NUMPY ARRAY

In [1]:

```
import numpy as np
```

In [6]:

```
a = np.array([1,3,5,7,9,11])
print(a)

a = np.arange(1,12,2)
print(a)

a = np.linspace(5,8,13)
print(a)

a = np.zeros((4,2))
print(a)

a = np.ones((2,3), dtype=np.int16)
print(a)

a = np.full((6),88)
print(a)

a = np.fromstring('25 30 35 40',dtype=np.int, sep=' ')
print(a)

a = np.array([[1,3,5],[7,9,11]])
print(a)

b = np.zeros_like(a)
print(b)
```

```
[ 1  3  5  7  9 11]
[ 1  3  5  7  9 11]
[5.   5.25 5.5  5.75 6.   6.25 6.5  6.75 7.   7.25 7.5  7.75 8.  ]
[[0. 0.]
 [0. 0.]
 [0. 0.]
 [0. 0.]]
[[1 1 1]
 [1 1 1]]
[88 88 88 88 88 88]
[25 30 35 40]
[[ 1  3  5]
 [ 7  9 11]]
[[0 0 0]
 [0 0 0]]
```

Numpy array attributes

In [9]:

```
print(a.size)
print(a.shape)
print(a.ndim)
print(a.itemsize)
print(a.dtype)
print(a.nbytes)
```

```
6
(2, 3)
2
4
int32
24
```

Indexing and slicing

In [12]:

```
print(a)
print(a[1])
print(a[0][2])
print(b[2:4])

print(a[:1])
print(a[1:3:2])
print(a[:, 1:2])
```

```
[[ 1  3  5]
 [ 7  9 11]]
[ 7  9 11]
5
[]
[[1 3 5]]
[[ 7  9 11]]
[[3]
 [9]]
```

array manupulation

In [14]:

```
c = np.arange(-9,-3,).reshape(2,3)
print(c)

c = c.swapaxes(0,1)
print(c)

c = c.flatten()
print(c)
```

```
[[-9 -8 -7]
 [-6 -5 -4]]
[[-9 -6]
 [-8 -5]
 [-7 -4]]
[-9 -6 -8 -5 -7 -4]
```

Use dtype to save space

In [16]:

```
d = np.arange(0,100)
print(d.dtype,type(d[1]))
print(d.nbytes)

d = np.arange(0,100, dtype='int8')
print(d.dtype,type(d[1]))
print(d.nbytes)
```

```
int32 <class 'numpy.int32'>
400
int8 <class 'numpy.int8'>
100
```

upcasting,rounding,Print formatting

In [19]:

```
e=np.array([(1.566666,2,3),(4,5,6)])
print(e.dtype)
```

```
print(e.dtype)

e=e.round(4)
print(e)

np.set_printoptions(precision=2, suppress=True)
print(e)
```

```
float64
[[1.5667 2.     3.    ]
 [4.     5.     6.    ]]
[[1.57 2.   3.  ]
 [4.   5.   6.  ]]
```

plotting

this is a plot of the *sin* function

$$f(x) = \frac{\sin(x)}{x}$$