**Datetime module:**

We have three modules in python for date and time

* **Time**
* **Calendar**
* **Datetime as dt**

The **datetime** module in Python provides classes for manipulating dates and times in both simple and complex ways. It allows you to create, manipulate, and format dates and times easily. Here are some of the key components of the **datetime** module:

1. **datetime Class**:
   * The **datetime** class represents a specific date and time. It's a combination of a date (year, month, day) and a time (hour, minute, second, microsecond).
   * You can create **datetime** objects using the **datetime()** constructor and access its attributes which are:

* **year**
* **month**
* **day**
* **hour**
* **minute**
* **second** and
* **microsecond**.

1. **date Class**:
   * The **date** class represents a date without time information. It includes attributes such as **year**, **month**, and **day**.
   * You can create **date** objects using the **date()** constructor and perform date arithmetic and comparison operations.
2. **time Class**:
   * The **time** class represents a time without date information. It includes attributes such as **hour**, **minute**, **second**, and **microsecond**.
   * You can create **time** objects using the **time()** constructor.
3. **timedelta Class**:
   * The **timedelta** class represents the difference between two **datetime** objects.
   * You can create **timedelta** objects using the **-** operator between two **datetime** objects or using the **timedelta()** constructor directly.
   * We use timedelta.days for days
4. **strftime() and strptime() Methods**:
   * The **strftime()** method of **datetime** objects allows you to format a **datetime** object into a string representing the date and time according to a specified format.
   * Dt.datetime.strftime(date format) or dt.datetime.strftime(time format)
   * The **strptime()** function in the **datetime** module parses a string representing a time according to a specified format and returns a **datetime** object.
   * Dt.datetime.strptime(date , date format) or dt.datetime(time , time format)

**Important:**

strptime() strftime()

String date -------------------------------🡪 datetime ----------------------------------------🡪string fomat

For us to work for others to show

|  |  |
| --- | --- |
| * day in number | "%d" |
| * day in word | "%A", "%a" |
| * month in number | "%m" |
| * month in word | "%B", "%b" |
| * year in number | " %Y", "%y" |
| * 24 hour format | "%H" |
| * 12 hour format | "%I" |
| * am or pm | "%p" |
| * Minutes | "%M" |

**Dateetime in pandas:**

* To convert a series of string in data frame to datetime:

**pd.to\_datetime(file\_name [“column\_name”])**

* To extract month in our desired format

**dt.strftime("%B")**

* For frequency table we may use

**Series\_name.values\_counts()**

* For duration of time we can calculate the difference of two time in the different series

**Duration = “Start\_time “– “end\_time**”

* For shortest time

**Series\_name.min()**

* For longest time

**Series\_name.max()**