

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

```
In [3]: s = 10
v = [1,2,3]
m = np.array([[1,2,3], [4,5,6]])

print(s, type(s))
print(v, type(v))
print(m, type(m))

10 <class 'int'>
[1, 2, 3] <class 'list'>
[[1 2 3]
 [4 5 6]] <class 'numpy.ndarray'>
```

## Vector 의 크기 계산

```
In [4]: import math
import numpy as np
```

벡터의 크기는 원소 각각의 제곱을 한 이후에 이를 더하고 이에 대한 제곱근 구하기

```
In [5]: x = np.array([2,2])
mag = lambda x:math.sqrt(sum(val**2 for val in x))
print(mag(x))
```

2.8284271247461903

## 벡터의 연산 ( + )

```
In [6]: import math
import numpy as np

d = np.array([4,5])
e = np.array([3,8])

f = d + e
print(f)
```

```
[ 7 13]
```

## 벡터의 연산 ( - )

```
In [7]: import math
import numpy as np

d = np.array([1,2])
e = np.array([2,1])
g = d - e
print(g)
```

```
[-1  1]
```

## 벡터의 스칼라 곱

```
In [8]: import math
import numpy as np

d = np.array([1,2])
i = 3 * d
print(i)
```

```
[3 6]
```

## multiply 함수 : 곱셈

```
In [9]: import numpy as np
```

```
In [10]: l = [1,2,3,4]
a = np.array(l, np.int)
li = a * 2
b = np.array(li, np.int)

print(a)
print(b)
print(np.multiply(a,b))
```

```
[1 2 3 4]
[2 4 6 8]
[ 2  8 18 32]
```

## 내적(dot) 구하기

```
In [11]: import math
import numpy as np
```

```
In [12]: d = np.array([4,5])
e = np.array([3,8])

j = d * e
print(j)
print(np.dot(d,e))
```

```
[12 40]
52
```

## 1차원 배열간의 dot연산

```
In [13]: import numpy as np  
a = np.array([1,2,3,4])  
b = np.array([2,4,6,8])  
print(a.dot(b))
```

60

In [ ]: