1. Common Date and Time Format Codes

The strftime method allows you to format datetime objects into strings using the following format codes:

Date Components:

- %Y - **Year** with century (e.g., 2024)
- %y - **Year** without century (e.g., 24)
- %m - **Month** (01-12)
- %d - **Day of the month** (01-31)
- %j - **Day of the year** (001-366)
- %A - **Full weekday name** (e.g., Monday)
- %a - **Abbreviated weekday name** (e.g., Mon)
- %B - **Full month name** (e.g., January)
- %b - **Abbreviated month name** (e.g., Jan)
- %U - **Week number of the year** (00-53), Sunday as the first day of the week
- %W - **Week number of the year** (00-53), Monday as the first day of the week

Time Components:

- %H - **Hour (24-hour clock)** (00-23)
- %I - **Hour (12-hour clock)** (01-12)
- %M - **Minute** (00-59)
- %S - **Second** (00-59)
- %f - **Microsecond** (000000-999999)
- %p - **AM/PM** (e.g., AM, PM)
- %z - **UTC offset** in the form +HHMM or -HHMM (e.g., +0530)

Other Common Codes:

- %c - **Locale's date and time representation** (e.g., Sat Dec 21 10:00:00 2024)
- %x - **Locale's date representation** (e.g., 12/21/24)

- %X - **Locale's time representation** (e.g., 10:00:00)
- %G - **ISO 8601 year** (e.g., 2024)
- %g - **Last two digits of the ISO 8601 year** (e.g., 24)
- %Z - **Time zone name** (e.g., UTC, Asia/Kolkata)
- %T - **Equivalent to %H:%M:%S** (e.g., 10:00:00)

---

2. Useful Methods and Properties of datetime Objects

- **datetime.now()**: Returns the current local date and time.
- **datetime.utcnow()**: Returns the current UTC date and time.
- **datetime.today()**: Returns the current local date and time (same as datetime.now()).
- **datetime.fromtimestamp(timestamp)**: Converts a timestamp (seconds since the epoch) to a local datetime.
- **datetime.utcfromtimestamp(timestamp)**: Converts a timestamp to a UTC datetime.
- **datetime.strptime(date_string, format)**: Parses a string into a datetime object according to a specific format.
- **datetime.timestamp()**: Returns the timestamp (seconds since the epoch) of the datetime object.
- **datetime.date()**: Returns the date part of the datetime (e.g., 2024-12-21).
- **datetime.time()**: Returns the time part of the datetime (e.g., 10:00:00).
- **datetime.weekday()**: Returns the weekday as an integer (0 for Monday, 6 for Sunday).
- **datetime.isoweekday()**: Returns the weekday as an integer (1 for Monday, 7 for Sunday).
- **datetime.isoformat()**: Returns a string in ISO 8601 format (e.g., 2024-12-21T10:00:00).

---

3. Useful Functions for Date and Time Arithmetic

- **timedelta(days=0, seconds=0, microseconds=0, milliseconds=0, minutes=0, hours=0, weeks=0)**: A class that represents the difference between two datetime objects.
    - Example: timedelta(days=5) represents a 5-day difference.

- **datetime + timedelta**: Adds a timedelta to a datetime object (e.g., now + timedelta(days=1)).
- **datetime - timedelta**: Subtracts a timedelta from a datetime object (e.g., now - timedelta(days=1)).
- **datetime + datetime**: Raises a TypeError, since addition of two datetime objects is not supported directly.
- **datetime - datetime**: Returns a timedelta object representing the difference between two datetime objects.

---

4. Working with Time Zones

- **pytz.timezone('Region/City')**: Returns a time zone object.
    - Example: timezone = pytz.timezone('Asia/Kolkata')
- **datetime.astimezone(tz)**: Converts a datetime object to the given time zone.
- **datetime.localize(datetime)**: Localizes a naive datetime object to a time zone (using pytz).
- **datetime.normalize(datetime)**: Adjusts a time to account for Daylight Saving Time (DST).

---

5. Extracting Components of a datetime Object

- **datetime.year**: Extracts the year (e.g., 2024).
- **datetime.month**: Extracts the month (e.g., 12).
- **datetime.day**: Extracts the day of the month (e.g., 21).
- **datetime.hour**: Extracts the hour (e.g., 10).
- **datetime.minute**: Extracts the minute (e.g., 00).
- **datetime.second**: Extracts the second (e.g., 00).
- **datetime.microsecond**: Extracts the microsecond (e.g., 123456).