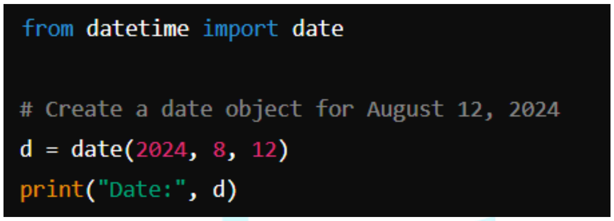
Task 4 - Date and Time

Introduction to the **datetime** Module

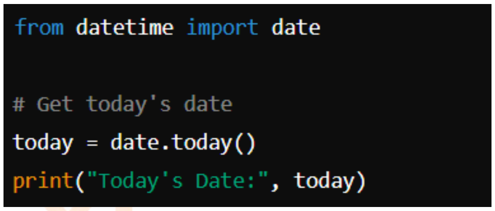
- The **datetime** module provides classes for manipulating dates and times.
- Commonly used classes include **date**, **time**, **datetime**, and **timedelta**

date Class

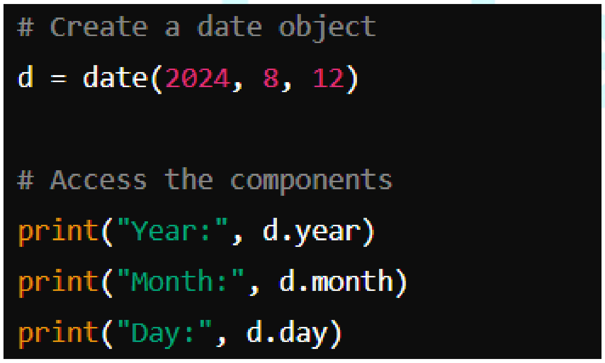
The **date** class represents a date in the calendar like (year, month, and day).



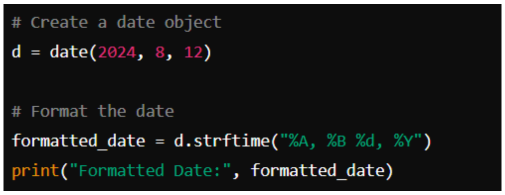
```
from datetime import date

# Create a date object for August 12, 2024
d = date(2024, 8, 12)
print("Date:", d)
```



```
from datetime import date

# Get today's date
today = date.today()
print("Today's Date:", today)
```



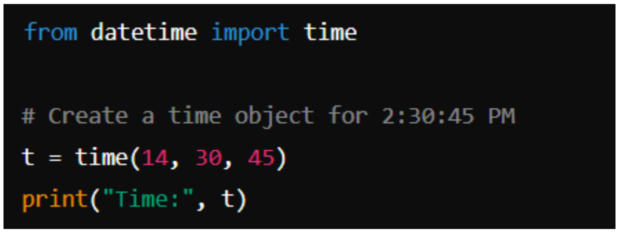
```
# Create a date object
d = date(2024, 8, 12)

# Access the components
print("Year:", d.year)
print("Month:", d.month)
print("Day:", d.day)
```



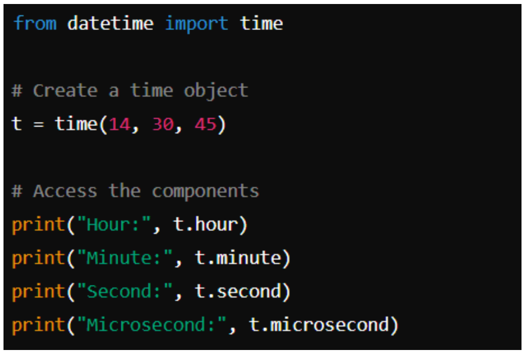
```
# Create a date object
d = date(2024, 8, 12)

# Format the date
formatted_date = d.strftime("%A, %B %d, %Y")
print("Formatted Date:", formatted_date)
```

time Class

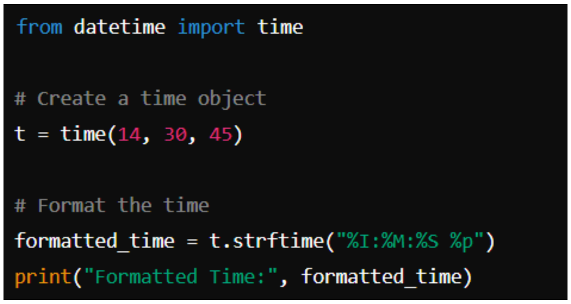
The time class represents a time of day independent of any particular day.



```
from datetime import time

# Create a time object for 2:30:45 PM
t = time(14, 30, 45)
print("Time:", t)
```



