In [4]:
```
from datetime import datetime, date, time, timedelta
```

In [3]:
```
# current datetime

print(datetime.datetime.now())
```

```
2025-08-08 08:46:20.768325
```

In [5]:
```
print(datetime.now())
```

```
2025-08-08 08:47:03.197959
```

In [6]:
```
print(datetime.today())
```

```
2025-08-08 08:47:13.658087
```

In [ ]:
```
date- Year-month-day
time- hour-minute-second-microsecond
```

In [10]:
```
print(dir(datetime))
```

```
['__add__', '__class__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format
__', '__ge__', '__getattribute__', '__getstate__', '__gt__', '__hash__', '__init_
_', '__init_subclass__', '__le__', '__lt__', '__ne__', '__new__', '__radd__', '__
reduce__', '__reduce_ex__', '__replace__', '__repr__', '__rsub__', '__setattr__',
'__sizeof__', '__str__', '__sub__', '__subclasshook__', 'astimezone', 'combine',
'ctime', 'date', 'day', 'dst', 'fold', 'fromisocalendar', 'fromisoformat', 'fromo
rdinal', 'fromtimestamp', 'hour', 'isocalendar', 'isoformat', 'isoweekday', 'ma
x', 'microsecond', 'min', 'minute', 'month', 'now', 'replace', 'resolution', 'sec
ond', 'strftime', 'strptime', 'time', 'timestamp', 'timetuple', 'timetz', 'toda
y', 'toordinal', 'tzinfo', 'tzname', 'utcfromtimestamp', 'utcnow', 'utcoffset',
'utctimetuple', 'weekday', 'year']
```

In [11]:
```
now = datetime.now()
```

In [12]:
```
print(now)
```

```
2025-08-08 08:51:06.662194
```

In [21]:
```
# Year
print(now.year)
# month
print(now.month)
# day
print(now.day)
# hour
print(now.hour)
# minute
print(now.minute)
# second
print(now.second)
# microsecond
print(now.microsecond)
############
print("=" * 20)
# weekday
print(now.weekday())
```

```
2025
8
8
8
51
6
662194
====================
4
```

In [35]:
```
# Formating
formatted_date = now.strftime("%d-%m-%Y %H:%M:%S")
print(formatted_date)

formatted_date1 = now.strftime("%d/%m/%Y %H:%M:%S")
print(formatted_date1)

formatted_date1 = now.strftime("%m %B,%Y %H:%M:%S") # %B full month name
print(formatted_date1)

formatted_date1 = now.strftime("%m %b,%Y %H:%M:%S") # %b short month name
print(formatted_date1)
```

```
08-08-2025 08:51:06
08/08/2025 08:51:06
08 August,2025 08:51:06
08 Aug,2025 08:51:06
```

In [45]:
```
# parsing

date_string = "2025-7-10 10:30:20"
parse_date = datetime.strptime(date_string,"%Y-%m-%d %H:%M:%S")
print(parse_date)
print(type(parse_date))
```

```
2025-07-10 10:30:20
<class 'datetime.datetime'>
```

In [ ]:
```
# Timedelta
```

In [51]:
```
td1 = now + timedelta(days = 7)
print(td1)

td2 = now - timedelta(days = 7)
print(td2)
```

```
2025-08-15 08:51:06.662194
2025-08-01 08:51:06.662194
```

In [54]:
```
td3 = now - timedelta(weeks =2 ,days = 7, hours= 4, minutes= 30)
print(td3)
```

```
2025-07-18 04:21:06.662194
```

In [55]:
```
## time differece

now - td3
```

Out[55]:
```
datetime.timedelta(days=21, seconds=16200)
```

Time

In [56]:
```
import time
```

In [58]:
```
print(dir(time))
```

```
['_STRUCT_TM_ITEMS', '__doc__', '__loader__', '__name__', '__package__', '__spec_
_', 'altzone', 'asctime', 'ctime', 'daylight', 'get_clock_info', 'gmtime', 'local
time', 'mktime', 'monotonic', 'monotonic_ns', 'perf_counter', 'perf_counter_ns',
'process_time', 'process_time_ns', 'sleep', 'strftime', 'strptime', 'struct_tim
e', 'thread_time', 'thread_time_ns', 'time', 'time_ns', 'timezone', 'tzname']
```

In [59]:
```
# time
time.time()
```

Out[59]:
```
1754624671.5946977
```

In [61]:
```
# sleep
print("start time...")
time.sleep(5)
print("Completed")
```

```
start time...
Completed
```

In [62]:
```
time.ctime() # human date time readable format
```

Out[62]:
```
'Fri Aug  8 09:16:59 2025'
```

In [65]:
```
print(time.gmtime()) # utc
```

```
time.struct_time(tm_year=2025, tm_mon=8, tm_mday=8, tm_hour=3, tm_min=47, tm_sec=
55, tm_wday=4, tm_yday=220, tm_isdst=0)
```

In [66]:
```
print(time.localtime())
```

```
time.struct_time(tm_year=2025, tm_mon=8, tm_mday=8, tm_hour=9, tm_min=18, tm_sec=
11, tm_wday=4, tm_yday=220, tm_isdst=0)
```

In [70]:
```
start_time = time.time()
perf_counter = time.perf_counter()
start_process_time = time.process_time()

result = sum( i**2 for i in range(10000000))
print(result)
endtime = time.time() - start_time
end_perf_counter = time.perf_counter()- perf_counter
end_processtime = time.process_time() - start_process_time

print(endtime)
print(end_perf_counter)
print(end_processtime)
```

```
333333283333335000000
2.9227259159088135
2.9228016000124626
2.765625
```

In [ ]: