Python Programming

Datetime

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

Datetime Module

- Python has several standard/built-in modules that we use in practice
  - Such as datetime and time modules
- **datetime module** has 4 commonly used classes
  - date Class
  - time Class
  - datetime Class
  - timedelta Class
- Common mistake:
  - To import datetime module and use it as the class (you have to import the internal class)

Date and Time classes

- In this code we shows the internal time and date classes

```
3   import datetime
4
5   # Constructor: hour=0, minute=0, second=0, microsecond=0, tzinfo=None, *, fold=0
6   dt = datetime.time(14, 7)     # 2:07 pm
7   print(dt)                     # 14:07:00
8   print(dt.hour)                # 14
9   print(dt.minute)              # 7
10  print(dt.second)              # 0
11  print(dt.microsecond)         # 0
12  print(type(dt))               # <class 'datetime.time'>
13  print(datetime.time(14, 7, 59, 300))     # 14:07:59.000300
14
15  dt = datetime.date.today()
16  print(dt, type(dt))           # 2021-01-11 <class 'datetime.date'>: yyyy-mm-dd
17  # we can access dt.year or month or day
18  print(dt.ctime())             # Mon Jan 11 00:00:00 2021
19
```

Datetime and Timedelta Classes

- Datetime can represent both info for time and date
- We can also get the difference between 2 dates

```
2   import datetime
3
4   dt = datetime.datetime(2021, 1, 11, 14, 7, 59, 300)
5   # or use
6   print(dt.ctime())       # Mon Jan 11 14:07:59 2021
7
8   newdt = dt.replace(year=1990, day=25, second=13)
9   print(newdt)    # 1990-01-25 14:07:59.000300
10
11  delta = dt - newdt
12  print(delta, type(delta))   # 111309 days, 0:00:46 <class 'datetime.timedelta'>
13  print(delta.seconds)            # 46
14  print(delta.total_seconds())    # 977097646
15
16  # immutables
```

Passing Arguments

- Be careful from this common mistake
- The default arguments first values are used
  - C++ is different in that

```
from datetime import datetime

def hello1(curdate = datetime.now()):
    print(curdate)


for i in range(10):
    hello1()  # ALL of them are SAME!
    # 2021-01-11 21:36:03.142533


def hello2(curdate=None):
    if curdate is None:
        curdate = datetime.now()
    print(curdate)  # ALL of them are Different!


for i in range(10):
    hello2()

# Never use mutable or varying values as default arguments!
```

Passing Arguments

- More clear with mutable objects

```
2  def hello(lst = []):
3      lst.append(1)
4      print(lst)
5
6  hello() # [1]
7  hello() # [1, 1]
8  hello() # [1, 1, 1]
```

About Date & Time

- Although seems trivial, Date & time are source of **pain & bugs** in software
  - Learn Why & Examples
  - Learn how to properly handle
  - A lot of your future tasks will seems easy. With deep thoughts:
    - Your realize critical concerns or different trade-offs among different designs
- Year 2038 problem / Year 2000 problem
- Leap second (extra second)
- Time Zones and Daylight savings time (DST)

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."