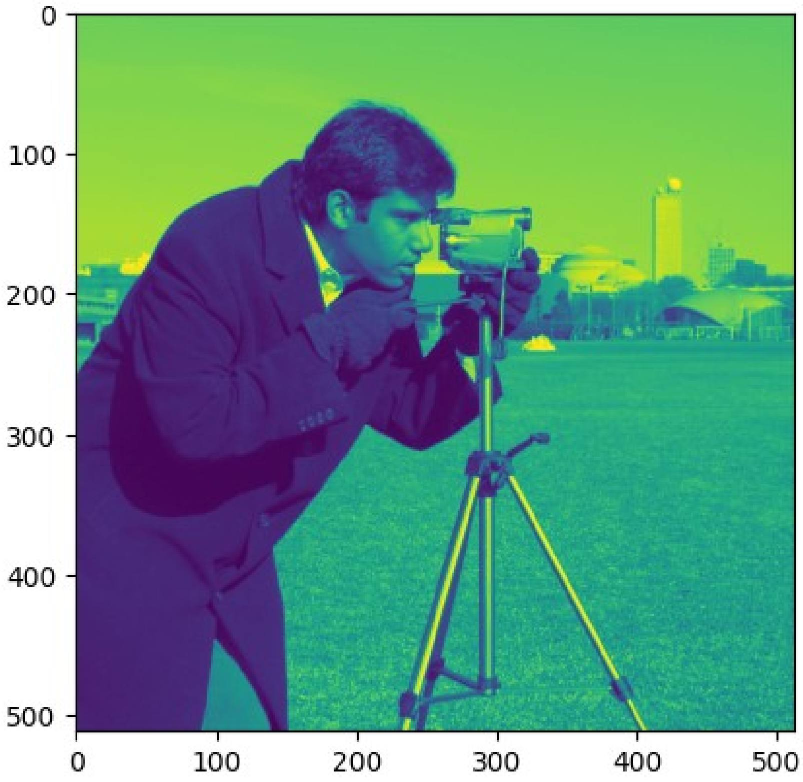
```
<matplotlib.image.AxesImage at 0x7d54468e5180>
```


