

데이터 분석을 위한 PANDAS

ROW·COL ELEMENTS



인덱스 & 열이름

◆ 인덱스 & 열이름 변경

■ 부분 변경

→ inplace 매개변수 : 원본 데이터 적용 여부 설정

```
DataFrame/Series.rename( index={ 기존인덱스: 새 인덱스, ..... }, inplace )
```

```
DataFrame/Series.rename( columns={ 기존컬럼: 새컬럼, ..... }, inplace)
```

◆ 인덱스 & 열이름 변경

행인덱스, 열이름 변경

```
# 열 이름 중, '나이'를 '연령', '성별'을 '남녀', '학교'를 '소속'으로
df2=df.rename(columns={'나이':'연령', '성별':'남녀', '학교':'소속'})

# df의 행 인덱스 중에서, '준서'를 '학생1', '예은'을 '학생2'로 바꾸기
df3=df.rename(index={'준서':'학생1', '예은':'학생2'}, inplace=True)

# df 출력(변경 후)
print(f"df====={df3}\n{df}")
print(f"df2=====\n{df2}")
```

◆ 축 이름 변경

- inplace 매개변수 : 원본 데이터 적용 여부 설정

```
DataFrame/Series.rename_axis( index= '축이름' , inplace )
```

```
DataFrame.rename_axis( columns= '열이름' , inplace )
```

◆ 축 이름 변경

Series 축 이름 변경

```
# 시리즈 인스턴스 생성
```

```
dataSR2 = pd.Series([10, 20, 30, 40.], name='sr2')
```

```
## sr2 => 번호 축 이름 변경
```

```
dataSR22 = dataSR2.rename_axis( index='번호' )
```

```
display( dataSR22 )
```

```
## 축 이름 제거
```

```
dataSR22 = dataSR2.rename_axis( index=None )
```

```
display( dataSR22 )
```



행 & 열 삭제

◆ 행/열 삭제

- inplace 매개변수 : 원본 데이터 적용 여부 설정

```
DataFrame/Series.drop( labels=[인덱스, .... ], axis=0, inplace )
```

```
DataFrame/Series.drop( index=[인덱스, .... ], inplace )
```

```
DataFrame.drop( labels=[ 컬럼, .... ], axis=1, inplace)
```

```
DataFrame.drop( columns=[ 컬럼, .... ], inplace)
```

◆ 행/열 삭제

DataFrame 행 삭제

```
# 데이터프레임 형식으로 저장
exam_data = {'수학' : [ 90, 80, 70], '영어' : [ 98, 89, 95],
             '음악' : [ 85, 95, 100], '체육' : [ 100, 90, 90]}

df = pd.DataFrame(exam_data, index=['서준', '우현', '인아'])

# df 복제 => 변수 df2에 저장. df2의 1개 행(row) 삭제
df2 = df[:]
df2.drop('우현', inplace=True)

# df 복제 => 변수 df3에 저장. df3의 2개 행(row) 삭제
df3 = df[:]
df3.drop(['우현', '인아'], axis=0, inplace=True)
```

◆ 행/열 삭제

DataFrame 열 삭제

df 복제 => 변수 df4에 저장. df4의 1개 열(column) 삭제

```
df4 = df[:]
```

```
df4.drop('수학', axis=1, inplace=True)
```

df 복제 => 변수 df5에 저장. df5의 2개 열(column) 삭제

```
df5 = df[:]
```

```
df5.drop(['영어', '음악'], axis=1, inplace=True)
```

◆ 행/열 삭제

Series 원소 삭제

```
## Series 인스턴스 생성
```

```
dataSR1 = pd.Series([10, 20, 30, 40.])
```

```
dataSR2 = pd.Series([10, 20, 30, 40.], index=['영','일','이','삼'])
```

```
## 0번, 2번 원소 삭제. 원본 미적용
```

```
dataSR11=dataSR1.drop(labels=[0, 2], axis=0)
```

```
display(dataSR11)
```

```
## 0번, 2번 원소 삭제. 원본 적용
```

```
dataSR1.drop(labels=[0, 2], axis=0, inplace=True)
```

```
display(dataSR1)
```

행 & 열 & 원소 선택

◆ 원소 다루기

■ 원소 선택

.iloc[] 속성 : 정수형 인덱스 지정 속성. 슬라이싱 시 범위 끝 미포함

DataFrame/Series객체[정수인덱스]

DataFrame/Series객체.iloc[정수인덱스]

.loc[] 속성 : 라벨 인덱스 지정 속성. 슬라이싱 시 범위 끝 미포함

DataFrame/Series객체[라벨인덱스]

DataFrame/Series객체.loc[라벨인덱스]

◆ Series 원소

■ 원소 선택 – 한 개

```
## Series 인스턴스 생성
dataSR1 = pd.Series([10, 20, 30, 40.])

print( ' 정수 인덱스 - 0번 원소' )
print( ' dataSR1[0]      :', dataSR1[0] )
print( ' dataSR1.iloc[0] :', dataSR1.iloc[0])
```

◆ Series 원소

■ 원소 선택 – 한 개

```
## Series 인스턴스 생성
dataSR1 = pd.Series([10, 20, 30, 40.],
                    index=['영', '일', '이', '삼'])

print( ' 정수 인덱스 - 0번 원소' )
print( ' dataSR1[0]      :', dataSR1[0] )
print( ' dataSR1.iloc[0] :', dataSR1.iloc[0] )

print( ' 라벨 인덱스 - 영번 원소' )
print( ' dataSR1["영"]   :', dataSR1["영"] )
print( ' dataSR1.loc[ "영" ] :', dataSR1.loc["영"] )
```


◆ Series 원소

■ 원소 선택 - 여러개

```
## Series인스턴스 생성
dataSR1 = pd.Series([10, 20, 30, 40.],
                    index=['영', '일', '이', '삼'])

## Series인스턴스[ [인덱스1, 인덱스2] ]
print( '정수 인덱스 - 0번, 1번 원소' )
print( 'dataSR1[[0, 1]]      :', dataSR1[[0, 1]] )
print( 'dataSR1.iloc[[0, 1]]:', dataSR1.iloc[[0, 1]])
```

◆ Series 원소

■ 원소 선택 - 여러개

```
## Series인스턴스 생성
```

```
dataSR1 = pd.Series([10, 20, 30, 40.],  
                    index=['영','일','이','삼'])
```

```
## Series인스턴스[ [인덱스1, 인덱스2] ]
```

```
print( '라벨 인덱스 - 영번, 일번 원소' )
```

```
print( 'dataSR1["열", "일"]      :', dataSR1[ ["영", "영"]] )
```

```
print( 'dataSR1.loc[["영","일"] :', dataSR1.loc[["영","일"]] )
```

◆ Series 원소

■ 원소 선택 - 여러개

```
## Series인스턴스 생성
```

```
dataSR1 = pd.Series([10, 20, 30, 40.])
```

```
## Series인스턴스[ 시작인덱스 : 끝인덱스:간격 ]
```

```
print( ' 정수 인덱스 - 0번 ~ 2번 원소' )
```

```
print( ' dataSR1[0]      :', dataSR1[ 0 : 3 : 2 ] )
```

```
print( ' dataSR1.iloc[0] :', dataSR1.iloc[ 0 : 3 : 2 ] )
```

◆ DataFrame 원소

■ 행 선택 – 한 개

```
## DataFrame 인스턴스 생성 -----  
dataDF1 = pd.DataFrame( [[10, 20, 30, 40.],  
                          [11, 22, 33, 44.]] )  
  
##- 1개 행 선택 -----  
one_row = dataDF1.iloc[ 0 ]  
display( one_row , type(one_row))
```

◆ DataFrame 원소

■ 행 선택 – 한 개

```
## DataFrame 인스턴스 생성 -----  
dataDF2 = pd.DataFrame( [[10, 20, 30, 40.],[11, 22, 33, 44.]],  
                        columns=['영', '일', '이', '삼'],  
                        index=['row0', 'row1'])
```

```
##- 1개 행 선택 -----
```

```
one_row = dataDF2.iloc[ 0 ]  
display( one_row , type(one_row))
```

```
one_row = dataDF2.loc[ 'row0' ]  
display( one_row , type(one_row))
```

◆ DataFrame 원소

■ 행 선택 – 여러 개

```
## DataFrame 인스턴스 생성 -----  
dataDF1 = pd.DataFrame( [[10, 20, 30, 40.],  
                          [11, 22, 33, 44.]] )  
  
##- 1개 행 선택 -----  
two_row = dataDF1.iloc[ [0, 2] ]  
display(two_row , type(two_row))
```

