

# 4-Week Python Automation Freelancing Roadmap

Python is a versatile, easy-to-learn language perfect for automating tasks and building scripts. Even as a beginner you can find small gigs – entry-level Python automation jobs (like simple data or file tasks) often pay **\$5–\$20 per hour**[python.plainenglish.io](https://python.plainenglish.io). With focus and practice, you can build useful scripts (for Excel/CSV processing, web scraping, bots, etc.) and start earning. This 4-week plan breaks down what to learn each week, practical projects to build, and how to showcase your work and find clients. You'll end with a portfolio of 4–5 projects and profiles on freelancing sites, ready to start earning.

## Week 1: Python Basics & Setup

**Goals:** Get comfortable with core Python syntax (variables, data types, loops, functions) and tools. Learn to install Python, use an editor or IDE (like VS Code or free [Repl.it][80]), and manage code with Git. Set up a GitHub account.

### Tasks:

- Follow a free beginner Python tutorial (e.g. freeCodeCamp or YouTube) to cover basics.
- Practice by writing small scripts: e.g. a unit converter, a text file parser, or a filename renamer. Commit your code to GitHub as you go.
- Learn Git/GitHub basics so you can share code. Push your first script to a public repo. A GitHub portfolio helps clients see your skills[medium.com](https://medium.com).
- (Optional) Do simple code challenges on HackerRank or Kaggle to build confidence.

==python - 1 hour - syntax==

- [ ] udemy python/ brocode/ vishal yaduvanshi ( python assignment)

- [ ] push to github

☐ excel - 1 hour

☐ tutorials point

☐ sql - 1hr

- ☐ brocode
- ☐ leetcode
- ☐ resume
- ☐ git projects
- ☐ anki
- ☐ s

my goal is to create an amazing git hub profile and to completely learn

11 - 7 - SLEEP

7 - 8 - FRESHEN UP

8 - 9 BREAKFAST

10-11 -

11 - 12

12- 1 -

1 - 2

3- 4 -

4- 5

5-6

6-7

7-8

8-9

9 - 10

![[<https://i.vgy.me/N7LMX9.png>]]

![[Pasted image 20250507083132.png]]

**Deliverables:** A GitHub repo with 1–2 simple Python scripts (e.g. file renamer, text analyzer) and a README explaining them. By end of Week 1 you should feel comfortable writing and running Python code.

## Week 2: Data & File Automation

**Goals:** Learn to automate data tasks using Python. Use libraries like **pandas**, **csv** / **openpyxl** (for Excel), and the **os** or **glob** modules for file operations. Understand scheduling basics (cron on Linux/Mac or Task Scheduler on Windows) to run scripts automatically.

**Tasks:**

- Build a script to **read/clean CSV or Excel data**. For example, use `pandas` to load a CSV, remove duplicates, and save a cleaned version. (Such data cleanup scripts are in high demand.) In fact, freelancers advertise “Data and Excel work done using automation in Python”[upwork.com](https://www.upwork.com/).
- Automate a file task: e.g. a script that renames files in a folder by a pattern, or batches convert text files to another format.
- Learn about scheduling: write a cron job (or use the `schedule` Python library) to run one of your scripts daily/weekly. Automating routine tasks (backups, reports) saves clients time. (*Cron jobs are built-in scheduled tasks on Unix/Linux systems*[sitepoint.com](https://www.sitepoint.com/).)
- Optional: Build a simple **Flask web app** that lets a user upload a CSV and see basic analysis (sum, average, chart). Even a very basic app (without styling) would be a strong project. Beginner Python freelancers can get paid for such work (~\$200 for a CSV-analyzer [python.plainenglish.io](https://python.plainenglish.io/)).

**Deliverables:** At least two scripts added to GitHub: (1) a data-cleaning script (with `pandas` or CSV module), (2) a file automation script (OS tasks). If you build the Flask app, publish its code (even if just local) too. Write a short blog post or README for each explaining what it does.

