Scheduling in Python

- Schedule is in-process scheduler for periodic jobs that use the builder pattern for configuration.
- Schedule lets you run Python functions (or any other callable) periodically at pre-determined intervals using a simple, human-friendly syntax.
- Schedule Library is used to schedule a task at a particular time every day or a particular day of a week.
- We can also set time in 24 hours format that when a task should run.
- Basically, Schedule Library matches your systems time to that of scheduled time set by you.
- Once the scheduled time and system time matches the job function (command function that is scheduled) is called.

Before using schedule library we have to install it as

Installation of Schedule module

```
pip install schedule
```

Classes and methods from schedule library

schedule.Scheduler class

schedule.every(interval=1)
Calls every on the default scheduler instance.
Schedule a new periodic job.

schedule.run_pending()
Calls run_pending on the default scheduler instance.
Run all jobs that are scheduled to run.

schedule.run_all(delay_seconds=0)
Calls run_all on the default scheduler instance.
Run all jobs regardless if they are scheduled to run or not.

schedule.idle_seconds()
Calls idle_seconds on the default scheduler instance.

schedule.next_run()
Calls next_run on the default scheduler instance.
Datetime when the next job should run.

schedule.cancel_job(job)
Calls cancel_job on the default scheduler instance.
Delete a scheduled job.

schedule.Job(interval, scheduler=None) class

A periodic job as used by Scheduler.
There are two parameters as

interval: A quantity of a certain time unit
scheduler: The Scheduler instance that this job will register itself with once it has been fully configured in Job.do().

Basic methods for Schedule.job

at(time_str)
Schedule the job every day at a specific time.
Calling this is only valid for jobs scheduled to run every N day(s).
Parameters: time_str – A string in XX:YY format.
Returns: The invoked job instance

do(job_func, *args, **kwargs)
Specifies the job_func that should be called every time the job runs.
Any additional arguments are passed on to job_func when the job runs.
Parameters: job_func – The function to be scheduled
Returns: The invoked job instance

run()
Run the job and immediately reschedule it.
Returns: The return value returned by the job_func

to(latest)
Schedule the job to run at an irregular (randomized) interval.
For example, every(A).to(B).seconds executes the job function every N seconds such that A <= N <= B.