part 9

numpy - 라이브러리 .import해서 불러와야함.

함수가 들어있는 어떤 라이브러리를 . 패키지.

numpy 라는 패키지 라이브러리 속에 또 작은게 들어있을수 있음, 그 밑에 패키지 이름 A, B 또 그안에 만들수도.

numpy 안에 내부적으로 함수가 존재할수 있음. A안에 패키지가 또있다면. 패키지. 내부적으로 안에 있는걸 부를때는 numby.A.D.어떤함수 . numpy가 젤 위에 있는 상위개념이고 그 안에 또 패키지... 점이 그런 용도로 쓰임.

import numpy

1) numpy-A-D-f를 쓰고 싶으면 numpy.A.D.f (너무 복잡)

>불러올 때 젤 큰걸 불러오는 방법.

2) from Numpy Import A (> numpy 에 있는 A를 불러오자)

A.D.f

from Numpy import A.D

>from 으로 import할 수도 있음

사진

[2]

import해서 들어오고 numpy를 줄여서 쓰고 싶을 때 as.

np 라고 줄여서 쓰고 싶을 때 as

matplotlib가 젤 위에 있고 그안에 포함되있는게 - pyplot가 속해있음. 포함되어 있는 subset을 plt로 받아온다.

쓸 수 있는 다른 방법중 틀린 것-시험

from mat import pyplot as plt

그 다음부턴 plt쓸수 있음

numpy는 왜 필요? list 하고 아주 비슷한데 수학적으로 계산도 할 수 있고 해서 처리할 모든 data는 다 np처리를 해야

[3]

np.empty

empty는 함수 (괄호로 input을 받음)

np라는 젤 큰 라이브러리 안에 empty .

list가 입력으로 들어감.

2 by 3 로 리스트로 .

가로 세로 행렬 : 직사각형의 숫자의 array

2행 3열 - 뚱뚱 옆으로

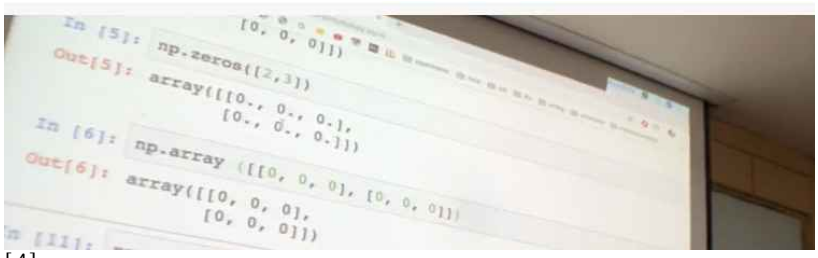
내부적으로 들어가는 것은 datatype을 .dtype을 int로 해서 하나 만들어라

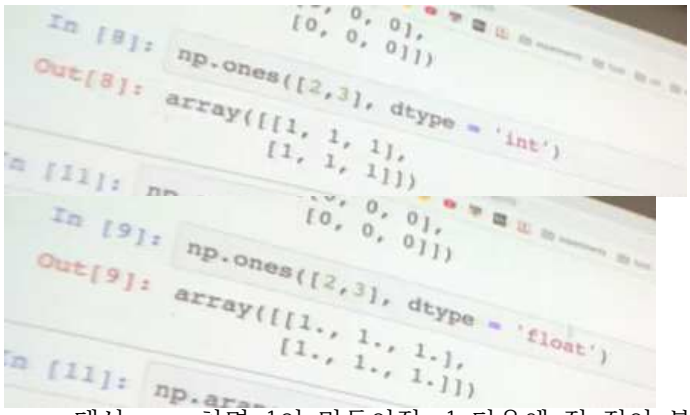
out[3]

numpy속에 들어있는 함수를 이용했기 때문에 숫자가 만들어지는데 계싼가능한 숫자

2by 3의 array로 되있음. - 3개짜리 list가 하나가 있고 그게 두줄이 있음.

int. 안에 든 숫자는 랜덤한 숫자. 소수점 숫자 아님.



[4]
np라이브러리 속에 있는 zeros라는 함수를 이용. 거기 입력에 list로 2곱하기 3을 넣음.
2by 3를 만드는데 0으로 채워진.
** [0,0,0]
>>그냥 0이 들어가있는 백타
이걸 2by 3로 만들고 싶을때는
** [[0,0,0] , [0,0,0]]
2행이 있고 3열이있는 리스트.
>>쓸수가 없음 계산안됨.
계산이 되는 array로 만드는 방법이
np.array ([[0,0,0] , [0,0,0]])
리스트에서 array로 바꿔줌
[4]와 같음.
np.array - list를 array 로 convert해줘라 >위에거랑 똑같은 결과


zeros대신 ones하면 1이 만들어짐. 1 다음에 점 점이 붙은건 int가 아니라 float
ones라는 함수가 디폴트로 datatype을 float로.
dtype= 'int' 해주면 점이 사라짐
float64 로해도 변화는 없음. 몇째짜리 까지 할까에 대한 정의
64 정교하고 싶을 때 - 숫자를 길게 하면 메모리를 많이 차지함. 1.00000...
정확도와 데이터 양 반비례.

```
In [5]: np.arange(5)
Out[5]: array([0, 1, 2, 3, 4])

In [6]: np.arange(0,10)
Out[6]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

In [7]: np.arange(0,10,2)
Out[7]: array([0, 2, 4, 6, 8])

In [8]: np.arange(0,10,2, dtype='float')
Out[8]: array([0., 2., 4., 6., 8.])
```

계산이 될 수 있는 array를 만들어줌. 숫자가 다섯 개. 5라고 쓰는순간 index 5개가 만들어짐.

arange (0,10) : 0부터 10까지 하면 10은 포함안되고 10 밑으로까지.