## Week 3: Web Automation & Scraping

**Goals:** Learn to interact with the web. Use **requests** and **BeautifulSoup** for scraping static sites, and **Selenium** (or [Playwright][68]) for dynamic/JavaScript-heavy sites. You’ll also learn to use APIs (for example, Twitter or Google APIs) and build small bots.

### Tasks:

- **Static scraping:** Use `requests` to fetch a web page (e.g. news headlines or a product page) and use BeautifulSoup to parse HTML. Extract some data (titles, prices, etc.) and save to CSV. (*Beautiful Soup makes it easy to parse HTML and extract data from static pages*[realpython.com](https://realpython.com/).)
- **Dynamic scraping:** Practice with Selenium to automate a browser. For example, log into a sample site or click a button and scrape data that’s only loaded after actions. (*Selenium “is a powerful suite of tools designed for automating web browsers”*[scrapingbee.com](https://scrapingbee.com/), *useful for sites that use a lot of JavaScript*.)
- **Bot/API project:** Build a simple bot or API-based project. Ideas: a Telegram bot that replies with weather by calling an API; a scheduler that sends you an email report

each morning; or a Twitter bot that posts content periodically.

- **Project ideas:** Price tracker (monitor an item's price on Amazon and alert if it drops), news scraper (email yourself today's headlines), or LinkedIn scraper (extract profile info – check terms of use first!).

**Deliverables:** One or two mid-level projects on GitHub. For example, a news scraper script plus a small bot. Write a short demo and instructions. By now you should have 3–4 projects total. Continue updating your GitHub portfolio with each.

## Week 4: Portfolio, Freelance Profiles & Finding Clients

**Goals:** Polish and showcase what you built, and start looking for work. Create profiles on freelance sites, prepare to pitch yourself, and tap communities. This week is about packaging your skills and actually *finding* clients.

### Tasks:

- **Portfolio:** Aim to have **4–5 solid projects** covering different automation skills (data scripts, web scrapers, bots, simple web app). A variety shows versatility. Publish all code on GitHub and write clear READMEs or blog posts explaining each project. (Clients often check GitHub to verify ability [medium.com](https://medium.com).) If possible, deploy one project (e.g. the Flask app on Heroku or Replit) and include the live link.
- **Profiles:** Sign up on Upwork, Fiverr, and Freelancer.com. Create a focused profile – instead of “Python Developer”, use titles like **“Python Automation & Web Scraping Expert”**. Add descriptions highlighting your projects and skills. Upload your portfolio link (GitHub) on your profile. Take any relevant skill tests (HackerRank offers a Python Certification you can try). Consider adding a LinkedIn profile too, and post about your projects there.
- **Finding work:** Start bidding on small jobs. Look for keywords like “data entry Python”, “web scraper needed”, “Excel automation Python”. Write short, specific proposals referring to your projects. Early on, aim for a few quick gigs (even low-paying) to get good reviews. Upwork and Freelancer have thousands of Python job postings [moldstud.com](https://moldstud.com), and many clients on PeoplePerHour. (PeoplePerHour even shows many Python gigs with rates ~\$30–\$120/hr [moldstud.com](https://moldstud.com).) Also share that you're available on community channels.
- **Communities & learning:** Join forums and groups to get support and leads. For example, StackOverflow and Reddit's r/learnpython or r/freelance often help newbies [moldstud.com](https://moldstud.com). Participate in Discord or Slack Python groups, and local

meetups if possible. Engage in open-source or coding contests (HackerRank, CodeChef) to build credentials and confidence. Networking is key – according to one survey, **70% of engineers** say community help boosts their skills [moldstud.com](https://moldstud.com).

- **Optional enrichment:** Consider taking a free online certification (e.g. Python Institute's PCEP or HackerRank skill badges) to add credibility. Attend virtual hackathons or competitions (Kaggle has beginner-friendly projects) – even if you don't win, they give ideas and practice.

