Python - Date & Time

Created By :-

The Easylearn Acacdemy

9662512857

introduction

- In Python, **date, time and datetime** classes provides a number of function to deal with dates, times and time intervals.
- Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.
- Whenever you manipulate dates or time, you need to import datetime function.
- The datetime classes in Python are categorized into main 5 classes.
  - date – Manipulate just date (Month, day, year)
  - time – Time independent of the day (Hour, minute, second, microsecond)
  - datetime – Combination of time and date (Month, day, year, hour, second, microsecond)
  - timedelta— A duration of time used for manipulating dates
  - tzinfo— An abstract class for dealing with time zones

How to Use Date & DateTime Class

- Before you use date and time classes you have to import it using following statement
- from datetime import date
- from datetime import time
- from datetime import datetime
- These import statements are pre-defined pieces of functionality in the Python library that let you manipulates dates and times, without writing any code.
- let us see example

example

```
from datetime import date
current_date = date.today() #cd means currentdate
print ("Today's date " , current_date )
  # print today's date
print ("Today in day-month-year format" ,
  current_date.day,"-", current_date.month,"-",
  current_date.year);
print ("Day of the week is ",
  current_date.weekday());
```

- If Monday then it return 1 if Sunday then it returns 7

Example of now()

- Like Date Objects, we can also use **"DATETIME OBJECTS"** in Python.
- It gives **date** along with time in **hours, minutes, seconds and milliseconds.**

```
from datetime import datetime
today = datetime.now() #now() method return both date and
   time
print ("Today's date " , today)
```

- With "DATETIME OBJECT", you can also call time class.
- Suppose we want to print just the current time without the date.

```
JustTime = datetime.time(datetime.now());
print(JustTime)
```

strftime()

- The strftime() method returns a string representing date and time using date, time or datetime object.
- It convert date, time into various suitable format.
- let us see example

```
from datetime import datetime

now = datetime.now() # current date and time

year = now.strftime("%Y")
print("year:", year)

month = now.strftime("%m")
print("month:", month)

day = now.strftime("%d")
print("day:", day)

time = now.strftime("%H:%M:%S")
print("time:", time)

date_time = now.strftime("%m/%d/%Y, %H:%M:%S")
print("date and time:",date_time)
```

Example 2

```
from datetime import datetime
timestamp = 1528797322
date_time = datetime.fromtimestamp(timestamp)

print("Date time object:", date_time)

d = date_time.strftime("%m/%d/%Y, %H:%M:%S")
print("Output 2:", d)

d = date_time.strftime("%d %b, %Y")
print("Output 3:", d)

d = date_time.strftime("%d %B, %Y")
print("Output 4:", d)

d = date_time.strftime("%I%p")
print("Output 5:", d)
```

Timedelta Objects

- **timedelta objects is used to estimate the time for both future and the past.**
- **This function is not for printing out the time or date, but something to CALCULATE about the future or past.**
- Let's see an example to understand it better.

```
from datetime import date
from datetime import time
from datetime import datetime
from datetime import timedelta
# construct a basic timedelta and print it
print (timedelta(days=365, hours=8, minutes=15))
# print today's date
print ("today is: " + str(datetime.now()))
# print today's date one year from now
print ("one year from now it will be:" + str(datetime.now() +
   timedelta(days=365)))
# create a timedelta that uses more than one argument
# print (in one week and 4 days it will be " + str(datetime.now() +
   timedelta(weeks=1, days=4)))
# How many days until New Year's Day?
today = date.today()  # get todays date
Month= 1;
Days=1;
nyd = date(today.year,Month,Days)  # get New Year Day for the same year
# use date comparison to see if New Year Day has already gone for this
   year
if nyd < today:
    print ("New Year day is already went by %d days ago" % ((today -
   nyd).days))
```

Get current timestamp using Python

- the *time* module provides various time-related functions.
- The function time, return the time in seconds since the epoch as a floating point number.
- epoch is defined as the point where the time starts and is platform dependent.
- # using time module
- `import time`
- `# ts stores the time in seconds`
- `ts = time.time()`
- `# print the current timestamp`
- `print(ts)`

Convert date to timestamp in Python

- The Most Common way we use to store dates and times into a Database is in the form of a timestamp.
- When we receive a date and time in form of a string before storing it into a database, we convert that date and time string into a timestamp.
- Sample code

```
import time
import datetime
string = "12/07/1985"
print(time.mktime(datetime.strptime(string,"%d/%m/%Y").ti
  metuple()))
```