In [1]:
```
import numpy as np
```

Numpy review homework

1. Make a numpy matrix from a Python list of lists...

In [ ]:
```
pylist = [[2,2],[2,2],[2,2]]
nplist = np.array(pylist)

print(nplist)
```

```
[2 2 2 2 2]
```

2. Make a 3D numpy matrix from a Python list of lists of lists!

In [8]:
```
pylist2 = [[[2,2],[2,2]],[[2,2],[2,2]],[[2,2],[2,2]]]
nplist2 = np.array(pylist2)

print(nplist2)
```

```
[[[2 2]
  [2 2]]

 [[2 2]
  [2 2]]

 [[2 2]
  [2 2]]]
```

3. Create a 5x3 array of Gaussian random numbers.

In [9]:
```
gauss = np.random.randn(5,3)
```

4. Write a script to go through the array created in 3. and announce (print) the value and its row and column indexes.

Hint: Use nested `for` loops - one to loop through the rows and one to loop through the columns.

In [11]:
```
for i in range(3):
    for j in range(5):
        print(f"the value for the ({j},{i}) element is {gauss[j][i]}")
```

```
the value for the (0,0) element is -0.03250178398345126
the value for the (1,0) element is 1.7848077978128047
the value for the (2,0) element is -1.497595897258597
the value for the (3,0) element is 0.50795408335424
the value for the (4,0) element is 1.2428260012015317
the value for the (0,1) element is 0.26742566408420293
the value for the (1,1) element is 0.799528120643741
the value for the (2,1) element is -1.2729660206804616
the value for the (3,1) element is -0.11485039938668555
the value for the (4,1) element is -0.09841693217962338
the value for the (0,2) element is 0.4850464715987454
the value for the (1,2) element is -0.6366085597578014
the value for the (2,2) element is 0.4829550754398065
the value for the (3,2) element is -0.4209398165190519
the value for the (4,2) element is 0.19378879523993628
```

5. Make an new array out of your random numbers such that the mean is 10 and the standard deviation is 3.

In [13]:
```
standardGauss = gauss * 3 +10
standardGauss
```

Out[13]:
```
array([[ 9.90249465, 10.80227699, 11.45513941],
       [15.35442339, 12.39858436,  8.09017432],
       [ 5.50721231,  6.18110194, 11.44886523],
       [11.52386225,  9.6554488 ,  8.73718055],
       [13.728478  ,  9.7047492 , 10.58136639]])
```

6. Count the number of values in your new array that are below 7.

In [14]:
```
belowseven = []
for i in range(3):
    for j in range(5):
        if standardGauss[j][i] < 7:
            belowseven.append(standardGauss[j][i])
belowseven
```

Out[14]:
```
[5.507212308224209, 6.181101937958616]
```

7. Make a numpy sequence that has the even numbers from 2 up to (and including) 20.

In [17]:
```
np.arange(2,21,2)
```

Out[17]:
```
array([ 2,  4,  6,  8, 10, 12, 14, 16, 18, 20])
```

8. Get the second and third rows of your array created in #5.

In [21]:
```
newGauss = standardGauss[1:3,]
newGauss
```

Out[21]:
```
array([[15.35442339, 12.39858436,  8.09017432],
       [ 5.50721231,  6.18110194, 11.44886523]])
```

9. Compute the mean of the columns of your array created in #5.

In [23]:
```
newGauss.mean(0)
```

Out[23]:
```
array([10.43081785,  9.28984315,  9.76951977])
```