

## ⌚ Python Task Scheduling: `while True` vs Cron (with Example)

---

### ❖ PART 1: Scheduling with `while True + sleep()` in Python

#### Use Case:

Useful when:

- You want **full control inside Python**
  - You don't want to use system tools like cron
  - You're running in **PyCharm, Jupyter, or scripts**
- 

#### ✍ Example Script: `run_every_2min.py`

```
import time
from datetime import datetime

def task():
    with open("/home/ubuntu/time_log.txt", "a") as f:
        f.write(f"Script ran at: {datetime.now()}\n")
    print(f"Task ran at: {datetime.now()}")

while True:
    task()
    time.sleep(120) # Wait 2 minutes
```

---

#### Run This in PyCharm:

1. Create a Python file `run_every_2min.py`
2. Paste the code above

3. Run it in PyCharm
  4. It will keep running every 2 minutes, printing and logging time
- 

## ⚠ Limitations of This Method

Feature	While True
Background Scheduling	✗ Manual only
Auto-start on reboot	✗ No
Time-based control	✓ Custom logic
Resource-efficient	✗ Keeps script running forever

---

---

## ⌚ PART 2: Scheduling with Cron Job (Linux/macOS Only)

### ☑ Use Case:

Best for:

- System-level automation
  - Tasks like backups, data processing, email jobs
  - Script runs in background every 1m, 2h, daily, etc.
- 

## ⌚ Step-by-Step to Schedule with Cron

### ❖ 1. Create Python Script (e.g., `print_time.py`)

from datetime import datetime

Gowtham SB

[www.linkedin.com/in/sbgowtham/](https://www.linkedin.com/in/sbgowtham/)

Instagram - @dataengineeringtamil

with open("/home/ubuntu/time\_log.txt", "a") as f:

f.write(f"Script ran at: {datetime.now()}\n")

---

## 2. Test It Manually

python3 /home/ubuntu/print\_time.py

---

## 3. Open Cron Editor

crontab -e

---

## 4. Add Cron Line (Every 2 minutes)

\*/2 \* \* \* \* /usr/bin/python3 /home/ubuntu/print\_time.py >> /home/ubuntu/cron\_debug.log 2>&1

---

 Done! Now your script runs automatically every 2 mins.

---

## Use This Website to Create Cron Expressions:

 Visit: <https://crontab.guru>

It gives:

- Human-readable meaning of your cron expression
- Helps generate complex patterns easily

For example:

\*/2 \* \* \* \* → every 2 minutes

0 9 \* \* 1 → every Monday at 9 AM

---

## Compare Cron vs While True

Feature	while True	cron
Works on Windows	<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> No (Linux/macOS only)
Background Execution	<input checked="" type="checkbox"/> Manual handling	<input checked="" type="checkbox"/> Built-in
System Reboot Persistence	<input checked="" type="checkbox"/> No	<input checked="" type="checkbox"/> Yes
Simple to Start in PyCharm	<input checked="" type="checkbox"/> Very easy	<input checked="" type="checkbox"/> Needs terminal
Best for real-time apps	<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> No

## Summary

Use `while True` if:

- You're learning, using PyCharm, or just testing

Use `cron` if:

- You want system-level background scheduling that just works!

**Airflow - Python ETL Automation****Etl\_dag.py**

```
from airflow import DAG
from airflow.operators.bash import BashOperator
from datetime import datetime, timedelta

default_args = {
    'owner': 'airflow',
    'depends_on_past': False,
    'email_on_failure': False,
    'email_on_retry': False,
    'retries': 1,
    'retry_delay': timedelta(minutes=1),
}

dag = DAG(
    'mysql_etl_dag', # DAG name
    default_args=default_args,
    description='A simple ETL DAG',
    schedule_interval=timedelta(minutes=5),
    start_date=datetime(2023, 7, 21),
    catchup=False,
)

run_etl = BashOperator(
    task_id='run_etl',
    bash_command='bash /home/ubuntu/wrapper_script.sh ',#give a space after the path
    dag=dag,
)
```

**Etl\_script.py**

```
import pymysql
import pandas as pd
from datetime import datetime
import os

def fetch_data_from_mysql():
    mysql_config = {
        'host': 'localhost',
        'user': 'root',
        'password': 'root',
        'database': 'etl_example'
    }

    connection = pymysql.connect(**mysql_config)
    query = 'SELECT * FROM sample_data'
    df = pd.read_sql(query, connection)
    connection.close()
    return df

def transform_data(df):
    df_transformed = df[df['age'] > 30]
    return df_transformed

def write_data_to_file(df):
    output_dir = '/home/ubuntu/extract'
    os.makedirs(output_dir, exist_ok=True)
    timestamp = datetime.now().strftime('%Y%m%d%H%M%S')
    file_name = f'etl_output_{timestamp}.csv'
    file_path = os.path.join(output_dir, file_name)
```

Gowtham SB

[www.linkedin.com/in/sbgowtham/](https://www.linkedin.com/in/sbgowtham/)

Instagram - @dataengineeringtamil

```
df.to_csv(file_path, index=False)
print(f'Data written to {file_path}')

def etl_process():
    df = fetch_data_from_mysql()
    df_transformed = transform_data(df)
    write_data_to_file(df_transformed)

if __name__ == "__main__":
    etl_process()
```

### **Mysql\_ddl**

```
CREATE DATABASE IF NOT EXISTS etl_example;
```

```
USE etl_example;
```

```
CREATE TABLE IF NOT EXISTS sample_data (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255),
    age INT,
    city VARCHAR(255),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

```
INSERT INTO sample_data (name, age, city) VALUES
('Alice', 30, 'New York'),
('Bob', 25, 'Los Angeles'),
('Charlie', 35, 'Chicago');
```

```
INSERT INTO sample_data (name, age, city) VALUES
('kumar', 40, 'New York');
```

### **wrapper\_script.sh**

```
#!/bin/bash
python3 /home/ubuntu/etl_script.py
```

Gowtham SB

[www.linkedin.com/in/sbgowtham/](https://www.linkedin.com/in/sbgowtham/)

Instagram - @dataengineeringtamil

## About the Author

**Gowtham SB** is a **Data Engineering expert, educator, and content creator** with a passion for **big data technologies, as well as cloud and Gen AI**. With years of experience in the field, he has worked extensively with **cloud platforms, distributed systems, and data pipelines**, helping professionals and aspiring engineers master the art of data engineering.

Beyond his technical expertise, Gowtham is a **renowned mentor and speaker**, sharing his insights through engaging content on **YouTube and LinkedIn**. He has built one of the **largest Tamil Data Engineering communities**, guiding thousands of learners to excel in their careers.

Through his deep industry knowledge and hands-on approach, Gowtham continues to **bridge the gap between learning and real-world implementation**, empowering individuals to build **scalable, high-performance data solutions**.

## Socials

 **YouTube** - <https://www.youtube.com/@dataengineeringvideos>

 **Instagram** - <https://instagram.com/dataengineeringtamil>

 **Instagram** - <https://instagram.com/thedatatech.in>

 **Connect for 1:1** - <https://topmate.io/dataengineering/>

 **LinkedIn** - <https://www.linkedin.com/in/sbgowtham/>

 **Website** - <https://codewithgowtham.blogspot.com>

 **GitHub** - <http://github.com/Gowthamdataengineer>

Gowtham SB

[www.linkedin.com/in/sbgowtham/](https://www.linkedin.com/in/sbgowtham/)

Instagram - @dataengineeringtamil

💬 WhatsApp - <https://lnkd.in/g5JrHw8q>

✉️ Email - [atozknowledge.com@gmail.com](mailto:atozknowledge.com@gmail.com)

📱 All My Socials - <https://lnkd.in/gf8k3aCH>