

Importing the `datetime` Module

Before working with dates and times, you need to import the `datetime` module:

```
import datetime
```

■ Working With Date, Year, Month, Day, Date, Time, WeekDay, Hour, Minute, Second

```
import datetime

current_datetime = datetime.datetime.now() # Current Date
print(current_datetime.year) # Current Year
print(current_datetime.month) # Current Month
print(current_datetime.day) # Current Day
print(current_datetime.date()) # Current Date
print(current_datetime.time()) # Current Time
print(current_datetime.weekday()) # Current weekday
print(current_datetime.hour) # Current Hour
print(current_datetime.minute) # Current Minute
print(current_datetime.second) # Current second
print(current_datetime.microsecond) # Current microsecond
```

■ Current Date Time With Expected Format

```
import datetime
current_datetime = datetime.datetime.now()
formatted_datetime = current_datetime.strftime("%d/%m/%Y %H:%M:%S")
print("Current Date and Time:", formatted_datetime)
```

■ Handling Time Zones

Python's `datetime` module can handle time zones with the help of the `pytz` library. Here's how to work with time zones

```
import pytz
import datetime

# Set a timezone
timezone = pytz.timezone("America/New_York")
ny_time = datetime.datetime.now(timezone)
print("New York Time:", ny_time)
```

```
# Convert to another timezone
london_timezone = pytz.timezone("Europe/London")
london_time = ny_time.astimezone(london_timezone)
print("London Time:", london_time)
```

■ Calculating Time Differences

```
import datetime

# Subtracting two dates
date1 = datetime.datetime(2024, 9, 1)
date2 = datetime.datetime(2023, 9, 1)
difference = date1 - date2
print("Difference:", difference)

# Adding 10 days to a date
new_date = date1 + datetime.timedelta(days=10)
print("New Date:", new_date)
```