0
100
200
300
400
500
0
100
200
300
400
500


0
50
100
150
200
250
0
50
100

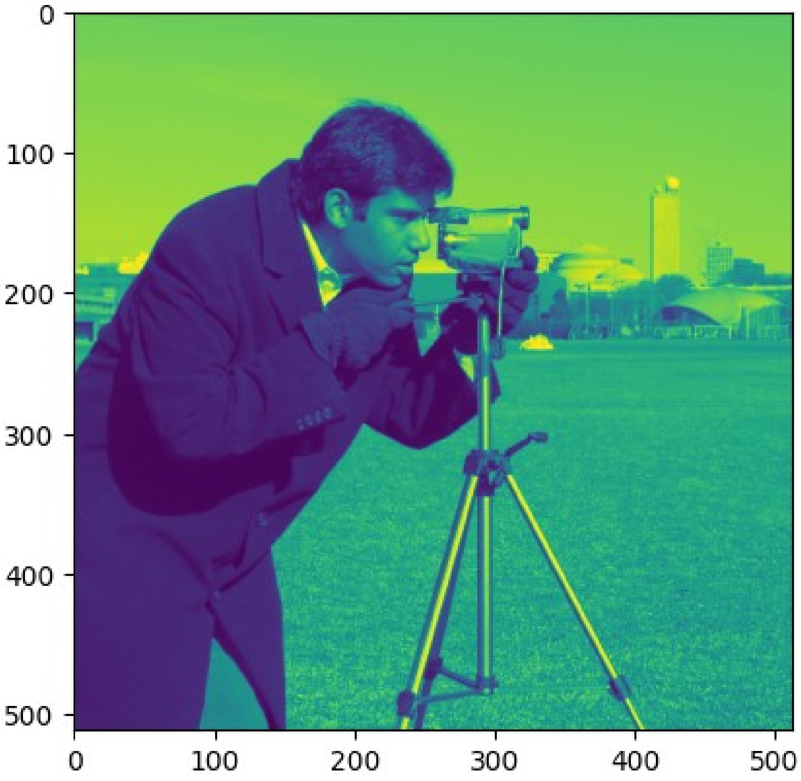

0
50
100
150
200
250
0
50
100

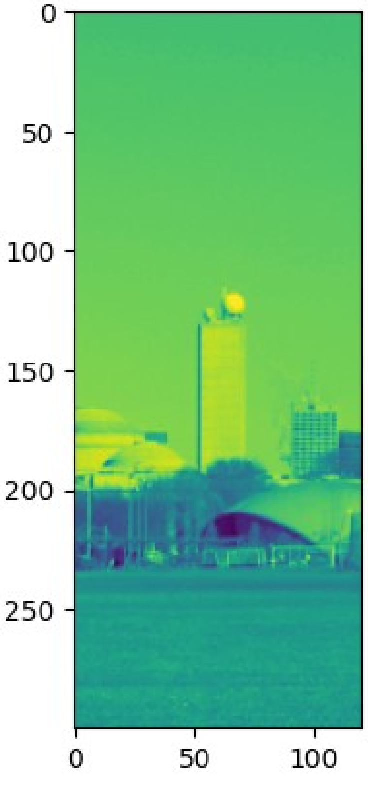

0
100
200
300
400
500
0
100
200
300
400
500

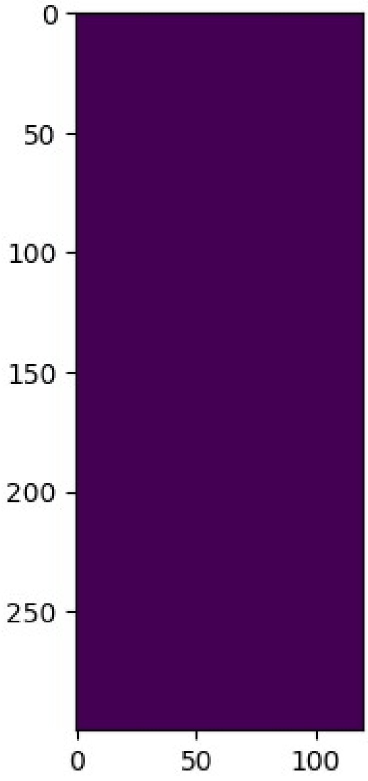


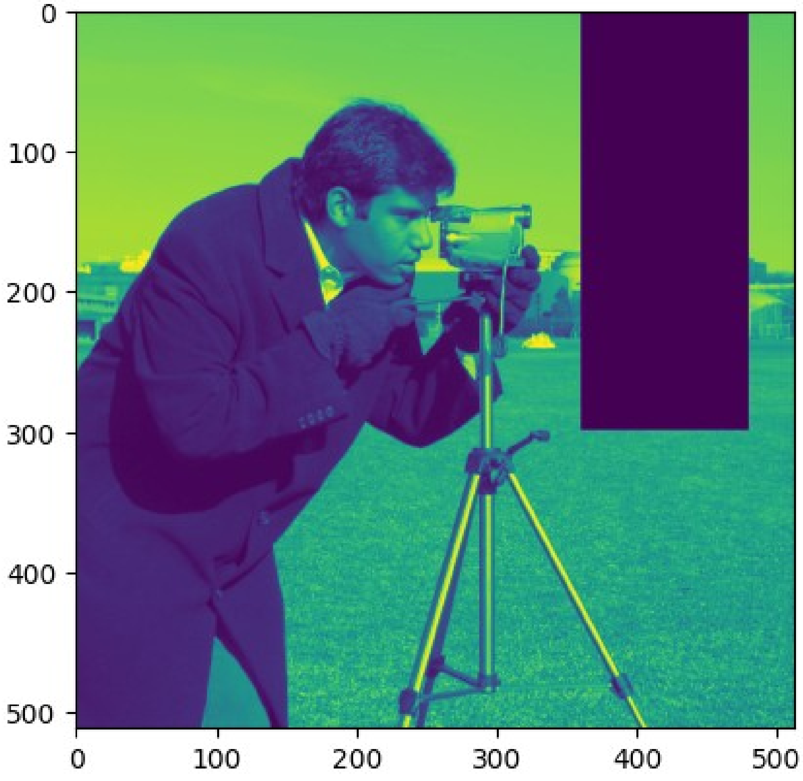
```
from skimage import data
import matplotlib.pyplot as plt
img = data.camera()
plt.imshow(img)

slice = img[0:300,360:480]
plt.figure()
plt.imshow(slice)
img[0:300,360:480]=0
plt.figure()
plt.imshow(slice)
plt.figure()
plt.imshow(img)
```

```
<matplotlib.image.AxesImage at 0x7d54468b8130>
```


