numpy

먼저 파이썬이 설치된 경로에 numpy 라이브러리를 설치해야합니다.

pip3 install numpy

그래프를 그릴 경우에는

pip3 install matplotlib

선언

```
import numpy as np
```

numpy array생성시

```
data1 = [6,7,8.2,0.1]
arr1 = np.array(data1)
arr1.shape #생성한 array의 크기를 보여줌 (열,행)으로 출력
#랜덤성분을 가지는 원하는 크기의 배열 생성
np.random.randn(n,m) #n by m size
```

초기화된 배열 생성

```
np.zeros((행,열))#0으로 초기화된 원하는 사이즈의 행렬 생성가능
np.ones((행,열))#데이터를 1개만 넣을 경우 열만 입력한 것으로 처리함.
```

dtype

array의 고유한 데이터형 확인

```
arr1.dtype
#데이터 생성시 직접 지정도 가능하다.
arr = np.array(data1,dtype = np.int64)
#데이터형 변환
floatArr = arr.astype(np.float64)
'''
데이터형                            | 데이터형 코드    | 설명
int8,int16,int32,int64              i1,i2,i4,i8    부호가 있는 8,16,32,64비트 정수
uint8,uint16,uint32,uint64          u1,u2,u4,u8    부호가 없는 8,16,32,64비트 정수
float16,float32,float64,float128    f2,f4,f8,f16   16,32,64,128비트 실수
```

```
complex64,complex128,complex256   c8,c16,c32    64,128,256비트 복소수
bool                              b
object                            O
string_                           S
unicode_                          U
'''
```

연산

+,-,*,/ - 각 성분들끼리의 연산

sqrt()와 같은 함수에 arr를 넘겨주면 각 성분에대한 연산 실행

indexing

성분 추출시 사용

1차원 python indexing과 slicing과 같음.

2차원[행 , 열]

행과 열의 index는 구간으로 설정가능

ex)

```
arr = [[1,2,3,4,5],
       [2,4,6,8,10]]
arr[:,:3]#모든 행에대한 0~2열까지의 데이터
```

boolean indexing

데이터 마스크를 생성하고 이용할 때 사용

True인 성분이 있는 위치의 데이터만 뽑아낼 때 사용

```
#이번 프로젝트의 예시를 들어서 설명하겠습니다.
#무언가를 감지한 센서들을 1, 그렇지않은 센서들을 0
sensor = [0,0,0,1,1]
data = [[1,2,3,4,5],
        [2,4,6,8,10],
        [3,6,9,12,15],
        [4,8,12,16,20],
        [5,10,15,20,25]]
data[(sensor == 1),:]
#그 외에도 데이터 array의 성분에 조건문을 달아서 마스크 생성도 가능
data[data[:,1:3]<10,:]
```

sort,sum등의 함수들도 이용 가능

np.unique함수를 이용하면 중복된 성분 제외한 배열 반환

데이터 읽어오기

```
np.loadtxt("파일경로",파일에서 사용한 구분자,데이터타입)
np.savetxt("파일명",해당 array변수,파이썬포맷,구분자)
```

//Todo

2020/11/17 이전에 업로드 예정

pandas 관련 내용.

시계열 데이터 생성에 관한 내용

데이터 모델링 관련 내용