Python timestamp to datetime and vice-versa

It's pretty common to store date and time as a timestamp in a database. A Unix timestamp is the number of seconds between a particular date and January 1, 1970 at UTC.

We can simply use the `fromtimestamp()` method from the `datetime` module to get a date from a UNIX timestamp.

Python timestamp to datetime

```
from datetime import datetime

# timestamp is number of seconds since 1970-01-01
timestamp = 1545730073

# convert the timestamp to a datetime object in the local timezone
dt_object = datetime.fromtimestamp(timestamp)

# print the datetime object and its type
print("dt_object =", dt_object)
print("type(dt_object) =", type(dt_object))
```

Â

Output

```
dt_object = 2018-12-25 09:27:53
type(dt_object) = <class 'datetime.datetime'>
```

Here, we have imported the `datetime` class from the datetime module.

Then, we used the `datetime.fromtimestamp()` class method which returns the local date and time (datetime object). This object is stored in the *dt_object* variable.

Note: We can easily create a string representing date and time from a `datetime` object using strftime() method.

Python datetime to timestamp

In Python, we can get timestamp from a datetime object using the `datetime.timestamp()` method. For example,

```
from datetime import datetime

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

# convert from datetime to timestamp
ts = datetime.timestamp(now)

print("Timestamp =", ts)
```

Output

```
Timestamp = 1672138646.118119
```

Here, the `datetime.timestamp()` method takes a datetime object as an argument and returns a Unix timestamp.

Also Read:

- Python Program to Convert String to Datetime

Table of Contents