

[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 [42]: from datetime import datetime
```

```
In [43]: 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 [44]: dt
```

```
Out[44]: [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 [45]: dt[0].year
```

```
Out[45]: 2023
```

```
In [46]: dt[0].month
```

```
Out[46]: 1
```

```
In [47]: dt[0].day
```

```
Out[47]: 31
```

```
In [48]: [x.month for x in dt] #리스트 컴프리헨션을 통해 year, month, day 등 추출
```

```
Out[48]: [1, 2, 3, 4, 5, 6]
```

```
In [49]: [x.year for x in dt]
```

```
Out[49]: [2023, 2023, 2023, 2023, 2023, 2023]
```

```
In [50]: [(x.year, x.month) for x in dt]
```

```
Out[50]: [(2023, 1), (2023, 2), (2023, 3), (2023, 4), (2023, 5), (2023, 6)]
```

문자형태로 출력(strftime 메서드)

```
In [51]: dt[0].strftime('%Y-%m-%d')
```

```
Out[51]: '2023-01-31'
```

```
In [52]: dt[1].strftime('%Y.%m.%d')
```

```
Out[52]: '2023.02.28'
```

```
In [53]: [x.strftime('%Y.%m.%d') for x in dt]
```

```
Out[53]: ['2023.01.31',  
          '2023.02.28',  
          '2023.03.31',  
          '2023.04.30',  
          '2023.05.31',  
          '2023.06.30']
```

날짜간 차이 계산

```
In [56]: diff = datetime(2023, 7, 1) - datetime(2023, 1, 1)  
diff
```

```
Out[56]: datetime.timedelta(days=181)
```

```
In [57]: diff.days
```

```
Out[57]: 181
```

오늘 날짜 객체 생성

```
In [58]: datetime.today()
```

```
Out[58]: datetime.datetime(2022, 11, 29, 17, 1, 56, 948754)
```

```
In [59]: datetime.now()
```

```
Out[59]: datetime.datetime(2022, 11, 29, 17, 2, 22, 469811)
```

```
In [ ]:
```

2) timedelta 객체

```
In [67]: from datetime import timedelta
```

```
In [68]: start = datetime(2023, 1, 1)
```

```
In [70]: start + timedelta(31) #days
```

```
Out[70]: datetime.datetime(2023, 2, 1, 0, 0)
```

```
In [71]: dt
```

```
Out[71]: [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 [73]: [x + timedelta(1) for x in dt]
```

```
Out[73]: [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 [74]: dt
```

```
Out[74]: [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 [76]: dtstr = [x.strftime('%Y.%m.%d') for x in dt]
dtstr
```

```
Out[76]: ['2023.01.31',
'2023.02.28',
'2023.03.31',
'2023.04.30',
'2023.05.31',
'2023.06.30']
```

```
In [83]: dtstr[0]
```

```
Out[83]: '2023.01.31'
```

```
In [ ]:
```

```
In [60]: import pandas as pd
```

```
In [78]: dtpd = pd.to_datetime(dtstr) #문자열로 작성된 날짜를 datetime 객체로 변환
dtpd
```

```
Out[78]: 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 [85]: dtpd + timedelta(1)
```

```
Out[85]: 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 [ ]:
```