◆ DataFrame 원소

■ 행 선택 – 여러 개

```
## DataFrame 인스턴스 생성 -----  
dataDF2 = pd.DataFrame( [[10, 20, 30, 40.],[11, 22, 33, 44.]],  
                        columns=['영', '일', '이', '삼'],  
                        index=['row0', 'row1'])  
  
##- 2개 행 선택 -----  
two_row = dataDF2.iloc[ [0, 2] ]  
display( two_row , type(two_row))  
  
two_row = dataDF2.loc[ ['row0' , 'row1'] ]  
display( two_row , type(two_row))
```

◆ DataFrame 원소

■ 행 선택 - 여러 개

```
## DataFrame 인스턴스 생성 -----  
dataDF2 = pd.DataFrame( [[10, 20, 30, 40.],[11, 22, 33, 44.]],  
                        columns=['영', '일', '이', '삼'],  
                        index=['row0', 'row1'])  
  
##- 2개 행 선택 -----  
two_row = dataDF2.iloc[ 0 : 2 ]  
display( two_row , type(two_row))  
  
two_row = dataDF2.loc[ 'row0' : 'row1' ]  
display( two_row , type(two_row))
```


◆ DataFrame 원소

■ 열 선택 – 한 개

```
## DataFrame 인스턴스 생성 -----  
dataDF1 = pd.DataFrame( [[10, 20, 30, 40.],  
                        [11, 22, 33, 44.]] )  
  
##- 1개 열 선택 -----  
one_col = dataDF1[0]  
display( one_col , type(one_col))
```

◆ DataFrame 원소

■ 열 선택 – 한 개

```
## DataFrame 인스턴스 생성 -----
dataDF2 = pd.DataFrame( [[10, 20, 30, 40.],[11, 22, 33, 44.]],
                        columns=['영', '일', '이', '삼'],
                        index=['row0', 'row1'])

##- 1개 열/컬럼 선택 -----
one_col = dataDF2[0]
display( one_col , type(one_col))

one_col = dataDF2['영']
display( one_col , type(one_col))
```

◆ DataFrame 원소

■ 열 선택 – 여러 개

```
## DataFrame 인스턴스 생성 -----  
dataDF1 = pd.DataFrame( [[10, 20, 30, 40.],  
                          [11, 22, 33, 44.]])  
  
##- 1개 행 선택 -----  
two_row = dataDF1.iloc[ [0, 2] ]  
display(two_row , type(two_row))
```

◆ DataFrame 원소

■ 열 선택 – 여러 개

```
## DataFrame 인스턴스 생성 -----
dataDF2 = pd.DataFrame( [[10, 20, 30, 40.],[11, 22, 33, 44.]],
                        columns=['영', '일', '이', '삼'],
                        index=['row0', 'row1'])

##- 2개 열/컬럼 선택 -----
two_rows = dataDF2[[0, 2]]
display( two_rows, type(two_rows) )

two_rows = dataDF2[['영', '이']]
display( two_rows, type(two_rows) )
```

◆ DataFrame 원소

■ 열 선택 – 여러 개

```
## DataFrame 인스턴스 생성 -----  
dataDF2 = pd.DataFrame( [[10, 20, 30, 40.],[11, 22, 33, 44.]],  
                        columns=['영', '일', '이', '삼'],  
                        index=['row0', 'row1'])  
  
##- 2개 열/컬럼 선택 -----  
two_rows1 = dataDF2.iloc[:, 0 : 3 : 2 ]  
display( two_rows1, type(two_rows1) )  
  
two_rows2 = dataDF2.loc[:, '영':'이':2 ]  
display( two_rows2, type(two_rows2) )
```

◆ DataFrame 원소

■ 원소 선택 – 한 개

행과 열이 교차하는 지점의 원소

`DataFrame`객체.`iloc` [행인덱스, 열인덱스]

`DataFrame`객체.`loc` [행라벨인덱스, 열라벨인덱스]

◆ DataFrame 원소

■ 원소 선택 – 한 개

```
##- -----  
##- 1개 0번째 0번 요소 선택  
##- -----  
one_e = dataDF2.iloc[0, 0]  
print(one_e, type(one_e))  
  
one_e = dataDF2.loc['row0', '영']  
print(one_e, type(one_e))
```

◆ DataFrame 원소

■ 원소 선택 – 여러 개

```
##- -----  
##- 여러 개 0번째 행 0번째 열, 0번째 행 2번째 열 요소 선택  
##- -----  
  
two_e = dataDF2.iloc[ 0, [0, 2] ]  
print(two_e, type(two_e))  
  
two_e = dataDF2.loc[ 'row0', [ '영', '이' ] ]  
print(two_e, type(two_e))
```