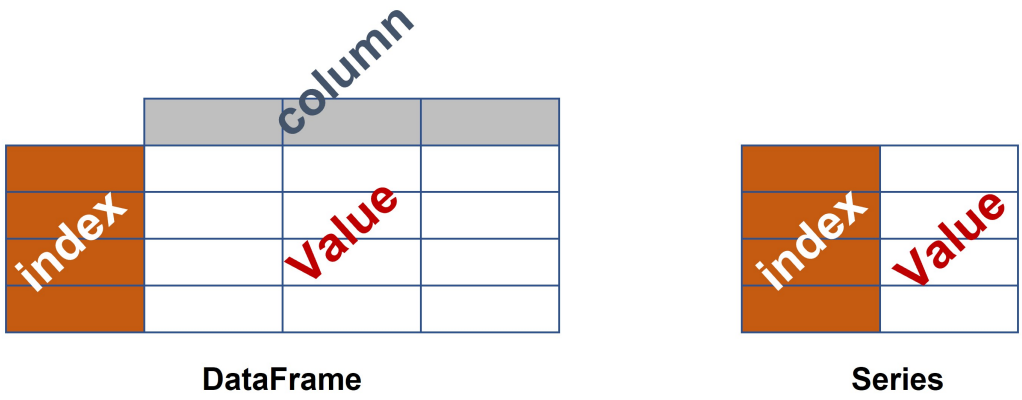
Data Frame

- Pandas는 라벨이 붙어있는 2차원 배열(DataFrame)이나 1차원 배열(Series)을 다루는 도구
- 2차원 배열(DataFrame)의 행에 붙어 있는 라벨을 index(인덱스), 열에 붙어 있는 라벨을 column(컬럼)이라 부름
- 1차원 배열에는 행에만 라벨이 붙어 있고, 이를 index(인덱스)라고 부름



Lab 1 : 시험 성적 데이터로 데이터 프레임 만들기

- import pandas : Pandas를 사용하기 위해서는 pandas 패키지를 로드해야 함
- 인덱스가 [0, 1, 2, 3]이고, 컬럼이 ['name', 'english', 'math']인 데이터 프레임을 생성해 봄
- 생성한 데이터 프레임의 이름은 df를 사용
- 데이터 프레임 생성시 [enter]로 줄바꿈 가능
- 별도로 인덱스를 지정하지 않으면 인덱스는 숫자 0부터 부여됨

In [ ]:
```
# Pandas를 사용하기 위해서는 pandas 패키지를 로드
import pandas as pd
```

In [ ]:
```
# Data Frame을 새로 생성(name, english, math의 3개 컬럼(변수)으로 구성)
df1 = pd.DataFrame({'name'    : ['김기훈', '이유진', '박동현', '김민지'],
                    'english' : [90, 80, 60, 70],
                    'math'    : [50, 60, 100, 20]})

# 생성한 Data Frame df1을 출력
df1
```

> 데이터 프레임으로 분석하기

In [ ]:
```
# Data Frame df1의 index 값 확인
df1.index
```

In [ ]:
```
# Data Frame df1의 컬럼(변수) 값 확인
df1.columns
```

In [ ]:
```
# Data Frame df1의 값을 확인 (numpy의 array)로 출력
df1.values
```

In [ ]:
```
# 컬럼 math 추출 (시리즈)
s1 = df1['math']
s1
```

In [ ]:
```
# df1에서 컬럼 math의 합 계산
sum(df1['math'])
```

In [ ]:
```
# df1에서 컬럼 english의 평균 값 계산
sum(df1['english'])/4
```

In [ ]:
```
# df1에서 컬럼 math의 합 계산
df1['math'].sum()
```

In [ ]:
```
# df1에서 컬럼 math의 평균 값 계산
df1['math'].sum()/4
```

In [ ]:
```
# df1에서 컬럼 math의 평균 값 계산
df1['math'].sum()/df1['math'].count()
```

In [ ]:
```
# df1에서 컬럼 math의 평균 값 계산
df1['math'].mean()
```

In [ ]:
```
# 컬럼 math 추출 (Data Frame)
s1 = df1[['math']]
s1
```

In [ ]:
```
# df1에서 컬럼 english, math 추출 (Data Frame)
df1[ ['english', 'math'] ]
```

In [ ]:
```
df1[['english', 'math']] > 70
```

In [ ]:
```
# Data Frame pd_em에서 컬럼(english, math)별 합 계산
# Python에서는 True는 1, False는 0
pd_em = df1[['english', 'math']] > 70
pd_em.sum()
```

Lab 2 : 데이터 프레임 만들기

- 이름을 index로 만들기

In [ ]:
```
# Data Frame 생성(컬럼 이름과 인덱스 이름을 명시적으로 지정)
df2 = pd.DataFrame([[90, 50], [80, 60], [60, 100], [70, 20]],
                   columns = ['english', 'math'],
                   index   = ['김기훈', '이유진', '박동현', '김민지'])
df2
```

