Python - Date & Time

- **Introduction**
- **datetime**
  - datetime Instances
  - Methods to Create datetime Instances
  - Convert String of Datetime to datetime Instances
  - Do Some Arithmetic
  - Show the datetime in the specific format
- **Timezone**
  - tzinfo of datetime Instance
  - Timezone in Python
  - Naive vs Aware Datetime
  - Convert Timezone of Datetime Value

Introduction

- There are **three** separate modules in the standard library to work with dates and times:
  - **datetime**
    - supplies classes for manipulating dates and times
    - There are 4 classes that make up the high-level interface that most people will use:
      - **datetime.date**
      - **datetime.time**
      - **datetime.datetime**
      - **datetime.timedelta**
  - **calendar**
    - outputs calendars and provides functions using an idealized Gregorian calendar
  - **time**
    - provides time-related functions where dates are not needed

datetime - datetime Instances

- **Create datetime instances**

```
>>> from datetime import date, time, datetime
>>>
>>> date(year=2020, month=11, day=23)
datetime.date(2020, 11, 23)
>>>
>>> time(hour=13, minute=53, second=45)
datetime.time(13, 53, 45)
>>>
>>> datetime(year=2020, month=11, day=23, hour=13, minute=53, second=45)
datetime.datetime(2020, 11, 23, 13, 53, 45)
>>>
```

datetime - Methods to Create datetime Instances

- **Some convenient ways to create datetime instances**
  - **date.today()**
    - creates a **datetime.date** instance with the current local date.
  - **datetime.now()**
    - creates a **datetime.datetime** instance with the current local date and time.
  - **datetime.utcnow()**
    - creates a **datetime.datetime** instance with the current **UTC** date and time.
  - **datetime.combine()**
    - combines instances of **datetime.date** and datetime.time into a single datetime.datetime instance.

datetime - Convert String of Datetime to datetime Instances

- **Using Strings to Create Python datetime Instances**
  - **fromisoformat()**

```
>>> from datetime import date
>>>
>>> date.fromisoformat("2020-11-23")
datetime.date(2020, 11, 23)
>>>
```

  - **strptime()** → Here is the format code

```
>>> from datetime import datetime
>>>
>>> date_string = "23-11-2020 13:53:45"
>>> format_string = "%d-%m-%Y %H:%M:%S"
>>>
>>> datetime.strptime(date_string, format_string)
datetime.datetime(2020, 11, 23, 13, 53, 45)
>>>
```

datetime - Do Some Arithmetic

- **Option 1:**
  - Uses **timedelta** instances to represent time intervals
- **Option 2:**
  - Uses **dateutil.relativedelta** (Please google it for more information)

```
>>> from datetime import datetime, timedelta
>>>
>>> now = datetime.now()
>>>
>>> now
datetime.datetime(2020, 11, 23, 14, 43, 45, 134959)
>>>
>>> tomorrow = timedelta(days=1)
>>>
>>> tomorrow
datetime.timedelta(1)
>>>
>>> now + tomorrow
datetime.datetime(2020, 11, 24, 14, 43, 45, 134959)
>>>
>>>
>>> yesterday = timedelta(days=-1)
>>>
>>> now + yesterday
datetime.datetime(2020, 11, 22, 14, 43, 45, 134959)
>>>
```

datetime - Show the datetime in the specific format

- **strftime()** → Here is the format code

```
>>> from datetime import datetime
>>>
>>> now = datetime.now()
>>> now
datetime.datetime(2020, 11, 23, 14, 53, 38, 736043)
>>>
>>> now.strftime("%A, %B %d, %Y at %H:%M %p %Z")
'Monday, November 23, 2020 at 14:53 PM '
>>>
```

Timezone - tzinfo of datetime Instance

```
>>> from datetime import datetime, timezone
>>>
>>> datetime.now()
datetime.datetime(2020, 11, 23, 15, 52, 7, 373421)
>>>
>>> datetime.now(tz=timezone.utc)
datetime.datetime(2020, 11, 23, 8, 52, 54, 990898, tzinfo=datetime.timezone.utc)
>>>
>>> datetime.utcnow()
datetime.datetime(2020, 11, 23, 8, 53, 15, 529693)
>>>
>>>
>>> datetime.now(tzinfo=timezone.utc)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'tzinfo' is an invalid keyword argument for now()
>>>
>>>
>>> datetime(2020, 11, 23, 15, 55, 5, tzinfo=timezone.utc)
datetime.datetime(2020, 11, 23, 15, 55, 5, tzinfo=datetime.timezone.utc)
>>>
```

Timezone - Timezone in Python

- **datetime.timezone**

```
>>> from datetime import datetime, timedelta, timezone
>>>
>>> gmt_7 = timedelta(hours=7)
>>>
>>> tz_vn = timezone(gmt_7)
>>>
>>> datetime.now(tz=tz_vn)
datetime.datetime(2020, 11, 23, 16, 10, 0, 739040, tzinfo=datetime.timezone(datetime.timedelta(seconds=25200)))
>>>
```

- **pytz**

```
>>> import pytz
>>>
>>> from datetime import datetime
>>>
>>> tz_vn = pytz.timezone('Asia/Ho_Chi_Minh')
>>>
>>> datetime.now(tz=tz_vn)
datetime.datetime(2020, 11, 23, 16, 12, 17, 725245, tzinfo=<DstTzInfo 'Asia/Ho_Chi_Minh' +07+7:00:00 STD>)
>>>
```

Timezone - Timezone in Python (2)

- **dateutil.tz**

```
>>> from datetime import datetime
>>> from dateutil import tz
>>>
>>> tz_vn = tz.gettz('Asia/Ho_Chi_Minh')
>>>
>>> datetime.now(tz=tz_vn)
datetime.datetime(2020, 11, 23, 16, 22, 2, 345953, tzinfo=tzfile('/usr/share/zoneinfo/Asia/Ho_Chi_Minh'))
>>>
```

Timezone - Naive vs Aware Datetime

```
>>> from datetime import datetime
>>> import pytz
>>>
>>> dt_naive = datetime.now()
>>> dt_naive
datetime.datetime(2020, 11, 23, 16, 33, 14, 287339)
>>>
>>> dt_aware = datetime.now(tz=pytz.timezone('Asia/Ho_Chi_Minh'))
>>> dt_aware
datetime.datetime(2020, 11, 23, 16, 33, 40, 968885, tzinfo=<DstTzInfo 'Asia/Ho_Chi_Minh' +07+7:00:00 STD>)
>>>
```

Timezone - Convert Timezone of Datetime Value

- **To convert timezone**
  - Only convert **timezone of Aware Datetime Value**
  - Only use **astimezone**
  - **Don't use** *replace(tzinfo=...)*

```
>>> from datetime import datetime
>>> import pytz
>>>
>>> dt_naive = datetime.now()
>>> dt_naive
datetime.datetime(2020, 11, 23, 16, 33, 14, 287339)
>>>
>>> dt_aware = datetime.now(tz=pytz.timezone('Asia/Ho_Chi_Minh'))
>>> dt_aware
datetime.datetime(2020, 11, 23, 16, 33, 40, 968885, tzinfo=<DstTzInfo 'Asia/Ho_Chi_Minh' +07+7:00:00 STD>)
>>>
>>>
>>> dt_aware.astimezone(pytz.utc)
datetime.datetime(2020, 11, 23, 9, 33, 40, 968885, tzinfo=<UTC>)
>>>
```

Q & A

Thank You!