

```
In [ ]: # 파이썬 참고 사이트.  
        # 벡터, 행렬연산.  
        # 연립방정식 풀이, 가우스 소거법.  
  
        '''  
        www.python.org  
        www.scipy.org  
        www.numpy.org  
        '''
```

```
In [4]: def func(x,y):  
        return x + y
```

```
In [2]: # Lambda statement  
a = lambda x, y : x + y
```

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

```
Out[3]: 3
```

```
In [5]: func(1,2)
```

```
Out[5]: 3
```

```
In [7]: import numpy as np  
import matplotlib.pyplot as plt  
import scipy  
from numpy import sin, cos, exp
```

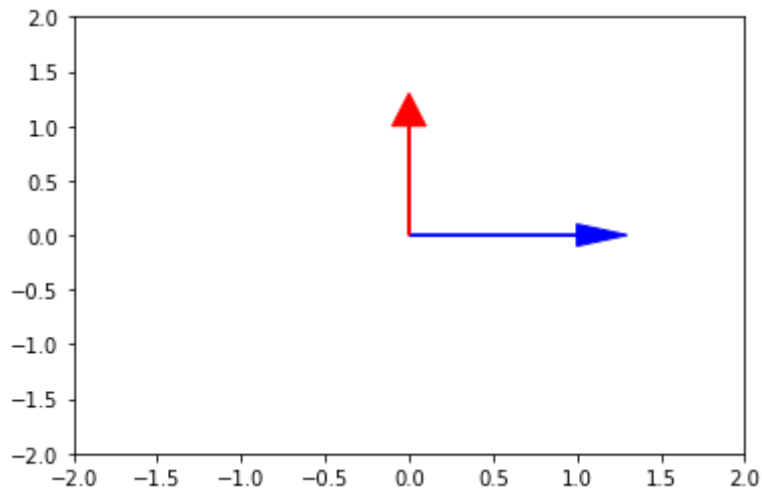
```
In [8]: sin(np.pi/2)
```

```
Out[8]: 1.0
```

```
In [12]: ex = np.array([1,0])
ey = np.array([0,1])

plt.arrow(0,0,ex[0],ex[1],head_width=0.2,color='b')
plt.arrow(0,0,ey[0],ey[1],head_width=0.2,color='r')
plt.xlim(-2,2)
plt.ylim(-2,2)
```

Out[12]: (-2, 2)



```
In [ ]: # 연산자 A 에 alpha 라는 매개변수를 바꾸고
# 단위 벡터 (0,1)에 연산을 하여 일직선을 생성함시다
# alpha = 0 -> (0,1)
# alpha = 1 -> (1,1)
# alpha = 2 -> (2,1)
# 그리고 이 벡터를 arrow command 를 써서 그립니다.
```

```

In [19]: # A = [[1, alpha],
#           [0, alpha]]
# 이 연산자를 u = [1 1] 에 적용했을때 생성되는 벡터들을 그려라

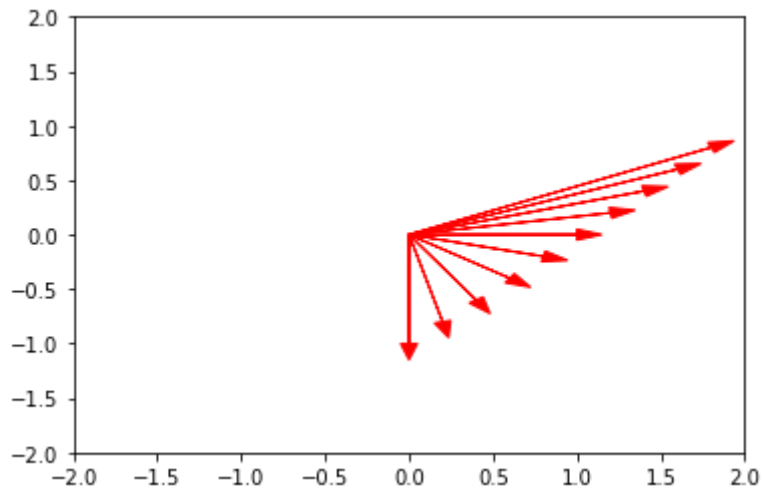
u = np.array([1,1])
for alpha in np.arange(-1,1,0.2):
    A = np.array([[1,alpha],
                  [0,alpha]])

    v = np.dot(A,u)

    plt.arrow(0,0,v[0],v[1],head_width=0.1,color='r')

plt.xlim(-2,2)
plt.ylim(-2,2)
plt.show()

```



```

In [ ]: # A*xvec = bvec
# Linear operator A*(a*xvec + b*yvec) = a*A*xvec + b*A*yvec

