CHAPTER

04

NumPy 라이브러리

NumPy Library

컴퓨터소프트웨어공학과
김대영

파이썬 실습 2

```
import numpy as np
import matplotlib.pyplot as plt
from numpy.linalg import inv
import random as rd

def main():

    points = np.arange(-5, 5, 0.01)    # 1D array
    x, y = np.meshgrid(points, points) # 2D array :
tuple (x, y)

    z = np.sqrt(x**2 + y**2) # (x, y) = z ; z = sqrt(x^2
+ y^2)
    print(z)

    plt.imshow(z, cmap=plt.cm.gray) # imagination of z
    plt.colorbar()
    plt.title("Image plot of $\sqrt{x^2 + y^2}$")
    plt.show() # graph


    xarr = np.array([1.1, 1.2, 1.3, 1.4, 1.5])
    yarr = np.array([2.1, 2.2, 2.3, 2.4, 2.5])
    cond = np.array([True, False, True, True, False])

    result = np.where(cond, xarr, yarr)
    print(result)

    arr = np.random.randn(4,4)
    print(arr)
    result = np.where(arr>0, 2, -2)
    print(result)
```

```
    arr = np.random.rand(5,4) #row 5; column 4
    print(arr)
    ret = arr.mean()
    print(ret)
    ret = np.mean(arr) #<- arr.mean()
    print(ret)
    ret = arr.sum()
    print(ret)

    ret = arr.mean(axis=1) #accross the columns (i.e.,
row)
    print(ret)
    ret = arr.sum(axis=0) # down the row (i.e, column)
    print(ret)

    arr = np.array([0, 1, 2, 3, 4, 5, 6, 7])
    ret = arr.cumsum()
    print(ret)

    arr = np.array([[0, 1, 2], [3, 4, 5], [6, 7, 8]])
    ret = arr.cumsum(axis=0)    #sum
    print(ret)
    ret = arr.cumprod(axis=1)   #product
    print(ret)

    arr = np.random.randn(100)
    print(arr)
    ret = (arr>0).sum()
    print(ret)
```

파이썬 실습 2

```
    bools = np.array([False, False, True, False])
    ret = bools.any()
    print(ret)
    ret = bools.all()
    print(ret)

    arr = np.random.randn(6)
    print(arr)
    ret = np.sort(arr)
    print(ret)

    arr = np.random.randn(5,3) #5x3
    print(arr)
    ret = np.sort(arr)
    print(ret)

    names = np.array(['Bob', 'Joe', 'Will', 'Bob',
'Will', 'Joe', 'Joel'])
    set = np.unique(names)
    print(set)

    values = np.array([6, 0, 0, 3, 2, 5, 6])
    set = np.in1d(values, [2,3,6])
    print(set)

    arr = np.arange(10)
    print(arr)
    np.save("some_arr", arr)

    ld = np.load("some_arr.npy")
    print(ld)
```

```
arr = np.arange(10)
arr2=np.arange(5)
np.savez("arr_arch.npz", a=arr, b=arr2)
arch = np.load("arr_arch.npz")
print(arch['a'])
print(arch['b'])


x = np.array([[1, 2, 3], [4, 5, 6]])
print(x)
y = np.array([[6, 23], [-1, 7], [8, 9]])
print(y)
ret = x.dot(y)
print(ret)


x = np.array([[1,2,3],[4,5,6]])
y = np.array([[6,23],[-1,7],[8,9]])
ret = np.dot(x,y)   # x.dot(y)
print(ret)

x = np.array([[1,2,3],[4,5,6]])
ret = x @ np.ones(3) # x.dot(np.ones(3)); x.dot(3x1)
print(ret)

x = np.random.randn(3,3)
mat = x.T.dot(x)
print(mat)

imat = inv(mat) # from numpy.linalg import inv
print(imat)

dimat = mat.dot(imat)
print(dimat)
```

파이썬 실습 2

```
    samples = np.random.normal(size=(4,4)) #normal
distribution
    print(samples)

    # np.random.seed(1234) #global seed
    rng = np.random.RandomState(1234)   # local seed
    ret = rng.randn(10)
    print(ret)


    position = 0
    points = [position]
    steps = 1000

    for i in range(steps):
        if rd.randint(0,1): #import random
            step = 1
        else:
            step = -1
        position += step
        points.append(position)

    plt.plot(points[:100])
    plt.show()
```

```
    nsteps = 1000
    draws = np.random.randint(0,2,size=nsteps)
    steps = np.where(draws>0, 1, -1)
    points = steps.cumsum()

    plt.plot(points[:100])
    plt.show()

    ret = points.min()
    print(ret)
    ret = points.max()
    print(ret)
    val = np.abs(walk) >= 10
    ret = val.argmax()
    print(ret)


if __name__ == '__main__': # main()
    main()
```