

# How to Automate a Recurring Report with a Python Script

*A step-by-step guide for non-developers to set up, run, and schedule a simple automation*

This guide walks you through setting up a small Python script that automates a recurring task, such as compiling a weekly report from a data file. It is written for someone comfortable using a computer but new to running code. By the end, you will have a working script you can run with a single command.

## Before you begin

This guide assumes you have a Windows computer and about 20 minutes. You will not need any prior programming experience. Every command is provided in full.

## What you will need

- A Windows computer with administrator access
- An internet connection (to download Python)
- A sample data file in CSV format (we will use one called report\_data.csv)

## Step 1: Install Python

Python is the program that will run your script. To install it:

1. Go to [python.org/downloads](https://python.org/downloads) in your web browser.
2. Click the yellow **Download Python** button. The site detects your system automatically.
3. Open the downloaded file. On the first screen, check the box labeled **Add Python to PATH** (this step is important, it lets you run Python from any folder).
4. Click **Install Now** and wait for it to finish, then close the installer.



## Step 2: Verify the installation

Confirm Python installed correctly before continuing:

5. Press the **Windows key**, type **cmd**, and press Enter to open the Command Prompt.
6. Type the following command and press Enter:

```
python --version
```

7. You should see a version number, such as **Python 3.12.1**. If you see an error instead, reinstall Python and make sure the **Add Python to PATH** box is checked.

## Step 3: Create your script

8. Open **Notepad** (press the Windows key, type Notepad, press Enter).
9. Paste in the following script:

```
import csv

total = 0
with open('report_data.csv') as file:
    reader = csv.DictReader(file)
    for row in reader:
        total += float(row['amount'])

print(f'Total for this period: {total:.2f}')
```

10. Save the file as **report.py** in the same folder as your CSV file. In the Save dialog, set **Save as type** to **All Files** so it does not save as a .txt file.

### Why this matters

If the file saves as report.py.txt, the script will not run. Setting 'Save as type' to 'All Files' prevents this common mistake.

## Step 4: Run the script

11. In the Command Prompt, navigate to your folder. For example, if it is on your Desktop:

```
cd Desktop
```

12. Run the script with:

```
python report.py
```

13. You should see your result, for example: **Total for this period: 4820.50**.

## Troubleshooting

Problem	Solution
'python' is not recognized	Python was not added to PATH during install. Reinstall and check the 'Add Python to PATH' box.
FileNotFoundError	The script cannot find your CSV. Make sure report.py and report_data.csv are in the same folder, and that you navigated to that folder with 'cd'.
KeyError: 'amount'	Your CSV column is named something other than 'amount'. Open the CSV, check the column header, and update the script to match.

## Next steps

Once your script runs reliably, you can schedule it to run automatically using Windows Task Scheduler, or expand it to write results to a new file instead of printing them. Both are natural follow-ups once the basic script is working.