In [ ]:
```
# df2의 index 값 출력
df2.index
```

In [ ]:
```
# df2의 컬럼 값 출력
df2.columns
```

In [ ]:
```
# df2의 데이터 값 출력(array)
df2.values
```

In [ ]:
```
# df2의 컬럼 math의 평균 구하기
df2['math'].mean()
```

In [ ]:
```
# df2의 컬럼별 값이 50이상 여부 확인 (True, False)
df2 > 50
```

In [ ]:
```
# True : 1, False : 0
(df2 > 50).sum()
```

Lab 3 실습 - 교재 84 문제 풀이

- 교재 84쪽의 표를 이용하여 데이터 프레임 df2를 만들어 출력해 보세요

df1	제품	가격	판매량
0	사과	1800	24
1	딸기	1500	38
2	수박	3000	13

df2	가격	판매량
사과	1800	24
딸기	1500	38
수박	3000	13

실습 1 : df1

- 교재 84쪽의 표를 이용하여 데이터 프레임 df1을 만들어 출력해 보세요
- df1의 인덱스, 컬럼, value 값을 추출하여 출력해 보세요
- 만든 데이터 프레임을 이용하여 과일의 평균 가격과 판매량의 평균을 구하시오

In [ ]:
```
#
#
```

실습 2 : df2

- 교재 84쪽의 표를 이용하여 데이터 프레임 df2를 만들어 출력해 보세요
- df2의 인덱스, 컬럼, value 값을 추출하여 출력해 보세요
- 만든 데이터 프레임을 이용하여 과일의 평균 가격과 판매량의 평균을 구하시오

In [ ]:
```
#
#
```

Lab 4 : 외부 데이터 이용하기

- 분석할 데이터를 엑셀 데이터에서 가져 오기 (https://bit.ly/doit_python)
- pd.read_excel('excel_exam.xlsx')
- 첫번째 행을 컬럼으로 지정하고, 인덱스는 0부터 할당됨
- 특정 컬럼을 인덱스로 만들기 위해서는 set_index() 함수 사용

In [ ]:
```
# excel 파일 'excel_exam.xlsx'를 읽어들여 Data Frame df_exam으로 저장
df_exam = pd.read_excel('excel_exam.xlsx')
```

In [ ]:
```
# Data Frame df_exam의 첫 5개 행을 출력
df_exam.head()
```

In [ ]:
```
# excel 파일 'excel_exam.xlsx'를 읽어들여 Data Frame df_exam으로 저장
# 단, Data Frame의 index를 'id' 값으로 지정
df_exam = pd.read_excel('excel_exam.xlsx').set_index('id')
```

In [ ]:
```
df_exam.head()
```

> 데이터 분석하기

In [ ]:
```
# 컬럼 math의 합 계산
df_exam['math'].sum()
```

In [ ]:
```
# df_exam의 행의 개수 확인
len(df_exam)
```

In [ ]:
```
# df_exam의 컬럼별 합 계산
df_exam.sum()
```

In [ ]:
```
# df_exam의 컬럼별 평균 계산
df_exam.mean()
```

Lab 5 엑셀의 첫번째 행이 변수명(컬럼 이름)이 아닌 경우

- 첫번째 행의 데이터를 컬럼 이름으로 인식함
- Lab4에서 사용한 엑셀 파일에서 첫 행을 삭제하여 실습에 사용

In [ ]:
```
# 엑셀 파일의 첫번째 행이 컬럼명이 아님
pd_excel_exam_novar = pd.read_excel('excel_exam_novar.xlsx')
pd_excel_exam_novar.head()
```

In [ ]:
```
# 엑셀 파일의 첫번째 행이 컬럼명이 아니므로, 컬럼명으로 사용하지 않음
pd_excel_exam_novar = pd.read_excel('excel_exam_novar.xlsx', header = None)
pd_excel_exam_novar.head()
```

In [ ]:
```
# 데이터 프레임 pd_excel_exam_novar의 컬럼명 확인
pd_excel_exam_novar.columns
```

In [ ]:
```
# 데이터 프레임 pd_excel_exam_novar의 컬럼명을 별도로 지정함
pd_excel_exam_novar.columns = ['id', 'nclass', 'math', 'english', 'science']
pd_excel_exam_novar.head()
```

In [ ]:
```
# 데이터 프레임 pd_excel_exam_novar의 index를 컬럼 id로 지정함
pd_excel_exam_novar.set_index('id')
```

Lab 6 : CSV 파일 불러오기

- Lab 4에서 사용한 엑셀 파일을 csv 유형 파일로 저장
- csv 파일 : 값이 쉼표(,)로 구분되어 저장된 자료 형태

In [ ]:
```
import pandas as pd
df_exam_csv  = pd.read_csv('excel_exam_csv.csv')
```

In [ ]:
```
df_exam_csv.head()
```

실습

- df_exam에서 math, english, science의 성적인 50점 이상인 학생의 수를 구하시오

In [ ]:
```
#
#
```