0
100
200
300
400
500
0
100
200
300
400
500

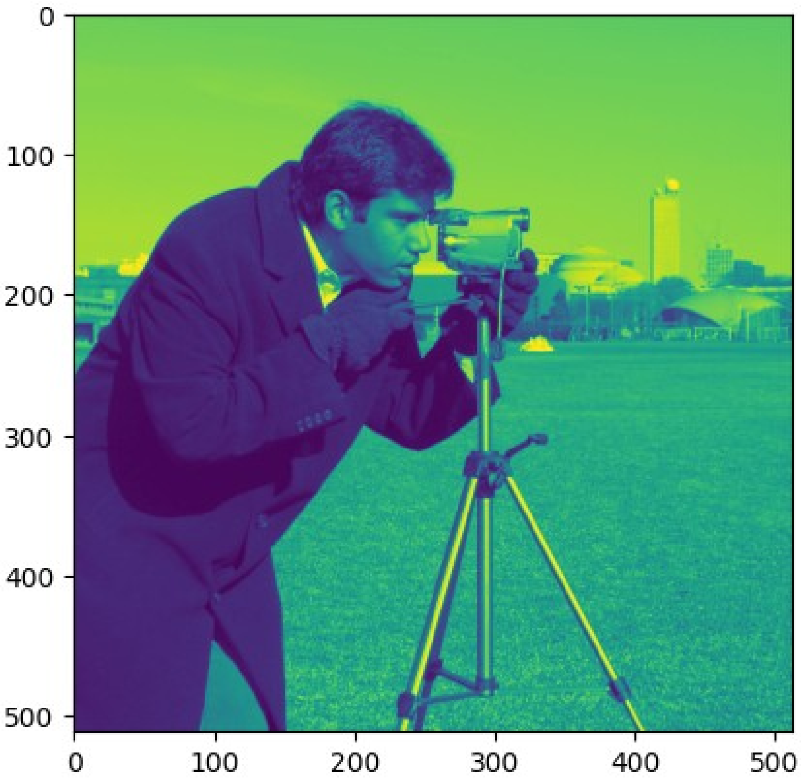

0
50
100
150
200
250
0
50
100

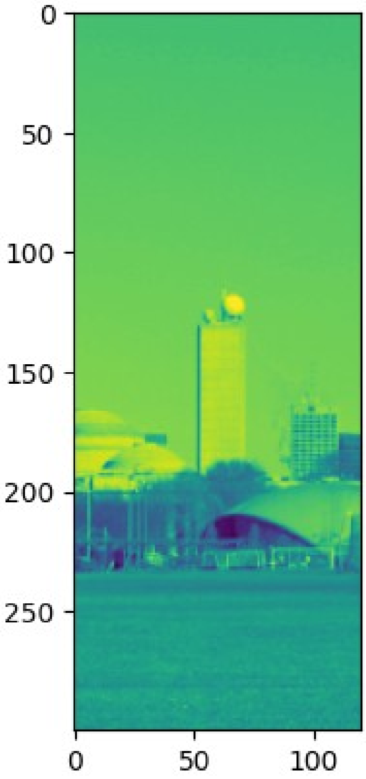

0
50
100
150
200
250
0
50
100

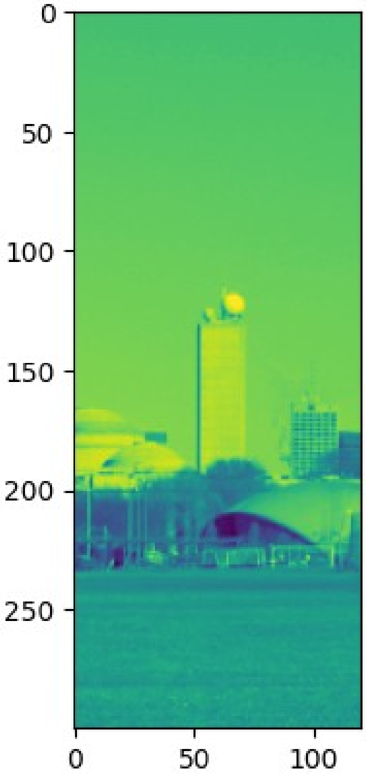

0
100
200
300
400
500
0
100
200
300
400
500

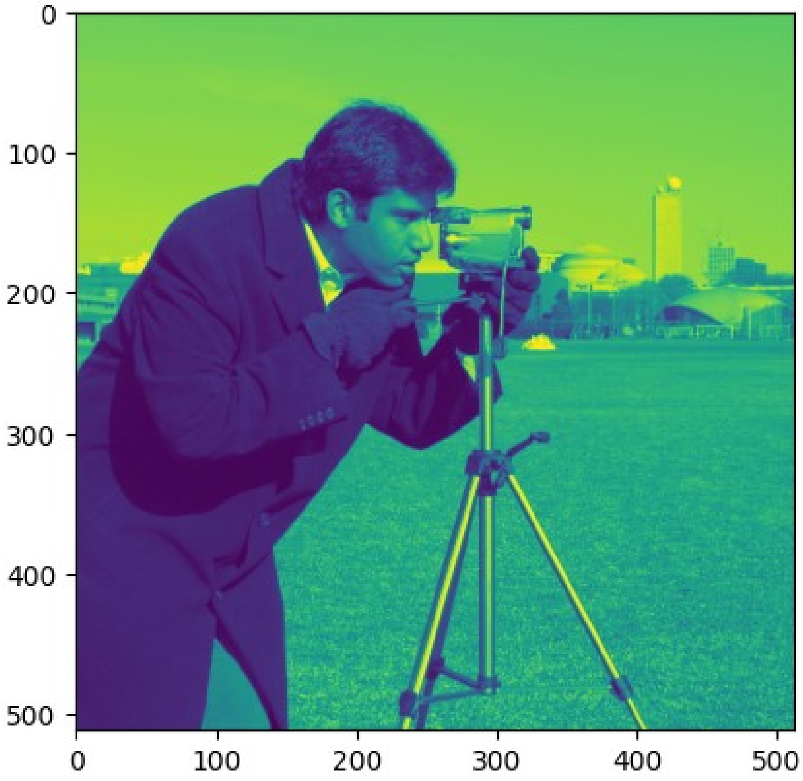
```python
from skimage import data
