Python Date and time

Python provides the **datetime** module work with real dates and times.

Get Current DateTime in Python

1 datatime.now()

```
import datetime as dt

d=dt.datetime.now()

print('data=',d)
```

current date time format

```
YYYY-MM-DD HH:MM:SS.MS
```

various formats.

- Use the `date()` function to get the date in `yyyy-mm-dd` format
- Use the `time()` function to get the time in the `hours:minutes:seconds.microseconds` format.

```
import datetime as dt

d=dt.datetime.now()

current_date=d.date()
print('data=',current_date)

current_time=d.time()
print('current_time=',current_time)
```

Python DateTime Format Using Strftime()

The strftime() method returns a string representing of a datetime object according to the format codes.

> example, you may need to represent a date numerically in format, like "**17-06-2021**". On the other hand, you want to convert dates in textual string format like "**Tuesday, 23 June 2021**

he format codes are standard directives for mentioning in which format you want to represent datetime. For example, the `%d-%m-%Y %H:%M:%S` codes convert date to `dd-mm-yyyy hh:mm:ss` format.

```
import datetime as dt

# current dateTime
d = dt.datetime.now()
print("current date time=",d)

# convert to date String
date = d.strftime("%d/%m/%Y")
print('Date String:', date)

# convert to time String
time = d.strftime("%H:%M:%S")
print('Time String:', time)

# year
year = d.strftime("%Y")
print('Year String:', year)

# Month
month = d.strftime("%m")
print('Month String:', month)

# Day
day = d.strftime("%d")
print('Day String:', day)
```

- `%d`: Returns the **day** of the month, from 1 to 31.
- `%m`: Returns the **month** of the year, from 1 to 12.
- `%Y`: Returns the year in four-digit format (**Year** with century). like, 2021.
- `%y`: Reurns year in two-digit format (**year** without century). like, 19, 20, 21
- `%A`: Returns the full name of the **weekday**. Like, Monday, Tuesday
- `%a`: Returns the short name of the **weekday** (First three character.). Like, Mon, Tue
- `%B`: Returns the full name of the **month**. Like, June, March
- `%b`: Returns the short name of the **month** (First three character.). Like, Mar, Jun
- `%H`: Returns the **hour**. from 01 to 23.
- `%I`: Returns the **hour** in 12-hours format. from 01 to 12.
- `%M`: Returns the **minute**, from 00 to 59.
- `%S`: Returns the **second**, from 00 to 59.
- `%f`: Return the **microseconds** from 000000 to 999999
- `%p`: Return time in **AM/PM** format
- `%c`: Returns a **locale's appropriate date and time** representation
- `%x`: Returns a locale's appropriate date representation
- `%X`: Returns a locale's appropriate time representation
- `%z`: Return the **UTC offset** in the form `±HHMM[SS[.ffffff]]` (empty string if the object is naive).
- `%Z`: Return the **Time zone name** (empty string if the object is naive).
- `%j`: Returns the day of the year from *01 to 366*
- `%w`: Returns weekday as a decimal number, where 0 is Sunday and 6 is Saturday.
- `%U`: Returns the week number of the year (Sunday as the first day of the week) from 00 to 53
- `%W`: Returns the week number of the year (Monday as the first day of the week) from 00 to 53

Represent time in 24-hours and 12-hours Format

- Use the `%H-%M-%S` format code to display time in **24-hours** format
- Use the `%I-%M-%S` format code to display time in **12-hours** format

```
import datetime as dt

x_time = dt.datetime.now()
print('Current Time:', x_time)

print("Time in 24 hours format:", x_time.strftime("%H-%M-%S"))
print("Time in 12 hours format:", x_time.strftime("%I-%M-%S"))
```

Represent Time in Microseconds Format

- Use the `%f` format code to represent time in **microsecond**
- Use the `%p` format code to represent time in **AM/PM** format