Python has a module named datetime to work with dates and times.

Let's create a few simple programs related to date and time before we dig deeper.

In [ ]:

```
# Example 1: Get Current Date and Time

import datetime

datetime_object = datetime.datetime.now()
print(datetime_object)
```

In [ ]:

```
# Example 2: Get Current Date

import datetime

date_object = datetime.date.today()
print(date_object)
```

In [ ]:

```
# What's inside datetime?
import datetime

print(dir(datetime))
```

datetime.date Class

- You can instantiate date objects from the date class. A date object represents a date (year, month and day).

In [ ]:

```
# Example 3: Date object to represent a date

import datetime

d = datetime.date(2019, 4, 13)
print(d)
```

In [ ]:

```
from datetime import date

a = date(2019, 4, 13)
print(a)
```

In [ ]:

```
# Example 4: Get current date

from datetime import date

today = date.today()

print("Current date =", today)
```

In [ ]:

```
# Example 6: Print today's year, month and day


from datetime import date

# date object of today's date
today = date.today()

print("Current year:", today.year)
print("Current month:", today.month)
print("Current day:", today.day)
```

datetime.time

- A time object instantiated from the time class represents the local time.

In [ ]:

```
from datetime import time

# time(hour = 0, minute = 0, second = 0)
a = time()
print("a =", a)

# time(hour, minute and second)
b = time(11, 34, 56)
print("b =", b)

# time(hour, minute and second)
c = time(hour = 11, minute = 34, second = 56)
print("c =", c)

# time(hour, minute, second, microsecond)
d = time(11, 34, 56, 234566)
print("d =", d)
```

In [ ]:

```
# Example 8: Print hour, minute, second and microsecond


from datetime import time

a = time(11, 34, 56)

print("hour =", a.hour)
print("minute =", a.minute)
print("second =", a.second)
print("microsecond =", a.microsecond)
```

datetime.datetime

- The datetime module has a class named dateclass that can contain information from both date and time objects.

In [ ]:

```
# Python datetime object

from datetime import datetime

#datetime(year, month, day)
a = datetime(2018, 11, 28)
print(a)

# datetime(year, month, day, hour, minute, second, microsecond)
b = datetime(2017, 11, 28, 23, 55, 59, 342380)
print(b)
```

In [ ]:

```
# Example 10: Print year, month, hour, minute and timestamp


from datetime import datetime

a = datetime(2017, 11, 28, 23, 55, 59, 342380)
print("year =", a.year)
print("month =", a.month)
print("hour =", a.hour)
print("minute =", a.minute)
print("timestamp =", a.timestamp())
```

datetime.timedelta

- A timedelta object represents the difference between two dates or times.

In [ ]:

```
# Difference between two dates and times

from datetime import datetime, date

t1 = date(year = 2018, month = 7, day = 12)
```

```
t2 = date(year = 2017, month = 12, day = 23)
t3 = t1 - t2
print("t3 =", t3)

t4 = datetime(year = 2018, month = 7, day = 12, hour = 7, minute = 9, second = 33)
t5 = datetime(year = 2019, month = 6, day = 10, hour = 5, minute = 55, second = 13)
t6 = t4 - t5
print("t6 =", t6)

print("type of t3 =", type(t3))
print("type of t6 =", type(t6))
```

In [ ]:

```
# Example 12: Difference between two timedelta objects

from datetime import timedelta

t1 = timedelta(weeks = 2, days = 5, hours = 1, seconds = 33)
t2 = timedelta(days = 4, hours = 11, minutes = 4, seconds = 54)
t3 = t1 - t2

print("t3 =", t3)
```

In [ ]:

```
# Printing negative timedelta object

from datetime import timedelta

t1 = timedelta(seconds = 33)
t2 = timedelta(seconds = 54)
t3 = t1 - t2

print("t3 =", t3)
print("t3 =", abs(t3))
```

In [ ]:

```
# Time duration in seconds
# You can get the total number of seconds in a timedelta object using total_seconds() method.

from datetime import timedelta

t = timedelta(days = 5, hours = 1, seconds = 33, microseconds = 233423)
print("total seconds =", t.total_seconds())
```

Python format datetime

- Python strftime() - datetime object to string
- The strftime() method is defined under classes date, datetime and time.
- The method creates a formatted string from a given date, datetime or time object.