import matplotlib.pyplot as plt
img = data.camera()
plt.imshow(img)
imgcopy= img.copy()
slice = imgcopy[0:300,360:480]
plt.figure()
plt.imshow(slice)
img[0:300,360:480]=0
plt.figure()
plt.imshow(slice)
plt.figure()
plt.imshow(imgcopy)
plt.figure()
plt.imshow(img)
```

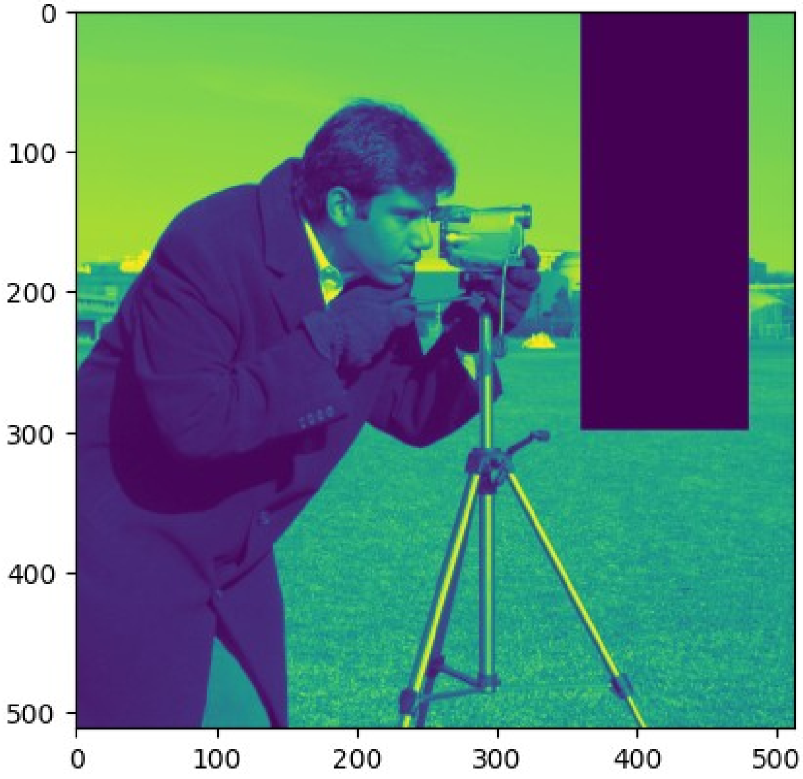
```
<matplotlib.image.AxesImage at 0x7ea1be073250>
```




