

## 22. Python Date와 Scheduling

# 1. Python 날짜시간 모듈에 관련 클래스

## 1. 날짜시간 모듈에 관련 클래스

Class	내용
datetime.date	일반적으로 사용되는 그레고리안 달력의 년,월,일을 표현
datetime.time	시간을 시,분,초,마이크로 초,시간대로 표현
datetime.datetime	date클래스와 time클래스의 조합으로 년,월,일,시,분,초,마이크로 초, 시간대 정보를 표현
datetime.timedelta	두 날짜 혹은 시간 사이의 기간을 표현

# 1. Python 날짜시간 모듈에 관련 클래스

## 1] datetime.date

### 1) datetime.date(year, month, day)

- ① 일반적으로 사용되는 년, 월, 일로 표기되는 현재 그레고리안 달력의 날짜를 표현
- ② 숫자로 년, 월, 일을 입력 받아서 date 객체를 생성

### 2) date 객체를 반환하는 클래스 메서드

- ① date.fromtimestamp(timestamp): 타임스탬프 값을 인자로 받아서 date 객체를 반환
- ② date.fromordinal(ordinal) : 1년1월1일 이후로 누적된 날짜로부터 date 객체를 반환
- ③ date.today() : 현재 시스템의 오늘 날짜 date 객체를 반환

### 3) date 객체 속성

- ① date.year : 년(읽기 전용)
- ② date.month : 월(읽기 전용)
- ③ date.day : 일(읽기 전용)
- ④ date.max : date 객체의 최대값(9999년 12월 31일)
- ⑤ date.min : date 객체의 최소값(1년 1월 1일)

### 4) date 객체를 이용하여 다른 값/형식으로 변환하는 메서드

- ① date.replace(year, month, day) : 입력된 인자로 변경된 date 객체를 반환. 원본 객체는 변경되지 않음
- ② date.timetuple() : date 객체의 값을 time.struct\_time 시퀀스 객체에 할당.  
해당되는 정보가 없는 시,분,초는 '0'으로 초기화
- ③ date.toordinal() : 1년1월1일 이후로 date객체까지 누적된 날짜를 반환
- ④ date.weekday(): 요일을 정수로 변환하여 반환 (예:월요일=0, 일요일=6)
- ⑤ date.isoweekday() : date.weekday()와 동일하나 월요일=1 ~ 일요일=7로 나타냄.

# 1. Python 날짜시간 모듈에 관련 클래스

## 2] datetime.time class

### 1) datetime.time(hour, minute, second, microsecond, tzinfo)

- ① time 클래스는 시, 분, 초와 같은 시간을 표현
- ② 숫자로 시, 분, 초 마이크로 초, 시간대 정보(Time Zone)를 입력 받아서 time 객체를 생성
- ③ 입력된 인자가 아래의 조건을 따르지 않는 경우, ValueError가 발생

- 예시  
 $0 \leq \text{hour} < 24$   
 $0 \leq \text{minute} < 60$   
 $0 \leq \text{second} < 60$   
 $0 \leq \text{microsecond} < 1000000$

### 2) time 객체 속성

- ① time.hour : 시(0~23, 읽기 전용)
- ② time.minute : 분(0~59, 읽기 전용)
- ③ time.second : 초(0~59, 읽기 전용)
- ④ time.microsecond : 마이크로초(0~999999, 읽기 전용)
- ⑤ time.min : time 객체가 표현할 수 있는 최소값(time(0, 0, 0, 0))
- ⑥ time.max : time 객체가 표현할 수 있는 최대값(time(23, 59, 59, 999999))

### 4) time 클래스 지원 메서드

- ① time.replace([hour, minute, second, microsecond, tzinfo]): 입력된 인자로 변경된 time 객체를 반환.  
원본 객체는 변경되지 않음
- ② time.isoformat() : time 객체의 값을 'HH:MM:SS.mmmmmmm' 형식이나 'HH:MM:SS' 형식(마이크로초가 없는 경우)의 문자열을 반환
- ③ time.strftime(format) : 지정된 포맷에 맞춰 time 객체의 정보를 문자열로 반환

# 1. Python 날짜시간 모듈에 관련 클래스

## 3] datetime. **datetime** class

1) datetime.datetime (year, month, day , hour, minute, second, microsecond, tzinfo)

- ① date 클래스와 time 클래스의 조합
- ② 숫자로 각 인자를 받아서 datetime 객체를 생성하며, 년, 월, 일은 생략이 불가
- ③ 명시적으로 인자를 지정하여 입력할 수 있으며, 생략된 인자는 '0'으로 초기화
- ④ 입력된 인자가 아래의 조건을 따르지 않는 경우, ValueError가 발생