**Deliverables:** A complete GitHub portfolio, a written portfolio page or blog, and active freelance profiles. By end of week 4, you should have applied to multiple jobs. With 4–5 projects ready, your profiles will stand out. Experienced clients value specialization; focusing on “Python automation” niches (data, scraping, bots) will help you win clients faster [moldstud.com](https://moldstud.com).

**Long-Term Paths:** Once you land some freelance gigs, you can increase rates and tackle bigger projects. Over months, build up a track record of client reviews. You could also explore remote full-time roles (sites like Turing.com or WeWorkRemotely list Python jobs) or even start a small automation business/consultancy. The skills you learn (web/API integration, data processing, scheduling, etc.) are widely useful. Keep learning new tools (e.g. Docker, AWS, or no-code platforms like Zapier – [Zapier's “Code” steps]([help.zapier.com](https://help.zapier.com) allow embedding Python in automation workflows). Stay active in communities, contribute to open-source, and gradually raise your profile. With persistence and a strong portfolio, a Python automation freelancer can earn well and even launch ventures down the road [moldstud.com](https://moldstud.com) [upwork.com](https://upwork.com).

**Sources:** Real-world freelancing data and guides inform this roadmap. For example, Upwork notes Python developers commonly earn **\$20–\$40/hr** [upwork.com](https://upwork.com) and specialists even higher [moldstud.com](https://moldstud.com). Beginner freelancers often start with small automation and data projects [python.plainenglish.io/upwork.com](https://python.plainenglish.io/upwork.com). Community involvement (StackOverflow, Reddit, Discord) is proven to accelerate learning [moldstud.com](https://moldstud.com). The steps above combine these insights into a practical 4-week plan.

Citations

Beginner Python Freelancer Projects: Pathways to Earning in 2023 | by Yancy Dennis | Python in Plain English <https://python.plainenglish.io/beginner-python-freelancer-projects-pathways-to-earning-in-2023-a0dd7e4ddac6>  
Is a GitHub Portfolio a Must-Have for Programmers? | by Soma | Javarevisited |

Medium<https://medium.com/javarevisited/is-a-github-portfolio-a-must-have-for-programmers-bf47d462d52b>

Data and Excel work done using automation in Python or VBA |  
Upwork<https://www.upwork.com/services/product/development-it-data-and-excel-work-done-using-automation-in-python-or-vba-1429646790198591488>

Cron Jobs: A Comprehensive Guide — SitePoint<https://www.sitepoint.com/cron-jobs/>

Beginner Python Freelancer Projects: Pathways to Earning in 2023 | by Yancy Dennis |  
Python in Plain English<https://python.plainenglish.io/beginner-python-freelancer-projects-pathways-to-earning-in-2023-a0dd7e4ddac6>

Beautiful Soup: Build a Web Scraper With Python – Real Python<https://realpython.com/beautiful-soup-web-scraper-python/>

Web Scraping Tutorial Using Selenium & Python (+ examples) |  
ScrapingBee<https://www.scrapingbee.com/blog/selenium-python/>

Best Freelance Platforms for Python Developers 2023 |  
MoldStud<https://moldstud.com/articles/p-unlocking-opportunities-top-freelance-websites-for-python-development-in-2023>

Best Freelance Platforms for Python Developers 2023 |  
MoldStud<https://moldstud.com/articles/p-unlocking-opportunities-top-freelance-websites-for-python-development-in-2023>

Best Online Forums for Python Developers and Community Support |  
MoldStud<https://moldstud.com/articles/p-top-10-online-forums-for-python-developers-networking-and-community-support>

Use Python code in Zaps – Zapier<https://help.zapier.com/hc/en-us/articles/8496326417549-Use-Python-code-in-Zaps>

Python Developer Hourly Rates | Cost to Hire Python Developer |  
Upwork<https://www.upwork.com/hire/python-developers/cost/>



All Sources

[python](#)[.plainenglish](#)[medium](#)[upwork](#)[sitepoint](#)[realpython](#)[scrapingbee](#)[moldstud](#)[help.zapier](#)