In [ ]:

```
# Format date using strftime()

from datetime import datetime

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

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

s1 = now.strftime("%m/%d/%Y, %H:%M:%S")
# mm/dd/YY H:M:S format
print("s1:", s1)

s2 = now.strftime("%d/%m/%Y, %H:%M:%S")
# dd/mm/YY H:M:S format
print("s2:", s2)
```

Python strptime() - string to datetime

- The strptime() method creates a datetime object from a given string (representing date and time).

In [ ]:

```
from datetime import datetime

date_string = "21 June, 2018"
print("date_string =", date_string)

date_object = datetime.strptime(date_string, "%d %B, %Y")
print("date_object =", date_object)
```

Handling timezone in Python

- Suppose, you are working on a project and need to display date and time based on their timezone.
- Rather than trying to handle timezone yourself, we suggest you to use a third-party pytZ module.

In [ ]:

```
from datetime import datetime
import pytz

local = datetime.now()
print("Local:", local.strftime("%m/%d/%Y, %H:%M:%S"))


tz_NY = pytz.timezone('America/New_York')
datetime_NY = datetime.now(tz_NY)
print("NY:", datetime_NY.strftime("%m/%d/%Y, %H:%M:%S"))

tz_London = pytz.timezone('Europe/London')
datetime_London = datetime.now(tz_London)
print("London:", datetime_London.strftime("%m/%d/%Y, %H:%M:%S"))
```

datetime to string using strftime()

- The program below converts a datetime object containing current date and time to different string formats.

In [ ]:

```
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)
```

In [ ]:

```
# Creating string from a timestamp

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)
```

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

In [ ]:

```python
# Locale's appropriate date and time

from datetime import datetime

timestamp = 1528797322
date_time = datetime.fromtimestamp(timestamp)

d = date_time.strftime("%c")
print("Output 1:", d)

d = date_time.strftime("%x")
print("Output 2:", d)

d = date_time.strftime("%X")
print("Output 3:", d)
```

In [ ]:

```python
# Example 1: Python get today's date
from datetime import date

today = date.today()
print("Today's date:", today)
```

In [ ]:

```python
# Example 2: Current date in different formats
from datetime import date

today = date.today()

# dd/mm/YY
d1 = today.strftime("%d/%m/%Y")
print("d1 =", d1)

# Textual month, day and year
d2 = today.strftime("%B %d, %Y")
print("d2 =", d2)

# mm/dd/y
d3 = today.strftime("%m/%d/%y")
print("d3 =", d3)

# Month abbreviation, day and year
d4 = today.strftime("%b-%d-%Y")
print("d4 =", d4)
```

In [ ]:

```python
# Get the current date and time
from datetime import datetime

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

print("now =", now)

# dd/mm/YY H:M:S
dt_string = now.strftime("%d/%m/%Y %H:%M:%S")
print("date and time =", dt_string)
```

In [ ]:

```python
# The sleep() function suspends (delays) execution of the current thread for the given number of seconds

import time

print("This is printed immediately.")
time.sleep(2.4)
print("This is printed after 2.4 seconds.")
```

In [ ]:

```
# Python create a digital clock
import time

while True:
  localtime = time.localtime()
  result = time.strftime("%H:%M:%S %p", localtime)
  print(result)
  time.sleep(1)
```

In [ ]:

```
# Python create a digital clock
import time

while True:
  localtime = time.localtime()
  result = time.strftime("%H:%M:%S %p", localtime)
  print(result)
  time.sleep(1)
```

In [ ]:

```
# Python create a digital clock
import time

while True:
  localtime = time.localtime()
  result = time.strftime("%H:%M:%S %p", localtime)
  print(result)
  time.sleep(1)
```

In [ ]:

```
import time

while True:
  localtime = time.localtime()
  result = time.strftime("%H:%M:%S %p", localtime)
  print(result, end="", flush=True)
  print("\r", end="", flush=True)
  time.sleep(1)
```