```

```

In [14]: A = np.array([[1,1],
                       [0,1]])

```

```

In [15]: np.dot(A,ex)

```

```

Out[15]: array([1, 0])

```

```

In [16]: np.dot(A,ey)

```

```

Out[16]: array([1, 1])

```

```
In [ ]: # A = np.array([[1,2],
#                  [0,3]])
# b = np.array([5,4])

# A*x = b 를 성립하는 x 를 찾으시오.
# 가) 역행렬   linalg.inv
# 나)         linalg.solve 를 사용하시오
# 다) 소거법을 사용하시오

# A-1 * A = 1
```

```
In [22]: A = np.array([[1,2],[0,3]])
Ainv = np.linalg.inv(A)
b = np.array([5,4])
```

```
In [21]: Ainv
```







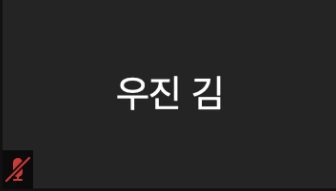
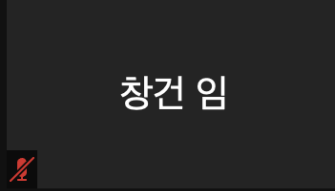
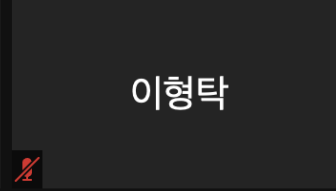

```
Out[21]: array([[ 1.          , -0.66666667],
               [ 0.          ,  0.33333333]])
```

```
In [23]: # A*x = b -> Ainv*A*x = x = Ainv*b
np.dot(Ainv,b)
```

```
Out[23]: array([2.33333333, 1.33333333])
```

```
In [25]: xsol = np.linalg.solve(A,b)
print('Sol ',xsol)
```

```
Sol  [2.33333333 1.33333333]
```

 김영찬	 George	 임재희	 서단우
 김규민	 상민 조	 박재욱	 2017550030 허우현
 박수연	 허지원	 김연주	 남상우
 신상구	 우성균	 형준 임	 윤종석
 배준영2	 창건 임	 정재일	 배준영1
 우진 김	 창건 임	 이형탁	 상민 조

```
In [ ]: # 노캠: 김우진, 임창건, 이형탁
```

행인 A와 B가 1차원 선상에서 등가속운동을 하고있다. A와 B는 서로 100m 떨어져 있으며 속력  $v_0 = 0$  m/s로 출발한다. A는 가속도  $a_1$ , B는 가속도  $a_2$ 로 걷는다고 가정하자. 두 사람이 반대 방향으로 걸을 때는 두 사람이 만날 때까지 30초가 걸리며, 두 사람이 같은 방향으로 걸을 때는 90초가 걸린다.  $a_1$  과  $a_2$ 를 구하는 script를 만들어라.

```
In [ ]: # 연립방정식 풀이
# A*x = b

# a11*x1 + a12*x2 = b1
# a21*x1 + a22*x2 = b2

# a11 + a12
# a21 + a22
```

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

```
In [ ]: # a11*a1 + a12*a2 = b1
# a21*a1 + a22*a2 = b2
```

```
In [27]: #1)
#x_A = a1*t^2/2
#x_B = 100 - a2*t^2/2

#x_A = x_B -> x1*t^2/2 = 100 - x2*t^2/2 # t_1 = 30
# x1*t^2/2 = 100 + x2*t^2/2 # t_2 = 90

# x1*t_1^2/2 + x2*t_1^2/2 = 100
# x1*t_2^2/2 - x2*t_2^2/2 = 100

t1 = 30
t2 = 90

A = np.array([ [(t1**2)/2, t1**2/2],
                [(t2**2)/2, -t2**2/2] ])

b = np.array([100,100])

x = np.linalg.solve(A,b) #
print('x ',x)

x [0.12345679 0.09876543]
```

$$u = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, v = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \text{ 이다.}$$

$$Au = \begin{pmatrix} 3 \\ 2 \end{pmatrix} \text{ 이고, } Av = \begin{pmatrix} -1 \\ -2 \end{pmatrix} \text{ 일 때 } A \text{를 찾아라.}$$

```
In [ ]: # A와 B행인이 등속 운동을 하고 있다. 서로의 초기 거리는 100미터이다.  
        # 서로 반대 방향으로 걸을때 걸리는 시간은 5초  
        # 서로 같은 방향으로 걸을때 걸리는 시간은 15초  
        # A, B 속도를 찾으시오.
```

```
In [ ]:
```