

Python 独自の日付と時刻モジュール：datetime

普通はこれだけでカレンダー計算などができる。

`datetime.datetime` 名前空間に関数が用意されている。

`datetime` はオブジェクトモジュール。

`datetime.datetime` で日時を得るには、`today()` または `now()` を使う。

違いは、`now()` は引き数にタイムゾーンオブジェクトを指定できる、というだけ。

`datetime.datetime` から `now()` や `today()` で `datetime` オブジェクトを取得できる。`datetime` オブジェクトからは、`date()` や `time()` でそれぞれ `date` オブジェクト、`time` オブジェクトを派生させることができる。

任意の `datetime`、`date`、`time` オブジェクトを作るには。

```
dt = datetime.datetime(2008, 10, 3, 9, 41, 30, 1000)
```

年、月、日が最低あれば作成可能。

```
d = datetime.date(2008, 10, 3)
```

年月日の3つが必ず必要。

```
t = datetime.time(9, 42, 15)
```

時だけで作成可能。

datetime の差分計算

ある `datetime` から `x` 日後・日前、`y` 時間後・時間前などの計算を行うためには、`datetime.timedelta` を使う。

```
datetime.timedelta(days=0, seconds=0, microseconds=0, milliseconds=0, minutes=0, hours=0, weeks=0)
```

```
h9 = datetime.timedelta(hours=9)
utcnow = datetime.datetime.now() - h9
```

datetime 同士の計算

`datetime` オブジェクト同士を計算すると、`TimeDelta` オブジェクトが返される。

```
dt_now = datetime.datetime.now()
dt_2000start = datetime.datetime(2000, 1, 1, 0, 0, 0)
dt_delta_since_2000 = dt_now - dt_2000start
print(type(dt_delta_since_2000))
```

`TimeDelta` オブジェクトは `total_seconds()` メソッドで総秒数になおせる。

`TimeDelta` は `Date Time` オブジェクト同士の「差分」なので「時間の長さ」を表す。ある時点の時刻を表す `Date Time` とは違うものなので、互いに変換はできない。その必要もない。

UNIX 時間を扱う : time

time は数値と日付時刻を相互変換するモジュール。

time.time(): UNIX 時間を数値で返す。

time.localtime(): UNIX 時間の数値を struct_time 型で返す。

time.mktime(): struct_time 型を UNIX 時間の数値に変換する。

日時フォーマット

datetime.datetime.strftime(format_string)

datetime.datetime.strptime(datetime_string, format_string)

strftime は、datetime 型を文字列型に変換する。

strptime は逆に、日時文字列を datetime 型に変換する。strptime では第 2 引数のフォーマット文字列指定は必須。

フォーマット指定子

フォーマット指定子の一覧 :

%a Weekday as locale 's abbreviated name.

%A Weekday as locale 's full name.

%w Weekday as a decimal number, where 0 is Sunday and 6 is Saturday.

%d Day of the month as a zero-padded decimal number.

%b Month as locale 's abbreviated name.

%B Month as locale 's full name.

%m Month as a zero-padded decimal number.

%y Year without century as a zero-padded decimal number.

%Y Year with century as a decimal number.

%H Hour (24-hour clock) as a zero-padded decimal number.

%I Hour (12-hour clock) as a zero-padded decimal number.

%p Locale 's equivalent of either AM or PM.

%M Minute as a zero-padded decimal number.

%S Second as a zero-padded decimal number.

%f Microsecond as a decimal number, zero-padded on the left right.

%z UTC offset in the form +HHMM or -HHMM (empty string if the the object is naive).

%Z Time zone name (empty string if the object is naive).

%j Day of the year as a zero-padded decimal number.

%U Week number of the year (Sunday as the first day of the week) as a zero padded decimal number. All days in a new year preceding the first Sunday are considered to be in week 0.

%W Week number of the year (Monday as the first day of the week) as a decimal number. All days in a new year preceding the first Monday are considered to be in week 0.

%c Locale 's appropriate date and time representation.

%x Locale 's appropriate date representation.

%X Locale 's appropriate time representation.

%% A literal '%' character.