In [1]:

```
# Numpy
import numpy as np
```

In [ ]:

```
# numpy.percentile()
# numpy.percentile(a, q, axis)
# a  Input array
# q  The percentile to compute must be between 0-100
# axis The axis along which the percentile is to be calculated
```

In [2]:

```
# Percentile (or a centile) is a measure used in statistics indicating the value below
# which a given percentage of observations in a group of observations fall.
a = np.array([[30,40,70],[80,20,10],[50,90,60]])
a
```

Out[2]:

```
array([[30, 40, 70],
       [80, 20, 10],
       [50, 90, 60]])
```

In [3]:

```
np.percentile(a,50)
```

Out[3]:

```
50.0
```

In [4]:

```
np.percentile(a,50, axis = 1)
```

Out[4]:

```
array([40., 20., 60.])
```

In [5]:

```
np.percentile(a,50, axis = 0)
```

Out[5]:

```
array([50., 40., 60.])
```

In [ ]:

```

```

In [6]:

```
# numpy.median()
# Median is defined as the value separating the higher half of a data sample from the lower
a = np.array([[30,65,70],[80,95,10],[50,90,60]])
a
```

Out[6]:

```
array([[30, 65, 70],
       [80, 95, 10],
       [50, 90, 60]])
```

In [7]:

```
np.median(a)
```

Out[7]:

```
65.0
```

In [8]:

```
np.median(a, axis = 0)
```

Out[8]:

```
array([50., 90., 60.])
```

In [9]:

```
np.median(a, axis = 1)
```

Out[9]:

```
array([65., 80., 60.])
```

In [ ]:

```

```

In [ ]:

```
# numpy.mean()
# Arithmetic mean is the sum of elements along an axis divided by the number of elements.
# The numpy.mean() function returns the arithmetic mean of elements in the array.
```

In [10]:

```
a = np.array([[1,2,3],[3,4,5],[4,5,6]])
```

In [11]:

```
a
```

Out[11]:

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

In [12]:

```
np.mean(a)
```

Out[12]:

```
3.6666666666666665
```

In [14]:

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

Out[14]:

```
array([2.66666667, 3.66666667, 4.66666667])
```

In [15]:

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

Out[15]:

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

In [ ]:

```

```

In [16]:

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

Out[16]:

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

In [17]:

```
np.average(a)
```

Out[17]:

```
2.5
```

In [ ]:

```

```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: