**datetime Module in Python**

The datetime module in Python provides classes and functions to work with **dates, times, and time intervals**.

The datetime module in Python is used to **work with dates and times**. It allows you to get the current date/time, format it, perform date arithmetic, and more.

**1. Importing the Module**

import datetime

**2. Getting Current Date and Time**

now = datetime.datetime.now()

print(now) # Example: 2025-04-04 21:00:30.123456

To get only the **date** or **time**:

print(now.date()) # 2025-04-04

print(now.time()) # 21:00:30.123456

**3. Get Today’s Date**

today = datetime.date.today()

print(today) # Example: 2025-04-04

**4. Create a Specific Date or Time**

my\_date = datetime.date(2025, 12, 25)

print(my\_date) # 2025-12-25

my\_time = datetime.time(14, 30, 0)

print(my\_time) # 14:30:00

**5. Date Arithmetic (Add/Subtract Days)**

Use timedelta for date calculations:

from datetime import timedelta

today = datetime.date.today()

tomorrow = today + timedelta(days=1)

yesterday = today - timedelta(days=1)

print("Today:", today)

print("Tomorrow:", tomorrow)

print("Yesterday:", yesterday)

**Date Arithmetic (Timedelta)**

**➤ Creating a timedelta**

from datetime import timedelta

delta = timedelta(days=5, hours=3)

print(delta) # 5 days, 3:00:00

**➤ Adding/Subtracting dates**

now = datetime.now()

future = now + timedelta(days=7)

past = now - timedelta(days=30)

print(future) # 7 days from now

print(past) # 30 days ago

**6. Formatting Date and Time (strftime)**

Convert a datetime object to a string format:

from datetime import timedelta

now = datetime.datetime.now()

formatted = now.strftime("%Y-%m-%d %H:%M:%S")

print(formatted) # 2025-04-04 21:03:00

print(now.strftime("%Y-%m-%d")) # 2025-04-04

print(now.strftime("%d/%m/%Y")) # 04/04/2025

print(now.strftime("%I:%M %p")) # 02:23 PM

print(now.strftime("%A, %B %d")) # Friday, April 04

| **Format Code** | **Meaning** | **Example** |
| --- | --- | --- |
| %Y | Year (4 digits) | 2025 |
| %m | Month (01 to 12) | 04 |
| %d | Day of the month | 04 |
| %H | Hour (24-hour) | 14 |
| %I | Hour (12-hour) | 02 |
| %p | AM or PM | PM |
| %A | Full weekday name | Friday |
| %B | Full month name | April |

**7. Parsing Date String (strptime)**

Convert a string into a datetime object:

from datetime import datetime

date\_string = "25/12/2023 10:30 AM"

parsed\_date = datetime.strptime(date\_string, "%d/%m/%Y %I:%M %p")

print(parsed\_date) # Output: 2023-12-25 10:30:00

**8. Get Day, Month, Year, etc.**

now = datetime.datetime.now()

print(now.year) # 2025

print(now.month) # 4

print(now.day) # 4

print(now.hour) # 21

print(now.minute) # 0

**9. Working with date and time classes**

**➤ date object (only date)**

from datetime import date

today = date.today()

print(today) # Output: 2025-04-04

print(today.year) # 2025

**➤ time object (only time)**

from datetime import time

t = time(10, 45, 30)

print(t) # Output: 10:45:30

print(t.hour) # 10

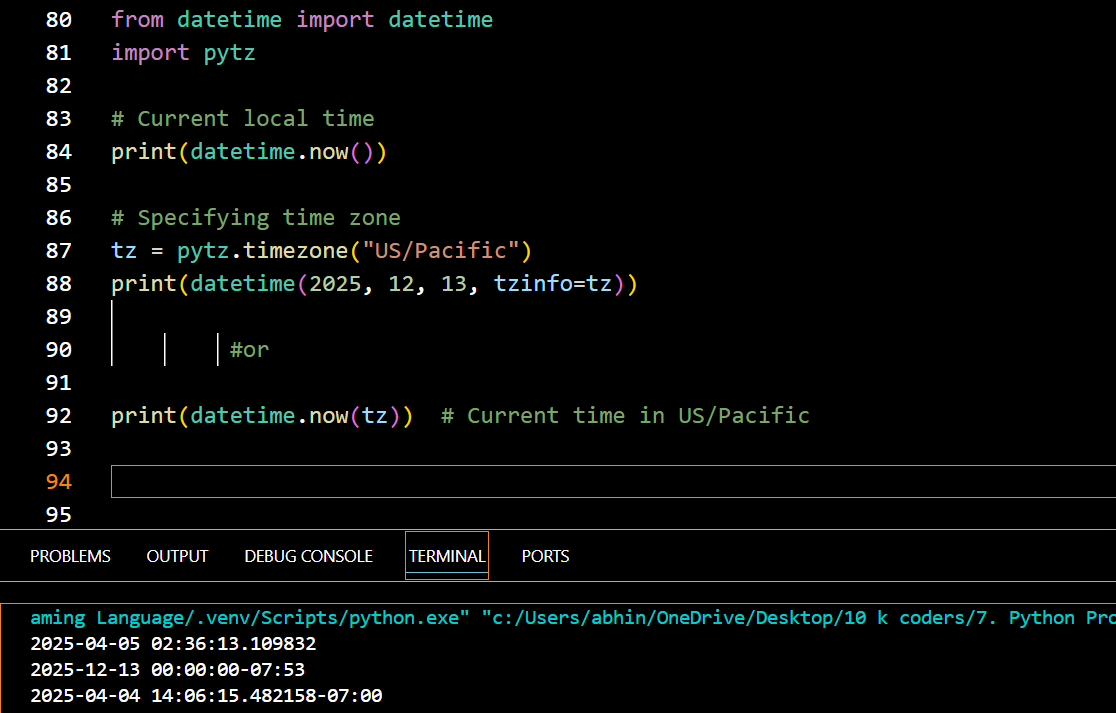
**10. Get Day of the Week**

today = datetime.today()

print(today.strftime("%A")) # Output: Friday

# Using weekday() method

print(today.weekday()) # Output: 4 (Monday=0, Sunday=6)



**✅ Summary Table**

| **Function / Class** | **Description** |
| --- | --- |
| datetime.now() | Current date and time |
| datetime.today() | Today's date |
| datetime.strptime() | Parse string to datetime |
| datetime.strftime() | Format datetime to string |
| datetime(year, m, d, h, m) | Create custom datetime object |
| date.today() | Returns current date |
| timedelta(days=, hours=) | Create time difference |
| datetime + timedelta | Future date |
| datetime - timedelta | Past date |

**💡 Real-World Example: Countdown Timer**

from datetime import datetime, timedelta

event\_date = datetime(2025, 12, 31, 23, 59)

now = datetime.now()

remaining = event\_date - now

print(f"Time remaining for New Year: {remaining.days} days and {remaining.seconds // 3600} hours")

Would you like a **mini-project using datetime** like a digital clock, birthday countdown, or age calculator? ⏳