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 [1]:
```
# Example 1: Get Current Date

import datetime

datetime_object = datetime.da
print(datetime_object)
```
```
2021-02-25 17:39:07.128086
```

In [2]:
```
# Example 2: Get Current Date

import datetime

date_object = datetime.date.t
print(date_object)
```
```
2021-02-25
```

In [3]:
```
# What's inside datetime?
import datetime

print(dir(datetime))
```
```
['MAXYEAR', 'MINYEAR', '__bui
ltins__', '__cached__', '__do
c__', '__file__', '__loader__
_', '__name__', '__package__
_', '__spec__', 'date', 'date
time', 'datetime_CAPI', 'sy
s', 'time', 'timedelta', 'tim
ezone', 'tzinfo']
```

datetime.date Class

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

In [4]:
```
# Example 3: Date object to r

import datetime

d = datetime.date(2019, 4, 13
print(d)
```
```
2019-04-13
```

In [5]:
```
from datetime import date

a = date(2019, 4, 13)
print(a)
```
```
2019-04-13
```

In [6]:
```
# Example 4: Get current date

from datetime import date

today = date.today()

print("Current date =", today
```
```
Current date = 2021-02-25
```

In [7]:
```
# Example 6: Print today's ye


from datetime import date

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

print("Current year:", today.
print("Current month:", today
print("Current day:", today.d
```
```
Current year: 2021
Current month: 2
Current day: 25
```

datetime.time

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

In [8]:
```
from datetime import time

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

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

# time(hour, minute and secon
c = time(hour = 11, minute =
print("c =", c)

# time(hour, minute, second,
d = time(11, 34, 56, 234566)
print("d =", d)
```
```
a = 00:00:00
b = 11:34:56
c = 11:34:56
d = 11:34:56.234566
```

In [9]:
```
# Example 8: Print hour, minu


from datetime import time

a = time(11, 34, 56)

print("hour =", a.hour)
print("minute =", a.minute)
print("second =", a.second)
print("microsecond =", a.micr
```
```
hour = 11
minute = 34
second = 56
microsecond = 0
```

datetime.datetime

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

In [10]:
```
# Python datetime object

from datetime import datetime

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

# datetime(year, month, day,
b = datetime(2017, 11, 28, 23
print(b)
```
```
2018-11-28 00:00:00
2017-11-28 23:55:59.342380
```

In [11]:
```
# Example 10: Print year, mon


from datetime import datetime

a = datetime(2017, 11, 28, 23
print("year =", a.year)
print("month =", a.month)
print("hour =", a.hour)
print("minute =", a.minute)
print("timestamp =", a.timest
```
```
year = 2017
month = 11
hour = 23
minute = 55
timestamp = 1511893559.34238
```

datetime.timedelta

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

In [12]:
```
# Difference between two date

from datetime import datetime

t1 = date(year = 2018, month
t2 = date(year = 2017, month
t3 = t1 - t2
print("t3 =", t3)

t4 = datetime(year = 2018, mo
t5 = datetime(year = 2019, mo
t6 = t4 - t5
print("t6 =", t6)

print("type of t3 =", type(t3
print("type of t6 =", type(t6
```
```
t3 = 201 days, 0:00:00
t6 = -333 days, 1:14:20
type of t3 = <class 'datetim
e.timedelta'>
type of t6 = <class 'datetim
e.timedelta'>
```

In [13]:
```
# Example 12: Difference betw

from datetime import timedelt

t1 = timedelta(weeks = 2, day
t2 = timedelta(days = 4, hour
t3 = t1 - t2

print("t3 =", t3)
```
```
t3 = 14 days, 13:55:39
```

In [14]:
```
# Printing negative timedelta

from datetime import timedelt

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

print("t3 =", t3)
print("t3 =", abs(t3))
```
```
t3 = -1 day, 23:59:39
t3 = 0:00:21
```

In [15]:
```
# Time duration in seconds
# You can get the total numbe

from datetime import timedelt

t = timedelta(days = 5, hours
print("total seconds =", t.to
```
```
total seconds = 435633.233423
```

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 [16]:
```
# 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,
# mm/dd/YY H:M:S format
print("s1:", s1)

s2 = now.strftime("%d/%m/%Y,
# dd/mm/YY H:M:S format
print("s2:", s2)
```
```
time: 17:39:13
s1: 02/25/2021, 17:39:13
s2: 25/02/2021, 17:39:13
```

Python strptime() - string to datetime

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

In [17]:
```
from datetime import datetime

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

date_object = datetime.strpti
print("date_object =", date_o
```
```
date_string = 21 June, 2018
date_object = 2018-06-21 00:0
0:00
```

