**교육일지**

**교육 제목 : 산술연산, 데이터 입출력**

**교육 장소 : YGL C6 강의실**

**교육 일시 : 2021/09/29**

**오전 교육**

**행 인덱스 재배열**

**import pandas as pd # reindex = index를 재배치**

**# 딕셔서리를 정의**

**dict\_data = {'c0':[1,2,3], 'c1':[4,5,6], 'c2':[7,8,9], 'c3':[10,11,12], 'c4':[13,14,15]}**

**# 딕셔서리를 데이터프레임으로 변환. 인덱스를 [r0, r1, r2]로 지정**

**df = pd.DataFrame(dict\_data, index=['r0', 'r1', 'r2'])**

**print(df)**

**print('\n')**

**#index 초기화 : df.reset\_index**

**print(df.reset\_index())**

**df = pd.DataFrame(dict\_data, index=['r0','r1','r2'])**

**print(df)**

**# sort : 인덱스로 정렬, 특정 칼럼의 값으로정렬**

**print(df.sort\_index(ascending=False)) #index로 정렬 asxending+False 역으로 정렬**

**df.sort\_values(by=['c0','c1'], ascending=False)**

**# 인덱스를 [r0,r1,r2,r3,r4]로 재지정**

**new\_index = ['r0', 'r1', 'r2', 'r3', 'r4'] #추가되는 데이터는 NaN로 입력됨**

**ndf = df.reindex(new\_index)**

**#ndf = df.reindex(new\_index, fill\_value=0) #추가되는데이터에 초기값설정**

**#reindex로 발생한 NaN값을 숫자 0으로 채우기**

산술연산

import pandas as pd

#딕셔너리 데이터로 판다스 시리즈 만들기

student1 = pd.Series({'국어':100, '영어':80, "수학":90})

student2 = pd.Series({'국어':90, '영어':85, "수학":95, '과학':100})

print(student1)

print('\n')

print(student1/200, "\n")

print(student1 + student2) # series 연산자(+,\*,/,-) series 또는 숫자

# 내부적으로 인덱스를 정렬한 후 각각의 인덱스로 계산함

student1.add(student2, fill\_value=0) # add, sub, mul,div(자료가 없으면 원하는 값으로 대체)

# 연산식에 NaN이 포함되면 결과는 NaN으로 나타남 -> 옵션으로 fill\_value 원하는값

**오후교육**

**import pandas as pd**

**file\_path = "./read\_csv\_sample.csv"**

**df = pd.read\_csv(file\_path) #read\_csv 컬럼 데이터가 ','로 구분된 자료를 읽어옴**

**print(df)**

**df1 = pd.read\_csv(file\_path,header=None,index\_col=None)**

**print("df1 === ")**

**print(df1)**

**df2 = pd.read\_csv(file\_path,index\_col='c0') #index 컬럼을 지정**

**print("df2 === ")**

**print(df2)**

**read\_csv(옵션,... )**

**path : 파일의 위치 포함한 파일명**

**sep : 필드를 구분하는 구분자 ','**

**header : 헤더가 정의되어 있는지 None**

**index\_col : 인덱스로 사용될 컬럼명, None 인덱스 없음**

**name : 컬럼 이름으로 사용 될 문자열 리스트**

**skiprows : 처음 행 부터 skip하고자 하는 행 수**

**skipfooter : 마지막 행 부터 skip하고자 하는 행 수**

**encoding : 텍스트 인코딩 종류를 지정 'utf-8'**

**파일 처리하는 방법**

**읽어오기**

**- read\_scv, read\_html, read\_excel, read\_json**

**저장하기**

**- df\_csv(파일명), df.to\_html, df.to\_excel, df.to\_json**

**json 파일은 구조가 DataFame과 동일**

**csv 파일은 2차원 배열로 -> DataFame으로 변경해서사용**

**df= pandas.DataFame ...**

**행 df.loc[인덱스명], df.iloc[인덱스 첨자] 여러개[start:end],[,..,]**

**열 df.열명 df[열명]**

**원소 df.loc[[인덱스명],[컬럼명]]**

**df.iloc[[인덱스명],[컬럼명]]**