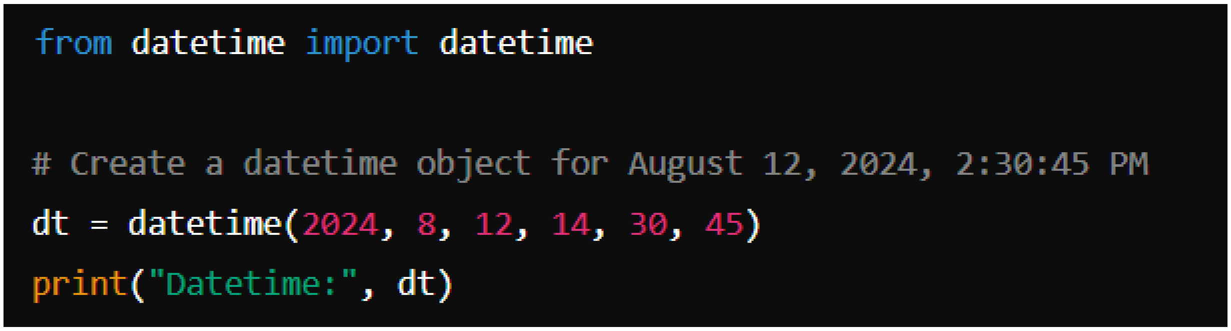
```
from datetime import time

# Create a time object
t = time(14, 30, 45)

# Access the components
print("Hour:", t.hour)
print("Minute:", t.minute)
print("Second:", t.second)
print("Microsecond:", t.microsecond)
```



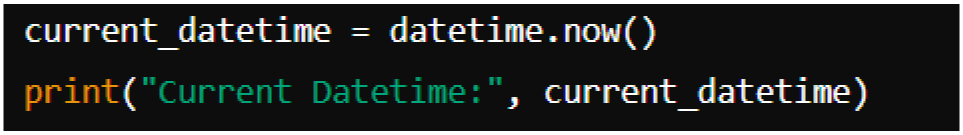
```
from datetime import time

# Create a time object
t = time(14, 30, 45)

# Format the time
formatted_time = t.strftime("%I:%M:%S %p")
print("Formatted Time:", formatted_time)
```

datetime Class

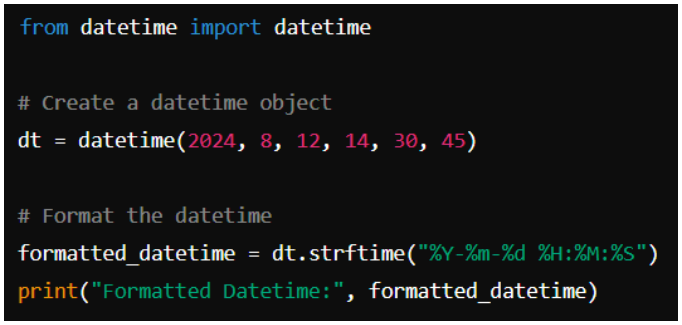
The **datetime** class is a combination of a date and a time.



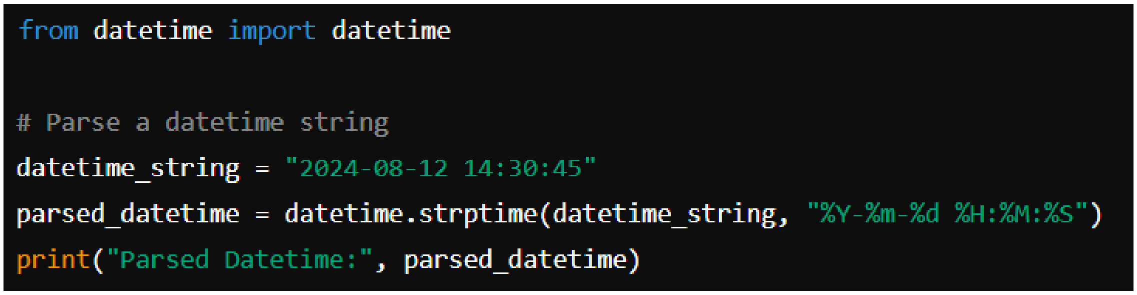
```
from datetime import datetime

# Create a datetime object for August 12, 2024, 2:30:45 PM
dt = datetime(2024, 8, 12, 14, 30, 45)
print("Datetime:", dt)
```



```
current_datetime = datetime.now()
print("Current Datetime:", current_datetime)
```



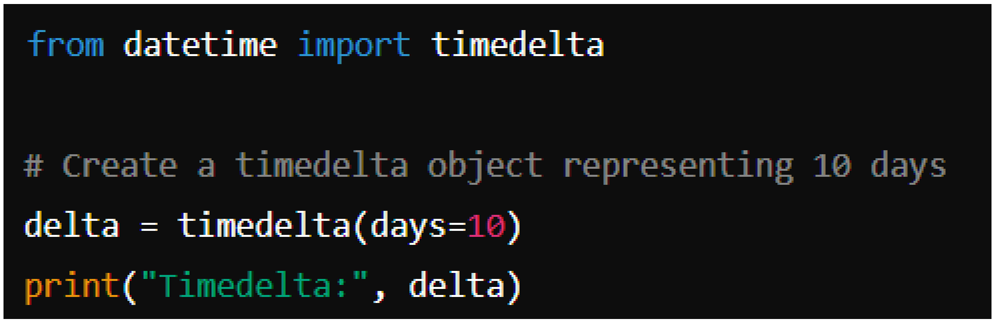
```
from datetime import datetime

# Create a datetime object
dt = datetime(2024, 8, 12, 14, 30, 45)

# Format the datetime
formatted_datetime = dt.strftime("%Y-%m-%d %H:%M:%S")
print("Formatted Datetime:", formatted_datetime)
```



```
from datetime import datetime

# Parse a datetime string
datetime_string = "2024-08-12 14:30:45"
parsed_datetime = datetime.strptime(datetime_string, "%Y-%m-%d %H:%M:%S")
print("Parsed Datetime:", parsed_datetime)
```

timedelta Class

The **timedelta** class represents a duration and can be used to perform arithmetic.



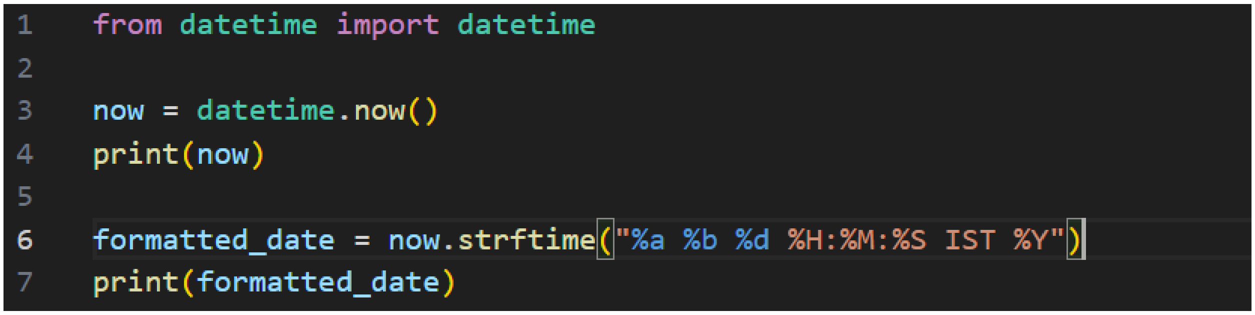
```
from datetime import timedelta

# Create a timedelta object representing 10 days
delta = timedelta(days=10)
print("Timedelta:", delta)
```

Format Codes

%a: Abbreviated weekday name (e.g., **Sun** for Sunday).
%A: Full weekday name (e.g., **Sunday**).
%w: Weekday as a decimal number, where Sunday is **0** and Saturday is **6** (e.g., **0** for Sunday).
%d: Day of the month as a zero-padded decimal number (e.g., **01**, **02**, ..., **31**).
%b: Abbreviated month name (e.g., **Jan** for January).
%B: Full month name (e.g., **January**).
%m: Month as a zero-padded decimal number (e.g., **01**, **02**, ..., **12**).
%y: Year without century as a zero-padded decimal number (e.g., **99** for 1999).
%Y: Year with century as a decimal number (e.g., **2017**).
%H: Hour (24-hour clock) as a zero-padded decimal number (e.g., **00, 01, ..., 23**).
%I: Hour (12-hour clock) as a zero-padded decimal number (e.g., **01, 02, ..., 12**).
%p: AM or PM designation (e.g., **AM, PM**).
%M: Minute as a zero-padded decimal number (e.g., **00, 01, ..., 59**).
%S: Second as a zero-padded decimal number (e.g., **00, 01, ..., 59**).
%f: Microsecond as a zero-padded decimal number (e.g., **000000, ..., 999999**).
%z: UTC offset in the form **+HHMM** or **-HHMM** (empty if the object is naive).
%Z: Time zone name (e.g., **EST**, **IST**, or empty string if the object is naive).
%j: Day of the year as a zero-padded decimal number (e.g., **001, 002, ..., 366**).
%U: Week number of the year (Sunday as the first day of the week) as a zero-padded decimal number (e.g., **00, 01, ..., 53**). All days in a new year preceding the first Sunday are considered to be in week 0.
%W: Week number of the year (Monday as the first day of the week) as a zero-padded decimal number (e.g., **00, 01, ..., 53**). All days in a new year preceding the first Monday are considered to be in week 0.
%c: Locale's appropriate date and time representation (e.g., **Tue Aug 09 21:37:00 2024**).
%x: Locale's appropriate date representation (e.g., **08/09/24** for **MM/DD/YY** in the U.S.).

%X: Locale’s appropriate time representation (e.g., **`21:37: 00`**).
%%: A literal **%** character.

Task: Write a python script to print the current date in the following format "Sun May 29 02:26:23 IST 2017"



```python
from datetime import datetime

now = datetime.now()
print(now)

formatted_date = now.strftime("%a %b %d %H:%M:%S IST %Y")
print(formatted_date)
```