

Chapter. 03

파이썬 프로그래밍 언어

I 날짜와 시간

FASTCAMPUS
ONLINE

금융공학/퀀트 I

강사. 서찬웅

I 이번 시간에 배울 내용

1. 날짜와 시간에 관한 모듈
2. datetime
3. dateutil

I 표준 모듈(Standard Library)

- 파이썬에서 파일을 찾거나, 시간을 확인하거나, 난수를 생성하거나 그런 일들을 하는 모듈이 존재하는데 이를 표준 라이브러리라고 합니다.(모듈 설명에서 한번 언급했습니다.)
- 이번 시간에는 표준 라이브러리에서 시간과 날짜 모듈에 대해서 알아보겠습니다.
 - 시간과 날짜 표준 라이브러리 모듈 이름 : datetime

객체이름	용도
date	날짜 다루기
time	시각 다루기
datetime	일시 다루기
timedelta	두 일시의 차 다루기

Python » 3.6.1 » Documentation »

Previous topic
10. Full Grammar specification

Next topic
1. Introduction

This Page
Report a Bug
Show Source

The Python Standard Library

While [The Python Language Reference](#) describes the exact syntax and optional components that are commonly included in Python distributions, this document describes the standard library modules that are included in the Python standard library.

Python's standard library is very extensive, offering a wide range of modules that provide functionality as file I/O that would otherwise be inaccessible to Python programs. These modules are explicitly designed to encourage and enhance the portability of Python programs.

The Python installers for the Windows platform usually include the standard library, so it may be necessary to use the packaging tools provided with the installer to install additional modules.

In addition to the standard library, there is a growing collection of third-party modules available for download from the [Python Package Index](#).

- datetime 모듈을 사용하면 문자열로 된 날짜를 datetime 객체로 변환하고, 반대로 변환할 수 있습니다.
- 주차 계산, 요일 계산등을 쉽게 할 수 있어서 시계열 분석에 활용할 수 있습니다.

I datetime 모듈

- date 객체는 날짜(연, 월, 일)를 취급합니다.

메서드이름	설명	반환값
date(year, month, day)	지정한 날짜의 date 객체를 생성	datetime.date
today()	오늘 날짜의 date 객체를 생성	datetime.date
weekday()	월요일을 0, 일요일을 6으로 하여 요일을 반환	int
isoweekday()	월요일을 1, 일요일을 7로 하여 요일을 반환	int
isoformat()	ISO 8601 형식(YYYY-MM-DD)의 날짜 문자열 반환	str
strftime(format)	지정한 포맷에 따라 날짜 문자열을 반환	str
__str__()	isoformat()과 같은 결과를 반환	str

객체이름	설명	반환값
year	년 값을 반환	int
month	월 값을 반환	int
day	일 값을 반환	int

```
# date 샘플 코드
from datetime import date
```

```
newyearsday = date(2017, 6, 3)
```

```
newyearsday
```

```
datetime.date(2017, 6, 3)
```

```
newyearsday.year, newyearsday.month, newyearsday.day
(2017, 6, 3)
```

```
newyearsday.weekday()
```

```
5
```

```
newyearsday.isoformat()
```

```
'2017-06-03'
```

```
str(newyearsday)
```

```
'2017-06-03'
```

```
newyearsday.strftime('%Y/%m/%d')
```

```
'2017/06/03'
```

```
newyearsday.strftime('%Y/%m/%d (%a)')
```

```
'2017/06/03 (Sat)'
```

```
date.today()
```

```
datetime.date(2017, 6, 2)
```

I datetime 모듈

- datetime 객체는 시각을 다룬다. 시분초뿐만 아니라 마이크로초도 포함된다.

