# **Working With Dates & Times**

## Datetime

Python has a built-in **datetime** module which provides convenient objects to work with dates and times.

#### **Code**

import datetime

## Datetime classes

Commonly used **classes** in the datetime module are:

* date class
* time class
* datetime class
* timedelta class

## Working with 'date' class

### Representing Date

A date object can be used to represent any valid **date** (year, month and day).

#### **Code**

import datetime

date\_object = datetime.date(2019, 4, 13)

print(date\_object)

#### **Output**

2019-04-1

### Date Object

#### **Code**

from datetime import date

date\_obj = date(2022, 2, 31)

print(date\_obj)

#### **Output**

ValueError: day is out of range

### Today’s Date

Class method

today() returns a date object with **today’s date**.

#### **Code**

import datetime

date\_object = datetime.date.today()

print(date\_object)

#### **Output**

2021-02-0

### Attributes of Date Object

#### **Code**

from datetime import date

date\_object = date(2019, 4, 13)

print(date\_object.year)

print(date\_object.month)

print(date\_object.day)

#### **Output**

2019

4

13

## Working with ‘time’ Class

### Representing Time

A time object can be used to represent any valid **time** (hours, minutes and seconds).

#### **Code**

from datetime import time

time\_object = time(11, 34, 56)

print(time\_object)

#### **Output**

11:34:5

### Attributes of Time Object

#### **Code**

from datetime import time

time\_object = time(11, 34, 56)

print(time\_object)

print(time\_object.hour)

print(time\_object.minute)

print(time\_object.second)

from datetime import time

#### **Output**

11:34:56

11

34

56

## Working with ‘datetime’ Class

### Datetime

The datetime class represents a valid **date and time** together.

*Example - 1*

#### **Code**

from datetime import datetime

date\_time\_obj = datetime(2018, 11, 28, 10, 15, 26)

print(date\_time\_obj.year)

print(date\_time\_obj.month)

print(date\_time\_obj.hour)

print(date\_time\_obj.minute)

#### **Output**

2018

11

10

15

*Example - 2*

It gives the current date and time

#### **Code**

import datetime

datetime\_object = datetime.datetime.now()

print(datetime\_object)

#### **Output**

2021-02-05 09:26:08.

### DateTime object

#### **Code**

from datetime import datetime

date\_time\_obj = datetime(2018, 11, 28)

print(date\_time\_obj)

#### **Output**

2018-11-28 00:00:00

## Formatting Datetime

The datetime classes have

strftime(format) method to format the datetime into any required format like

* mm/dd/yyyy
* dd-mm-yyyy

| Format Specifier | Meaning | Example |
| --- | --- | --- |
| %y | Year without century as a zero-padded decimal number | 19, 20, ... |
| %Y | Year with century as a decimal number | 2019, 2020, ... |
| %b | Month as abbreviated name | Jan, Feb, ... |
| %B | Month as full name | January, February |
| %m | Month as a zero-padded decimal number | 01, 02, …, 12 |
| %d | Day of the month as a zero-padded decimal number | 01, 02, …, 31 |
| %a | Weekday as abbreviated name | Sun, Mon, ... |
| %A | Weekday as full name | Sunday, Monday, ... |
| %H | Hour (24-hour clock) as a zero-padded decimal number | 00, 01, …, 23 |
| %I | Hour (12-hour clock) as a zero-padded decimal number | 01, 02, …, 12 |
| %p | AM or PM | AM, PM |
| %M | Minute as a zero-padded decimal number | 00, 01, …, 59 |
| %S | Second as a zero-padded decimal number | 00, 01, …, 59 |

#### **Code**

from datetime import datetime

now = datetime.now()

formatted\_datetime\_1 = now.strftime("%d %b %Y %I:%M:%S %p")

print(formatted\_datetime\_1)

formatted\_datetime\_2 = now.strftime("%d/%m/%Y, %H:%M:%S")

print(formatted\_datetime\_2)

#### **Output**

05 Feb 2021 09:26:50 AM

05/02/2021, 09:26

## Parsing Datetime

The class method

strptime() creates a **datetime object** from a given string representing date and time.

#### **Code**

from datetime import datetime

date\_string = "28 November, 2018"

print(date\_string)

date\_object = datetime.strptime(date\_string, "%d %B, %Y")

print(date\_object)

#### **Output**

28 November, 2018

2018-11-28 00:0

## Working with ‘timedelta’ Class

Timedelta object represents **duration**.

### *Example 1*

#### **Code**

from datetime import timedelta

delta = timedelta(days=365, hours=4)

print(delta)

#### **Output**

365 days, 4:00:

### *Example 2*

#### **Code**

from datetime import timedelta, datetime

delta = timedelta(days=365)

current\_datetime = datetime.now()

print(current\_datetime)

next\_year\_datetime = current\_datetime + delta

print(next\_year\_datetime)

#### **Output**

2021-02-05 09:28:30.239095

2022-02-05 09:28:30.239095

### Calculating Time Difference

#### **Code**

import datetime

dt1 = datetime.datetime(2021, 2, 5)

dt2 = datetime.datetime(2022, 1, 1)

duration = dt2 - dt1

print(duration)

print(type(duration))

#### **Output**

330 days, 0:00:00

<class 'datetime.timedelta'>