0
50
100
150
200
250
0
50
100


0
50
100
150
200
250
0
50
100


0
100
200
300
400
500
0
100
200
300
400
500


0
100
200
300
400
500
0
100
200
300
400
500

```
Arange=np.arange(0,10000)
Arange

array([   0,    1,    2, ..., 9997, 9998, 9999])
```

Here it has default step of 1. But we can change it.

```
Arange=np.arange(0,10000, 10)
Arange

array([   0,   10,   20,   30,   40,   50,   60,   70,   80,   90,
100,
        110,  120,  130,  140,  150,  160,  170,  180,  190,  200,
210,
        220,  230,  240,  250,  260,  270,  280,  290,  300,  310,
320,
        330,  340,  350,  360,  370,  380,  390,  400,  410,  420,
430,
        440,  450,  460,  470,  480,  490,  500,  510,  520,  530,
540,
        550,  560,  570,  580,  590,  600,  610,  620,  630,  640,
650,
        660,  670,  680,  690,  700,  710,  720,  730,  740,  750,
760,
        770,  780,  790,  800,  810,  820,  830,  840,  850,  860,
870,
        880,  890,  900,  910,  920,  930,  940,  950,  960,  970,
980,
        990, 1000, 1010, 1020, 1030, 1040, 1050, 1060, 1070, 1080,
1090,
       1100, 1110, 1120, 1130, 1140, 1150, 1160, 1170, 1180, 1190,
1200,
       1210, 1220, 1230, 1240, 1250, 1260, 1270, 1280, 1290, 1300,
1310,
       1320, 1330, 1340, 1350, 1360, 1370, 1380, 1390, 1400, 1410,
1420,
       1430, 1440, 1450, 1460, 1470, 1480, 1490, 1500, 1510, 1520,
1530,
       1540, 1550, 1560, 1570, 1580, 1590, 1600, 1610, 1620, 1630,
1640,
       1650, 1660, 1670, 1680, 1690, 1700, 1710, 1720, 1730, 1740,
1750,
       1760, 1770, 1780, 1790, 1800, 1810, 1820, 1830, 1840, 1850,
1860,
       1870, 1880, 1890, 1900, 1910, 1920, 1930, 1940, 1950, 1960,
1970,
       1980, 1990, 2000, 2010, 2020, 2030, 2040, 2050, 2060, 2070,
2080,
       2090, 2100, 2110, 2120, 2130, 2140, 2150, 2160, 2170, 2180,
2190,
       2200, 2210, 2220, 2230, 2240, 2250, 2260, 2270, 2280, 2290,
2300,
       2310, 2320, 2330, 2340, 2350, 2360, 2370, 2380, 2390, 2400,
2410,
       2420, 2430, 2440, 2450, 2460, 2470, 2480, 2490, 2500, 2510,
2520,
       2530, 2540, 2550, 2560, 2570, 2580, 2590, 2600, 2610, 2620,
2630,
       2640, 2650, 2660, 2670, 2680, 2690, 2700, 2710, 2720, 2730,
2740,
       2750, 2760, 2770, 2780, 2790, 2800, 2810, 2820, 2830, 2840,
2850,
       2860, 2870, 2880, 2890, 2900, 2910, 2920, 2930, 2940, 2950,
2960,
       2970, 2980, 2990, 3000, 3010, 3020, 3030, 3040, 3050, 3060,
3070,
       3080, 3090, 3100, 3110, 3120, 3130, 3140, 3150, 3160, 3170,
3180,
       3190, 3200, 3210, 3220, 3230, 3240, 3250, 3260, 3270, 3280,
3290,
       3300, 3310, 3320, 3330, 3340, 3350, 3360, 3370, 3380, 3390,
3400,
       3410, 3420, 3430, 3440, 3450, 3460, 3470, 3480, 3490, 3500,
3510,
       3520, 3530, 3540, 3550, 3560, 3570, 3580, 3590, 3600, 3610,
3620,
       3630, 3640, 3650, 3660, 3670, 3680, 3690, 3700, 3710, 3720,
3730,
       3740, 3750, 3760, 3770, 3780, 3790, 3800, 3810, 3820, 3830,
3840,
       3850, 3860, 3870, 3880, 3890, 3900, 3910, 3920, 3930, 3940,
3950,
       3960, 3970, 3980, 3990, 4000, 4010, 4020, 4030, 4040, 4050,
4060,
       4070, 4080, 4090, 4100, 4110, 4120, 4130, 4140, 4150, 4160,
4170,
       4180, 4190, 4200, 4210, 4220, 4230, 4240, 4250, 4260, 4270,
4280,
       4290, 4300, 4310, 4320, 4330, 4340, 4350, 4360, 4370, 4380,
4390,
       4400, 4410, 4420, 4430, 4440, 4450, 4460, 4470, 4480, 4490,
4500,
       4510, 4520, 4530, 4540, 4550, 4560, 4570, 4580, 4590, 4600,
4610,
       4620, 4630, 4640, 4650, 4660, 4670, 4680, 4690, 4700, 4710,
4720,
       4730, 4740, 4750, 4760, 4770, 4780, 4790, 4800, 4810, 4820,
4830,
       4840, 4850, 4860, 4870, 4880, 4890, 4900, 4910, 4920, 4930,
4940,
       4950, 4960, 4970, 4980, 4990, 5000, 5010, 5020, 5030, 5040,
5050,
       5060, 5070, 5080, 5090, 5100, 5110, 5120, 5130, 5140, 5150,
5160,
       5170, 5180, 5190, 5200, 5210, 5220, 5230, 5240, 5250, 5260,
5270,
       5280, 5290, 5300, 5310, 5320, 5330, 5340, 5350, 5360, 5370,
5380,
       5390, 5400, 5410, 5420, 5430, 5440, 5450, 5460, 5470, 5480,
5490,
       5500, 5510, 5520, 5530, 5540, 5550, 5560, 5570, 5580, 5590,
5600,
       5610, 5620, 5630, 5640, 5650, 5660, 5670, 5680, 5690, 5700,
5710,
       5720, 5730, 5740, 5750, 5760, 5770, 5780, 5790, 5800, 5810,
5820,
       5830, 5840, 5850, 5860, 5870, 5880, 5890, 5900, 5910, 5920,
5930,
       5940, 5950, 5960, 5970, 5980, 5990, 6000, 6010, 6020, 6030,
6040,
       6050, 6060, 6070, 6080, 6090, 6100, 6110, 6120, 6130, 6140,
6150,
       6160, 6170, 6180, 6190, 6200, 6210, 6220, 6230, 6240, 6250,
6260,
       6270, 6280, 6290, 6300, 6310, 6320, 6330, 6340, 6350, 6360,
6370,
       6380, 6390, 6400, 6410, 6420, 6430, 6440, 6450, 6460, 6470,
6480,
       6490, 6500, 6510, 6520, 6530, 6540, 6550, 6560, 6570, 6580,
6590,
       6600, 6610, 6620, 6630, 6640, 6650, 6660, 6670, 6680, 6690,
6700,
       6710, 6720, 6730, 6740, 6750, 6760, 6770, 6780, 6790, 6800,
6810,
       6820, 6830, 6840, 6850, 6860, 6870, 6880, 6890, 6900, 6910,
6920,
       6930, 6940, 6950, 6960, 6970, 6980, 6990, 7000, 7010, 7020,
7030,
       7040, 7050, 7060, 7070, 7080, 7090, 7100, 7110, 7120, 7130,
7140,
       7150, 7160, 7170, 7180, 7190, 7200, 7210, 7220, 7230, 7240,
7250,
       7260, 7270, 7280, 7290, 7300, 7310, 7320, 7330, 7340, 7350,
7360,
       7370, 7380, 7390, 7400, 7410, 7420, 7430, 7440, 7450, 7460,
7470,
       7480, 7490, 7500, 7510, 7520, 7530, 7540, 7550, 7560, 7570,
7580,
          7590, 7600, 7610, 7620, 7630, 7640, 7650, 7660, 7670, 7680,
7690,
