Chapter-17

KEEPING TIME, SCHEDULING TASKS, AND LAUNCHING PROGRAMS

1. What is the Unix epoch?

> The Unix epoch is a time reference commonly used in programming: 12 AM on January 1, 1970, Coordinated Universal Time (UTC).

2. What function returns the number of seconds since the Unix epoch?

> **time.time()**function

3. How can you pause your program for exactly 5 seconds?

> By using **time.sleep(5)**

4. What does the round() function return?

> **round()** rounds your number to the **nearest whole integer**

5. What is the difference between a datetime object and a timedelta object?

> **datetime** => this function returns a datetime object of the moment specified by the arguments. If hour, minute, or second arguments are not provided, they default to 0.

> **timedelta** => It represents the duration of time.

6. Using the datetime module, what day of the week was January 7, 2019?

In [2]:
```
import datetime
```

In [10]:
```
Day = datetime.datetime(2019,1,7)
Day.weekday()
```

Out[10]:
```
0
```

> ***Day of the week, '0' (Sunday) to '6' (Saturday). so Jan 7, 2019 is sunday***

7. Say you have a function named spam(). How can you call this function and run the code inside it in a separate thread?

> Thread1 = threading.Thread(target=spam)

> Thread1.start()

8. What should you do to avoid concurrency issues with multiple threads?

To avoid concurrency issues:

> Never let multiple threads read or write the same variables.

> When you create a new Thread object, make sure its target function uses only local variables in that function.