

Dates and Times

파이썬에서 기본적으로 지원하는 시간 관련 라이브러리들은 다음과 같다.

- `time` — Clock Time
- `datetime` — Date and Time Value Manipulation
- `calendar` — Work with Dates

이 외에 파이썬에서 기본적으로 지원하는 라이브러리는 아니지만, 편리하게 사용할 수 있도록 오픈 소스로 제공하는 라이브러리들은 다음과 같다.

- [dateutil](#)
- [pytz](#)
- [pendulum](#)

datetime

`datetime` 을 먼저 임포트 한다.

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

날짜 객체를 생성할 때는 다음과 같이 각각 년, 월, 일 순으로 넣어준다.

```
In [2]: date = datetime.date(year=2018, month=6, day=7)
```

시간 객체를 생성할 때는 다음과 같이 `datetime.time` 으로 접근하여 시, 분, 초, 마이크로초 단위로 각각 넣어준다.

```
In [3]: time = datetime.time(hour=12, minute=30, second=19, microsecond=463198)
```

`datetime.datetime` 객체에 위에서 입력한 모든 정보를 받을 수 있다.

```
In [4]: dt = datetime.datetime(year=2013,
                               month=6,
                               day=7,
                               hour=12,
                               minute=30,
                               second=19,
                               microsecond=463198)
```

`timedelta` 는 시간과 시간을 계산하기 위한 객체이다.

```
In [5]: td = datetime.timedelta(weeks=2,
                                days=5,
                                hours=10,
                                minutes=20,
                                seconds=6.73,
                                milliseconds=99,
                                microseconds=8)
```

```
In [6]: date + td
```

```
Out[6]: datetime.date(2018, 6, 26)
```

```
In [7]: dt + td
```

```
Out[7]: datetime.datetime(2013, 6, 26, 22, 50, 26, 292206)
```

`datetime.time` 객체와 `datetime.timedelta` 객체는 서로 연산이 불가능하다.

```
In [8]: time + td
```

```
-----
TypeError                                 Traceback (most recent call last)
<ipython-input-8-1bc8485497ad> in <module>
----> 1 time + td

TypeError: unsupported operand type(s) for +: 'datetime.time' and 'datetime.timedelta'
```

pandas Timestamp

판다스에서의 `Timestamp` 객체는 파이썬의 `datetime` 객체와 같다.

```
In [9]: import pandas as pd
```

```
In [10]: issubclass(pd.Timestamp, datetime.datetime)
```

```
Out[10]: True
```

```
In [11]: pd.Timestamp(year=2018,
                       month=12,
                       day=21,
                       hour=5,
                       minute=10,
                       second=8)
```

```

second=0,
microsecond=99)

Out[11]: Timestamp('2018-12-21 05:10:08.000099')

In [12]: pd.Timestamp('2016/1/10')
Out[12]: Timestamp('2016-01-10 00:00:00')

In [13]: pd.Timestamp('2014-5/10')
Out[13]: Timestamp('2014-05-10 00:00:00')

In [14]: pd.Timestamp('Jan 3, 2019 20:45.56')
Out[14]: Timestamp('2019-01-03 20:45:33')

In [15]: pd.Timestamp('2016-01-05T05:34:43.123456789')
Out[15]: Timestamp('2016-01-05 05:34:43.123456789')

In [16]: pd.Timestamp(500)
Out[16]: Timestamp('1970-01-01 00:00:00.000000500')

In [17]: pd.Timestamp(5000, unit='D')
Out[17]: Timestamp('1983-09-10 00:00:00')

In [18]: pd.to_datetime('2015-5-13')
Out[18]: Timestamp('2015-05-13 00:00:00')

In [19]: pd.to_datetime('2015-13-5', dayfirst=True)
Out[19]: Timestamp('2015-05-13 00:00:00')

In [20]: pd.to_datetime('Start Date: Sep 30, 2017 Start Time: 1:30 pm',
                        format='Start Date: %b %d, %Y Start Time: %I:%M %p')
Out[20]: Timestamp('2017-09-30 13:30:00')

In [21]: pd.to_datetime(100, unit='D', origin='2013-1-1')
Out[21]: Timestamp('2013-04-11 00:00:00')

In [22]: pd.to_timedelta('67:15:45.454')
Out[22]: Timedelta('2 days 19:15:45.454000')

In [23]: s = pd.Series([10, 100])
In [24]: pd.to_timedelta(s, unit='s')
Out[24]: 0    00:00:10
         1    00:01:40
         dtype: timedelta64[ns]

In [25]: time_strings = ['2 days 24 minutes 89.67 seconds', '00:45:23.6']
In [26]: pd.to_timedelta(time_strings)
Out[26]: TimedeltaIndex(['2 days 00:25:29.670000', '0 days 00:45:23.600000'], dtype='timedelta64[ns]', freq=None)

In [27]: pd.Timedelta('12 days 5 hours 3 minutes') * 2
Out[27]: Timedelta('24 days 10:06:00')

In [28]: pd.Timestamp('1/1/2017') + pd.Timedelta('12 days 5 hours 3 minutes') * 2
Out[28]: Timestamp('2017-01-25 10:06:00')

```

TimeSeries 는 LMS 사이트의 강의자료를 참고.

IPA 주관 인공지능센터 기본(fundamental) 과정

- GitHub link: [here](#)
- E-Mail: windkyle7@gmail.com