

# 1. Python numpy

## 1. 1 numpy 생성

- zeros(array) 생성

① np.array()

② np.arange()

np.linspace() → 같은 간격

③ np.zeros(), np.ones(), np.eye()

- 데이터형(dtype) 설정

num - arr = arr - arr.astype(dtype)

④ np.random.rand([x,y,z,...])

np.random.randint([low]high [size])

## 1. 2 인덱싱과 슬라이스

① 배열[행렬][열][원소: 원소]

② 배열[행:행]

③ 배열[행행: 행행, 열열: 열열]

## 1. 3 numpy 특징

速度快 → 다른 프로그래밍 언어에 비해 속도가 빠름.

① 두 배열을 학습하는 데 있어 예상하는 차원=1

② 두 배열의 차원이 다른 경우 예외 처리

## 2. Python pandas

### 2.1 Python 콜렉션 자료형을 위한 pandas

- `Series` : 2차원 테이블을 갖는 1차원 데이터  
`Dataframe` : 2차원 테이블을 갖는 2차원 데이터
- pandas는 데이터의 원본은 모두 DataFrame인지 알기 어렵다.  
numpy는 데이터의 원본은 모두 ndarray인지 알기 어렵다.
- `pd.date_range(start='1984', end='1985', periods=125, freq='D')`  
→ end와 periods는 단위가 날짜 단위이다.
- `df = pd.DataFrame(data[, index= , columns= ])`  
→ data: dict, list, array, Series, DataFrame
- `index: axis=0 , columns: axis=1`
- `df.index , df.values , df.columns`은 주제
- numpy처럼 NaN pandas 콜렉션에서는 모두 `None`으로 표기된다. None  
→ index, column이 일치하는 행과 열을 제거한 예상.

- df.mean() → axis=0 (1차원 데이터 평균)  
axis=1 (2차원 데이터 행평균)

### • 행평균

df[시작:끝] 끝-1 : 평균 사용

df.loc[시작:'끝'] 끝-1 : ~~인덱스 사용~~ 사용

### • 열평균

df['열이름']

df[['열이름']] [행위치 : 행위치] 끝-1  
시작 끝

df[['열이름']] [행이름 : 행이름] 끝-1  
시작 끝

- loc[행번호, 열번호] : 원하는 위치 가능.  
행번호 열번호  
인덱스 인덱스  
컴퓨터상의  
(이름과 소문자 X)

### • DataFrame 연산(통합)

① 결합 : df1.append(df2, ignore\_index=True)  
(결합 columnnum)

② 합집합 : df.join(df2)  
(결합 indexnum) → 결합 한데도 NaN

③ 통합연산 : dfl.merge(dfr, how='left', on='id')

left  
right  
inner  
outer  
cross

## 2-2 pandas의 CSV 파일 읽고 쓰기

- df = pd.read\_csv ('인수', index\_col = , sep = , dtype = )
- df.to\_csv ('출력')