## **🔁 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

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](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 | ✅ Yes | ❌ No (Linux/macOS only) |
| Background Execution | ❌ Manual handling | ✅ Built-in |
| System Reboot Persistence | ❌ No | ✅ Yes |
| Simple to Start in PyCharm | ✅ Very easy | ❌ Needs terminal |
| Best for real-time apps | ✅ Yes | ❌ 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,

)

**Edl\_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)

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

### **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/](https://instagram.com/dataengineeringtamil)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

💬𝐖𝐡𝐚𝐭𝐬 𝐀𝐩𝐩 - https://lnkd.in/g5JrHw8q

📧𝐄𝐦𝐚𝐢𝐥 - atozknowledge.com@gmail.com

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