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 [18]:
```
from datetime import datetime
import pytz

local = datetime.now()
print("Local:", local.strftim

tz_NY = pytz.timezone('Americ
datetime_NY = datetime.now(tz
print("NY:", datetime_NY.strf

tz_London = pytz.timezone('Eu
datetime_London = datetime.no
print("London:", datetime_Lon
```
```
Local: 02/25/2021, 17:39:14
NY: 02/25/2021, 07:09:15
London: 02/25/2021, 12:09:15
```

datetime to string using strftime()

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

In [19]:
```
from datetime import datetime

now = datetime.now() # curren

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/
print("date and time:",date_t
```
```
year: 2021
month: 02
day: 25
time: 17:39:15
date and time: 02/25/2021, 1
7:39:15
```

In [20]:
```
# Creating string from a time

from datetime import datetime

timestamp = 1528797322
date_time = datetime.fromtime

print("Date time object:", da

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

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

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

d = date_time.strftime("%I%p"
print("Output 5:", d)
```
```
Date time object: 2018-06-12
15:25:22
Output 2: 06/12/2018, 15:25:2
2
Output 3: 12 Jun, 2018
Output 4: 12 June, 2018
Output 5: 03PM
```

In [21]:
```
# Locale's appropriate date a

from datetime import datetime

timestamp = 1528797322
date_time = datetime.fromtime

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)
```
```
Output 1: Tue Jun 12 15:25:22
2018
Output 2: 06/12/18
Output 3: 15:25:22
```

In [22]:
```
# Example 1: Python get today
from datetime import date

today = date.today()
print("Today's date:", today)
```
```
Today's date: 2021-02-25
```

In [23]:
```
# Example 2: Current date in
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, %
print("d2 =", d2)

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

# Month abbreviation, day and
d4 = today.strftime("%b-%d-%Y
print("d4 =", d4)
```
```
d1 = 25/02/2021
d2 = February 25, 2021
d3 = 02/25/21
d4 = Feb-25-2021
```

In [24]:
```
# Get the current date and ti
from datetime import datetime

# datetime object containing
now = datetime.now()

print("now =", now)

# dd/mm/YY H:M:S
dt_string = now.strftime("%d/
print("date and time =", dt_s
```
```
now = 2021-02-25 17:39:16.815
455
date and time = 25/02/2021 1
7:39:16
```

In [25]:
```
# The sleep() function suspen

import time

print("This is printed immedi
time.sleep(2.4)
print("This is printed after
```
```
This is printed immediately.
This is printed after 2.4 sec
onds.
```

In [26]:
```
# Python create a digital clo
import time

while True:
  localtime = time.localtime(
  result = time.strftime("%H:
  print(result)
  time.sleep(1)
```
```
17:39:19 PM
17:39:20 PM
17:39:21 PM
17:39:22 PM
17:39:23 PM
17:39:24 PM
17:39:25 PM
17:39:26 PM
17:39:27 PM
17:39:28 PM
17:39:29 PM
17:39:30 PM
17:39:31 PM
17:39:32 PM
17:39:33 PM
17:39:34 PM
---------------------------------------------------------
KeyboardInterrupt                Traceback (most recent call l
ast)
<ipython-input-26-42f3b0c3762
a> in <module>
      6   result = time.strft
ime("%H:%M:%S %p", localtime)
      7   print(result)
----> 8   time.sleep(1)

KeyboardInterrupt:
```

In [27]:
```
# Python create a digital clo
import time

while True:
  localtime = time.localtime(
  result = time.strftime("%H:
  print(result)
  time.sleep(1)
```
```
17:39:48 PM
17:39:49 PM
17:39:50 PM
17:39:51 PM
17:39:52 PM
17:39:53 PM
17:39:54 PM
17:39:55 PM
17:39:56 PM
17:39:57 PM
---------------------------------------------------------
KeyboardInterrupt                Traceback (most recent call l
ast)
<ipython-input-27-42f3b0c3762
a> in <module>
      6   result = time.strft
ime("%H:%M:%S %p", localtime)
      7   print(result)
----> 8   time.sleep(1)

KeyboardInterrupt:
```

In [28]:
```
# Python create a digital clo
import time

while True:
  localtime = time.localtime(
  result = time.strftime("%H:
  print(result)
  time.sleep(1)
```
```
17:40:01 PM
17:40:02 PM
17:40:03 PM
17:40:04 PM
17:40:05 PM
17:40:06 PM
17:40:07 PM
---------------------------------------------------------
KeyboardInterrupt                Traceback (most recent call l
ast)
<ipython-input-28-42f3b0c3762
a> in <module>
      6   result = time.strft
ime("%H:%M:%S %p", localtime)
      7   print(result)
----> 8   time.sleep(1)

KeyboardInterrupt:
```

In [ ]:
```
import time

while True:
  localtime = time.localtime(
  result = time.strftime("%H:
  print(result, end="", flush
  print("\r", end="", flush=T
  time.sleep(1)
```
```
17:40:28 PM
```

In [ ]: