Cheat Sheet

Working With Dates & Times

Datetime

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

Code

PYTHON

```
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

PYTHON

```
import datetime

date_object = datetime.date(2019, 4, 13)
```

```
4 print(date_object)
```

Output

```
2019-04-13
```

Date Object

Code

PYTHON

```
1 from datetime import date
2 date_obj = date(2022, 2, 31)
3 print(date_obj)
```

Output

```
ValueError: day is out of range for month
```

Today's Date

Class method

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

Code

PYTHON

```
1 import datetime
2
3 date_object = datetime.date.today()
4 print(date_object)
```

Output

```
2021-02-05
```

Attributes of Date Object

Code

PYTHON

```python
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

PYTHON

```python
from datetime import time

time_object = time(11, 34, 56)
print(time_object)
```

Output

```
11:34:56
```

Attributes of Time Object

Code

PYTHON

```python
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)
```

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

PYTHON

```python
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

PYTHON

```
import datetime

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

Output

```
2021-02-05 09:26:08.077473
```

DateTime object

Code

PYTHON

```
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

PYTHON

```python
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, 09:26:50
```

Parsing Datetime

The class method

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

Code

PYTHON

```
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 00:00:00
```

Working with ‘timedelta’ Class

Timedelta object represents **duration**.

Example 1

Code

PYTHON

```python
from datetime import timedelta

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

Output

```
365 days, 4:00:00
```

Example 2

Code

PYTHON

```python
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

PYTHON

```python
import datetime
```

```
2
3  dt1 = datetime.datetime(2021, 2, 5)
4  dt2 = datetime.datetime(2022, 1, 1)
5  duration = dt2 - dt1
6  print(duration)
7  print(type(duration))
```

Output

```
330 days, 0:00:00
<class 'datetime.timedelta'>
```

Submit Feedback