

## [ 2-4. 시계열 다루기(datetime) ]

### 1. datetime 라이브러리

- 파이썬에서 날짜와 시간을 다루고자 하는 경우 datetime 라이브러리를 많이 사용함.
- datetime 라이브러리는 파이썬 표준라이브러리임.
- datetime 라이브러리에는 date객체(날짜), time객체(시간), datetime객체(날짜와 시간)가 있고, 이러한 객체들을 이용하여 날짜와 시간을 다루게 됨.

#### 1) datetime 객체

```
In [1]: from datetime import datetime
```

```
In [2]: dt = [datetime(2023, 1, 31), datetime(2023, 2, 28), datetime(2023, 3, 31),  
              datetime(2023, 4, 30), datetime(2023, 5, 31), datetime(2023, 6, 30)]
```

```
In [3]: dt
```

```
Out[3]: [datetime.datetime(2023, 1, 31, 0, 0),  
         datetime.datetime(2023, 2, 28, 0, 0),  
         datetime.datetime(2023, 3, 31, 0, 0),  
         datetime.datetime(2023, 4, 30, 0, 0),  
         datetime.datetime(2023, 5, 31, 0, 0),  
         datetime.datetime(2023, 6, 30, 0, 0)]
```

```
In [4]: dt[0].year
```

```
Out[4]: 2023
```

```
In [5]: dt[0].month
```

```
Out[5]: 1
```

```
In [6]: dt[0].day
```

```
Out[6]: 31
```

```
In [7]: [x.month for x in dt] #리스트 컴프리헨션을 통해 year, month, day 등 추출
```

```
Out[7]: [1, 2, 3, 4, 5, 6]
```

```
In [8]: [x.year for x in dt]
```

```
Out[8]: [2023, 2023, 2023, 2023, 2023, 2023]
```

```
In [9]: [(x.year, x.month) for x in dt]
```

```
Out[9]: [(2023, 1), (2023, 2), (2023, 3), (2023, 4), (2023, 5), (2023, 6)]
```

### 문자형태로 출력(strftime 메서드)

```
In [10]: dt[0].strftime('%Y-%m-%d')
```

```
Out[10]: '2023-01-31'
```

```
In [11]: dt[1].strftime('%Y.%m.%d')
```

```
Out[11]: '2023.02.28'
```

```
In [12]: [x.strftime('%Y.%m.%d') for x in dt]
```

```
Out[12]: ['2023.01.31',  
          '2023.02.28',  
          '2023.03.31',  
          '2023.04.30',  
          '2023.05.31',  
          '2023.06.30']
```

### 날짜간 차이 계산

```
In [13]: diff = datetime(2023, 7, 1) - datetime(2023, 1, 1)  
diff
```

```
Out[13]: datetime.timedelta(days=181)
```

```
In [14]: diff.days
```

```
Out[14]: 181
```

### 오늘 날짜 객체 생성

```
In [15]: datetime.today()
```

```
Out[15]: datetime.datetime(2022, 12, 1, 23, 43, 35, 181691)
```

```
In [16]: datetime.now()
```

```
Out[16]: datetime.datetime(2022, 12, 1, 23, 43, 35, 366120)
```

```
In [ ]:
```

## 2) timedelta 객체

```
In [17]: from datetime import timedelta
```

```
In [18]: start = datetime(2023, 1, 1)
```

```
In [19]: start + timedelta(31) #days
```

```
Out[19]: datetime.datetime(2023, 2, 1, 0, 0)
```

```
In [20]: dt
```

```
Out[20]: [datetime.datetime(2023, 1, 31, 0, 0),  
          datetime.datetime(2023, 2, 28, 0, 0),  
          datetime.datetime(2023, 3, 31, 0, 0),  
          datetime.datetime(2023, 4, 30, 0, 0),  
          datetime.datetime(2023, 5, 31, 0, 0),  
          datetime.datetime(2023, 6, 30, 0, 0)]
```

```
In [21]: [x + timedelta(1) for x in dt]
```

```
Out[21]: [datetime.datetime(2023, 2, 1, 0, 0),  
          datetime.datetime(2023, 3, 1, 0, 0),  
          datetime.datetime(2023, 4, 1, 0, 0),  
          datetime.datetime(2023, 5, 1, 0, 0),  
          datetime.datetime(2023, 6, 1, 0, 0),  
          datetime.datetime(2023, 7, 1, 0, 0)]
```

```
In [ ]:
```

## 2. pandas.to\_datetime

- 일반적으로 사용하는 날짜 시간의 문자열 형식을 datetime 객체로 변환

```
In [22]: dt
```

```
Out[22]: [datetime.datetime(2023, 1, 31, 0, 0),
          datetime.datetime(2023, 2, 28, 0, 0),
          datetime.datetime(2023, 3, 31, 0, 0),
          datetime.datetime(2023, 4, 30, 0, 0),
          datetime.datetime(2023, 5, 31, 0, 0),
          datetime.datetime(2023, 6, 30, 0, 0)]
```

```
In [23]: dtstr = [x.strftime('%Y.%m.%d') for x in dt]
          dtstr
```

```
Out[23]: ['2023.01.31',
          '2023.02.28',
          '2023.03.31',
          '2023.04.30',
          '2023.05.31',
          '2023.06.30']
```

```
In [24]: dtstr[0]
```

```
Out[24]: '2023.01.31'
```

```
In [ ]:
```

```
In [25]: import pandas as pd
```

```
In [26]: dtpd = pd.to_datetime(dtstr)  #문자열로 작성된 날짜를 datetime 객체로 변환
          dtpd
```

```
Out[26]: DatetimeIndex(['2023-01-31', '2023-02-28', '2023-03-31', '2023-04-30',
                        '2023-05-31', '2023-06-30'],
                        dtype='datetime64[ns]', freq=None)
```

```
In [27]: dtpd + timedelta(1)
```

```
Out[27]: DatetimeIndex(['2023-02-01', '2023-03-01', '2023-04-01', '2023-05-01',
                        '2023-06-01', '2023-07-01'],
                        dtype='datetime64[ns]', freq=None)
```

```
In [ ]:
```