(0,10,2) : 2만큼 증가하면서 10미만까지

data type설정도 할수있음. float, float32, 64도 쓸수있음. - 얼마나 정확성을 높이느냐와 관련.

```
In [9]: np.linspace(0,10,6)
Out[9]: array([ 0.,  2.,  4.,  6.,  8., 10.])

In [10]: np.linspace(0,10,7)
Out[10]: array([ 0.        ,  1.66666667,  3.33333333,  5.        ,  6.66666667,
         8.33333333, 10.        ])

In [15]: X= np.array([4,5,6])
         X
Out[15]: array([4, 5, 6])

In [14]: X= np.array([[1,2,3],[4,5,6]])
         X
Out[14]: array([[1, 2, 3],
         [4, 5, 6]])

In [17]: X= np.array([[1,2],[3,4],[5,6]])
         X
Out[17]: array([[1, 2],
         [3, 4],
         [5, 6]])

In [18]: X= np.array([[[1,2],[3,4],[5,6]],[[1,2],[3,4],[5,6]]])
         X
Out[18]: array([[[1, 2],
         [3, 4],
         [5, 6]],

        [[1, 2],
         [3, 4],
         [5, 6]]])

In [20]: X. ndim
Out[20]: 3
```

[9]

linear의 준말

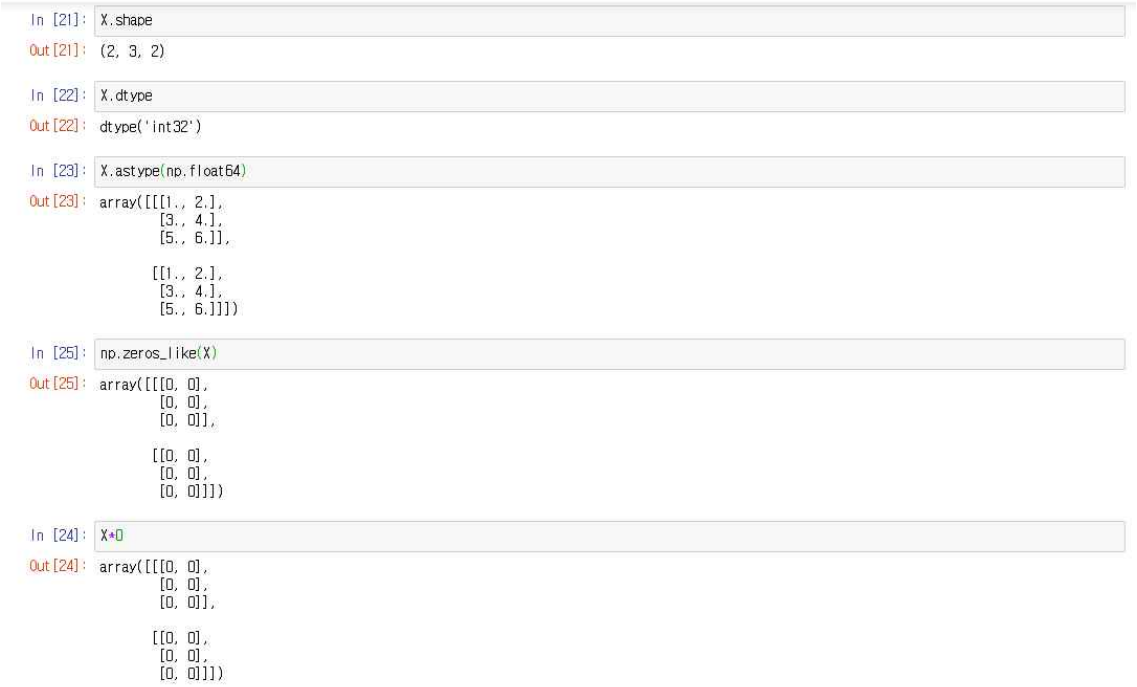
linspace(0,10,6) : 0부터 10까지 0, 10포함. 그걸 총 6개로 똑같이 나누어준다.

차이가 다 똑같음. 첫째에서 둘째로 가는

[11]

list에서 array 로 바꾸는거

2by3를 만들고 싶을 때.
3by 2 만들고 싶을 때.
*백터면 1차원. 직사각형 2차원. 직육면체로 되면 3차원. 2차원 행렬 두 개 있으면 3차원
대괄호 두 개 2차원. 대괄호 하나 더 쓰고 컴마 >3차원



x. shape
(2,3,2)
2곱하기 3곱하기 2
-3개니까 3차원인데 3곱하기 2가 2차원짜리 직사각형. 이게 두 개가 있다. 젤 큰 괄호속에있는게 두 개. (가 첫 번째 나오는 2) 그다움에 3개의 리스트 안에 2개의 숫자.
X. dtype
: x속에 나오는 숫자가 어떤 type인가 >int
타입을 바꾸고 싶을 때 astype 이라는 function .
np.float64로 바꿔줘라

모든 형태를 유지한채 숫자를 0으로 바꿔주라. : zeros like
X*0 해도 계산이 가능하니 똑같이 만들어짐.

<11/5>
numpy - list에 숫자 담는것보다
np. array [] 하면 np형태의 데이터로 바뀐다.

pure tone -
sinusoidal phasor
싸인하고 코사인에 들어가는 입력 - sin( ) degrees 가 들어가면 안되고 radians 가 들어가야함.
sin(π/4)

radians
Phasor
sampling rate : 1초에 총 만개의 숫자로 표현.
#

<11/7>

젤큰 라이브러리 밑에 sub라이브러리.
#parameter setting
time 은

0부터 2 파이까지 만들어짐.