          7700, 7710, 7720, 7730, 7740, 7750, 7760, 7770, 7780, 7790,
7800,
          7810, 7820, 7830, 7840, 7850, 7860, 7870, 7880, 7890, 7900,
7910,
          7920, 7930, 7940, 7950, 7960, 7970, 7980, 7990, 8000, 8010,
8020,
          8030, 8040, 8050, 8060, 8070, 8080, 8090, 8100, 8110, 8120,
8130,
          8140, 8150, 8160, 8170, 8180, 8190, 8200, 8210, 8220, 8230,
8240,
          8250, 8260, 8270, 8280, 8290, 8300, 8310, 8320, 8330, 8340,
8350,
          8360, 8370, 8380, 8390, 8400, 8410, 8420, 8430, 8440, 8450,
8460,
          8470, 8480, 8490, 8500, 8510, 8520, 8530, 8540, 8550, 8560,
8570,
          8580, 8590, 8600, 8610, 8620, 8630, 8640, 8650, 8660, 8670,
8680,
          8690, 8700, 8710, 8720, 8730, 8740, 8750, 8760, 8770, 8780,
8790,
          8800, 8810, 8820, 8830, 8840, 8850, 8860, 8870, 8880, 8890,
8900,
          8910, 8920, 8930, 8940, 8950, 8960, 8970, 8980, 8990, 9000,
9010,
          9020, 9030, 9040, 9050, 9060, 9070, 9080, 9090, 9100, 9110,
9120,
          9130, 9140, 9150, 9160, 9170, 9180, 9190, 9200, 9210, 9220,
9230,
          9240, 9250, 9260, 9270, 9280, 9290, 9300, 9310, 9320, 9330,
9340,
          9350, 9360, 9370, 9380, 9390, 9400, 9410, 9420, 9430, 9440,
9450,
          9460, 9470, 9480, 9490, 9500, 9510, 9520, 9530, 9540, 9550,
9560,
          9570, 9580, 9590, 9600, 9610, 9620, 9630, 9640, 9650, 9660,
9670,
          9680, 9690, 9700, 9710, 9720, 9730, 9740, 9750, 9760, 9770,
9780,
          9790, 9800, 9810, 9820, 9830, 9840, 9850, 9860, 9870, 9880,
9890,
          9900, 9910, 9920, 9930, 9940, 9950, 9960, 9970, 9980, 9990])
```

```
Arange=np.arange(0,10000, 5)
print(Arange)
len(Arange)
```

```
[   0    5   10 ... 9985 9990 9995]
```

```
2000
```

```
Linspace=np.linspace(0,10000)
print(Linspace)
len(Linspace)
```

```
[    0.           204.08163265   408.16326531   612.24489796
   816.32653061  1020.40816327  1224.48979592  1428.57142857
  1632.65306122  1836.73469388  2040.81632653  2244.89795918
  2448.97959184  2653.06122449  2857.14285714  3061.2244898
  3265.30612245  3469.3877551   3673.46938776  3877.55102041
  4081.63265306  4285.71428571  4489.79591837  4693.87755102
  4897.95918367  5102.04081633  5306.12244898  5510.20408163
  5714.28571429  5918.36734694  6122.44897959  6326.53061224
  6530.6122449   6734.69387755  6938.7755102   7142.85714286
  7346.93877551  7551.02040816  7755.10204082  7959.18367347
  8163.26530612  8367.34693878  8571.42857143  8775.51020408
  8979.59183673  9183.67346939  9387.75510204  9591.83673469
  9795.91836735 10000.        ]
```

```
50
```

Above one has default entry of 50. If we include values, it will change.

```
Linspace=np.linspace(0,10000, 2)
print(Linspace)
len(Linspace)
```

```
[    0. 10000.]
```

```
2
```