메서드이름	설명	반환값
time(hour=0, minute=0, second=0, microsecond=0, tzinfo=None)	지정한 시각의 time 객체를 생성	datetime.time
today()	기본 표준시간대의 현재 일시를 반환	datetime.datetime
date()	같은 연월일의 date 객체를 반환	datetime.date
time()	같은 시분초의 time 객체를 반환	datetime.time
isoformat()	ISO 8601 형식	str
strftime(format) strptime(format)	지정한 포맷에 따라 날짜 문자열을 반환 지정한 포맷에 따라 날짜 객체로 반환	str datetime.datetime
__str__()	isoformat()과 같은 결과를 반환	str
tzname()	표준시간대 이름의 문자열을 반환	str

객체이름	설명	반환값
year	년 값을 반환	int
month	월 값을 반환	int
day	일 값을 반환	int
hour	시 값을 반환	int
minute	분 값을 반환	int
second	초 값을 반환	int
microsecond	마이크로초 값을 반환	int
tzinfo	표준시간대 정보	객체

```
from datetime import datetime
```

```
today = datetime.today()
```

```
today.date()
```

```
datetime.date(2017, 6, 2)
```

```
today.time()
```

```
datetime.time(14, 8, 18, 88234)
```

```
today.isoformat()
```

```
'2017-06-02T14:08:18.088234'
```

```
today.strftime("%Y-%m-%d hh:mm:ss")
```

```
'2017-06-02 hh:mm:ss'
```

I 날짜 변경 포맷 형식

- strftime() 메소드를 사용하면 datetime 객체를 문자열로 변경할 수 있다고 설명했습니다.
- 변경시에 사용될 format을 정리했습니다.

%Y	Year with century as a decimal number.	0001, 0002, ..., 2013, 2014, ..., 9998, 9999
%m	Month as a zero-padded decimal number.	01, 02, ..., 12
%d	Day of the month as a zero-padded decimal number.	01, 02, ..., 31
%H	Hour (24-hour clock) as a zero-padded decimal number.	00, 01, ..., 23
%M	Minute as a zero-padded decimal number.	00, 01, ..., 59
%S	Second as a zero-padded decimal number.	00, 01, ..., 59

I dateutil 모듈

- dateutil 모듈은 datetime 모듈에 대한 확장 기능을 제공합니다.
 - 다양한 문자열 형식의 날짜 구문
 - 날짜의 차이 계산
- 별도로 설치를 하여야 사용할 수 있습니다.
 - `pip install python-dateutil`
- relativedelta 모듈을 사용하면 날짜 사이의 차이를 계산할 수 있다.
 - `relativedelta(datetime1, datetime2)`

I dateutil 모듈

```
from dateutil.relativedelta import relativedelta
from datetime import datetime, date
```

```
now = datetime.now()
```

```
now + relativedelta(months =+ 1)
```

```
datetime.datetime(2017, 7, 2, 15, 24, 47, 659553)
```

```
now + relativedelta(months =- 1, weeks =+1)
```

```
datetime.datetime(2017, 5, 9, 15, 24, 47, 659553)
```

```
from dateutil.relativedelta import MO, TU, WE, TH, FR, SA, SU
```

```
now + relativedelta(weekday = SA)
```

```
datetime.datetime(2017, 6, 3, 15, 24, 47, 659553)
```

```
now + relativedelta(day = 30, weekday = SA(-1))
```

```
datetime.datetime(2017, 6, 24, 15, 24, 47, 659553)
```

```
date(2017,1,1) + relativedelta(yearday = 100)
```

```
datetime.date(2017, 4, 10)
```

```
relativedelta(now, date(2017,10,10))
```

```
relativedelta(months=-4, days=-7, hours=-8, minutes=-35, seconds=-13, microseconds=+659553)
```


Itime 모듈

- time 모듈의 sleep 메소드를 사용하면 프로그램의 동작을 정해진 시간 만큼 정지할 수 있습니다.

```
from time import sleep
```

```
for x in range(0,10):  
    print (x)  
    sleep(1)
```

```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9
```

- 데이터 수집시 프로그램의 동작을 잠시 정지하고 싶을 때 사용합니다.

I 정리

- datetime의 사용법
- 문자열 < - > datetime 객체
- 시간 차이 계산
- sleep

감사합니다