2) datetime 객체 속성

- ① datetime.year : 년(읽기 전용)
- ② datetime.month : 월(읽기 전용)
- ③ datetime.day : 일(읽기 전용)
- ④ datetime.hour : 시(0~23, 읽기 전용)
- ⑤ datetime.minute : 분(0~59, 읽기 전용)
- ⑥ datetime.second : 초(0~59, 읽기 전용)
- ⑦ datetime.microsecond : 마이크로초(0~999999, 읽기 전용)
- ⑧ datetime.min : datetime 객체가 표현할 수 있는 최소값. (datetime(1, 1, 1))
- ⑨ datetime.max : datetime 객체가 표현할 수 있는 최대값. ((9999, 12, 31, 23, 59, 59, 999999))

3) datetime 클래스를 다른 형식으로 변환하는 메서드

- ① datetime.date() : 원본 객체의 년,월,일 정보를 가지고 있는 date 객체 반환
- ② datetime.time() : 원본 객체의 시,분,초,마이크로초를 가지고 있는 time 객체 반환
- ③ datetime.replace() : 입력된 값으로 변경된 datetime 객체를 반환
- ④ datetime.timetuple() : datetime 객체의 값을 time.struct\_time 형식의 시퀀스 객체로 변환하여 반환

- 예시

```
MINYEAR(1) <= year <= MAXYEAR(9999)
```

```
0 <= month <= 12
```

```
0 <= day <= 주어진 년도 그리고 달의 날짜 개수
```

```
0 <= hour < 24
```

```
0 <= minute < 60
```

```
0 <= second < 60
```

```
0 <= microsecond < 1000000
```

# 1. Python 날짜시간 모듈에 관련 클래스

## 3] datetime.datetime class

### 4) datetime 클래스 출력과 관계된 메서드

- ① `datetime.weekday()`: 각 요일을 숫자로 나타냄 (월요일=0 ~ 일요일=6) (`time.struct_time.tm_wday` 참조)
- ② `datetime.isoweekday()`: ISO 형식에 맞도록 각 요일을 숫자로 나타냄.(월요일=1 ~ 일요일=7)
- ③ `datetime.isocalendar()`: ISO 형식에 맞는 날짜 표현(ISO year, ISO week number, ISO weekday)을 튜플로 반환.
- ④ `datetime.isoformat()`: datetime 객체를 'YYYY-MM-DDTHH:MM:SS.mmmmmm' 형식이나  
'YYYY-MM-DDTHH:MM:SS'(마이크로초가 0인 경우) 형식으로 변환하여 문자열로 반환
- ⑤ `datetime.ctime()`: datetime 객체를 'Thu Mar 26 18:53:03 2009' 형식의 문자열을 반환
- ⑥ `datetime.strftime(format)`: 입력된 포맷 형식에 맞추어 datetime 객체를 문자열로 반환
- ⑦ `datetime.microsecond`: 마이크로초(0~999999, 읽기 전용)

# 1. Python 날짜시간 모듈에 관련 클래스

## 4] datetime. **timedelta** class

1) datetime. **timedelta** (days seconds, microseconds, milliseconds, minutes, hours, weeks)

- ① timedelta 클래스는 두 날짜 혹은 시간 사이의 기간을 표현
- ② 생성되는 timedelta 객체의 값을 확인해보면 생성자에 전달된 값과 다른 것을 확인(이러한 시간, 날짜의 연산)
- ③ 동일한 기간을 표현하는 방식이 다양하게 표현(1weeks와 7days는 동일)될 수 있기 때문에, 입력된 값을 가지고

timedelta 객체에서 정규화 과정을 거쳐 유일한 표현 방식으로 변경

- ④ 생성된 timedelta 객체를 이용하여 date, datetime 객체를 변경할 수 있음.

지원하는 연산은 아래와 같으며, 각 객체 간의 비교연산도 가능

## 2) **timedelta** 사용예시

```
timedelta_3 = timedelta_1 + timedelta_2
```

```
date_2 = date_1 + timedelta
```

```
timedelta = date_2 - date_1
```

[ 정규화 결과 timedelta 객체에 저장되는 값 ]

```
0 <= microseconds < 1000000
```

```
0 <= seconds < 3600*24
```

(초로 표현할 수 있는 최대값은 하루)

```
0 <= days <= 999999999
```

## 2. Python subprocess

### 1] subprocess 개념

- 다른 Application(process)을 수행

### 2] 주요 함수

#### 1) subprocess.Popen

- 실제 설치된 Program 실행

#### 2) 예시

*# system의 계산기 프로그램 실행*

```
subprocess.Popen('c:\\Windows\\System32\\cmd.exe')
```

*# 현재 directory의 'date.py' 프로그램 실행*

```
subprocess.Popen('C:\\Users\\wtg\\AppData\\Local\\Programs\\Python\\Python35\\python.exe', 'date.py')
```