

# Series 자료형

## #01. Pandas 패키지

쉽고 직관적인 관계형 또는 분류된 데이터로 작업 할 수 있도록 설계된 빠르고 유연하며 표현이 풍부한 데이터 구조를 제공하는 Python 패키지

실용적인 실제 데이터 분석을 수행하기 위한 고수준의 객체를 사용할 수 있다.

### 1) Pandas가 사용되는 경우

- SQL 테이블 또는 Excel 스프레드 시트에서와 같이 이질적으로 유형이 지정된 열이있는 테이블 형식 데이터
- 정렬되고 정렬되지 않은 시계열 데이터
- 행 및 열 레이블이 포함 된 임의의 행렬 데이터
- 일반적인 통계 데이터 세트

Pandas의 두 가지 주요 데이터 구조인 Series (1차원) 및 DataFrame (2차원)은 재무, 통계, 사회 과학 및 다양한 엔지니어링 분야의 일반적인 사용 사례의 대부분을 처리함

### 2) Pandas 데이터 구조

차원	이름	설명
1차원	Series	균일한 유형의 배열로 표시된 1차원 데이터
2차원	DataFrame	행과 열이있는 크기가 가변적인 테이블 형식의 2차원 데이터

### 3) Pandas 패키지 설치

```
$ pip install --upgrade pandas
```

### 4) 필요한 패키지 가져오기

```
from pandas import Series
from pandas import DataFrame
from pandas import read_csv, read_excel
```

## #02. Series 자료형

### 1) Series 자료형의 구조

pandas 패키지에 포함되어 있는 자료형으로 `list > numpy array > series` 순으로 기능이 확장된다.

인덱스를 명시적으로 포함하고 있는 형태이며, 엑셀의 열 하나를 표현하고 있다고 이해할 수 있다.

```
# 기본 시리즈 만들기
# -> 리스트를 통해 만들 수 있다.
# -> 즉 리스트 자료형을 가공하여 생성된 데이터 구조.
items = [10, 30, 50, 70, 90]
```

```
column = Series(items)
column
```

```
0    10
1    30
2    50
3    70
4    90
dtype: int64
```

```
# -> 인덱스를 활용한 개별 값 확인
print(column[0])
print(column[2])
print(column[4])
```

```
10
50
90
```

```
# 시리즈의 값들만 추출
print( column.values )

# 타입을 확인하면 Numpy 배열임을 알 수 있다.
print(type(column.values))
```

```
[10 30 50 70 90]
<class 'numpy.ndarray'>
```

```
# -> 시리즈의 값들을 저장하고 있는 numpy 배열을 list로 변환
value_list = list(column.values)
value_list
```

```
[10, 30, 50, 70, 90]
```

```
# 시리즈의 색인(index)만 추출
column.index
```

```
RangeIndex(start=0, stop=5, step=1)
```

```
# 색인들의 타입 확인
type(column.index)
```

```
pandas.core.indexes.range.RangeIndex
```

```
# 시리즈의 색인(index)을 list로 변환
mylist = list(column.index)
mylist
```

```
[0, 1, 2, 3, 4]
```

### 3) Series 자료형 활용

```
# 특정 조건에 맞는 항목들만 추출
# -> 이름[이름에 대한 조건식]
# -> 30초과인 항목들만 추출
in1 = column[column > 30]
in1
```

```
2    50
3    70
4    90
dtype: int64
```

```
# -> 70이하 and 10초과인 항목들만 추출
in2 = column[column ≤ 70][column > 10]
in2
```

```
1    30
2    50
3    70
dtype: int64
```

```
# -> 10이하 or 70이상인 항목들만 추출 -> ()필수
in3 = column[(column ≤ 10) | (column ≥ 70)]
in3
```

```
0    10
3    70
4    90
dtype: int64
```

```
# 인덱스를 직접 지정하면서 시리즈 만들기
# -> 지난주 주말에 대한 매출액
week1 = Series([290000, 310000], index=['sat', 'sun'])
# -> 이번주 주말에 대한 매출액
week2 = Series([120000, 220000], index=['sun', 'sat'])

print( week1 )
print("-" * 30)
```

```
print( week2 )  
print( "-" * 30 )
```

```
# 시리즈 객체의 사칙연산  
# -> index가 동일한 항목끼리 연산이 수행된다.  
merge = week1 + week2  
print(merge)
```

```
sat    290000  
sun    310000  
dtype: int64
```

---

```
sun    120000  
sat    220000  
dtype: int64
```

---

```
sat    510000  
sun    430000  
dtype: int64
```