

# 02\_Setup\_Computational\_Environment

✓ Published

👤 Assign To

✎ Edit

⋮

**Due:** Before Week 2

**Format:** Individual, completion-based

**Goal:** Ensure every student has a fully working computational environment for the course.

I will be away during Week 1, so this assignment is designed to be completed independently.  
Take your time and follow each step carefully.

This setup gives you a modern scientific computing stack.

## Learning Objectives

By completing this assignment, you will:

- Set up a professional programming environment
- Install tools used in scientific computing and research
- Create your first reproducible software workspace
- Learn the basics of version control and environments

## Required Tools

You will install and configure:

- **Visual Studio Code**
- **Miniconda (Python + environment manager)**
- **GitHub Desktop**
- **A GitHub account**

These tools will be used throughout the course.

## Part 1 — Create a GitHub Account

1. Go to <https://github.com> ➞ (<https://github.com/>)

2. Click **Sign up**

3. Choose:

- A professional username (your name or similar)
- An email address you check regularly


4. Choose the **Free** plan

5. Verify your email

## ✓ Checkpoint

- You can log in and view your GitHub profile page.
- 

## Part 2 — Install Visual Studio Code

1. Go to <https://code.visualstudio.com>  (<https://code.visualstudio.com/>)
2. Download and install the version for your operating system
3. Launch VS Code

## Install extensions

Open the Extensions panel and install:

- **Python**
- **Jupyter**
- *(Optional)* GitHub Pull Requests and Issues

## ✓ Checkpoint

- VS Code opens and extensions install successfully.
- 

## Part 3 — Install Miniconda

1. Go to <https://docs.conda.io/en/latest/miniconda.html>  (<https://docs.conda.io/en/latest/miniconda.html>)
2. Download **Miniconda (Python 3)** for your OS
3. Install using default options

- Allow Conda to initialize your shell if prompted

## Verify installation

Open a terminal (a powershell on Windows, or Applications/Utilities/Terminal on MacOS) and run:

```
conda --version
```

If it says something like 'conda not found', then it isn't on your "path" (the list of directories where your operating system looks for programs). Troubleshoot with ChatGPT, or similar.

### ✓ Checkpoint

- A Conda version number is printed.

---

## Part 4 — Create a Conda Environment

In your terminal:

```
conda create -n compsci python=3.11
```

Activate it:

```
conda activate compsci
```

Install packages:

```
conda install numpy scipy matplotlib jupyter pandas
```

### ✓ Checkpoint

- Your terminal prompt shows `(compsci)`.

---

## Part 5 — Install GitHub Desktop

1. Go to <https://desktop.github.com> ➞ [\(https://desktop.github.com/\)](https://desktop.github.com/)
2. Download and install
3. Sign in using your GitHub account
4. Configure your name and email

## ✓ Checkpoint

- GitHub Desktop opens and shows you as logged in.
- 

## Part 6 — Create Your Course Repository

1. In GitHub Desktop, click **New Repository**

2. Name it:

```
compsci2026
```

3. Choose a location on your computer

4. Click **Open in Visual Studio Code**

## ✓ Checkpoint

- The same folder is visible in both VS Code and GitHub Desktop.
- 

## Part 7 — First Program Test

In VS Code, open a terminal and activate your environment:

```
conda activate compsci
```

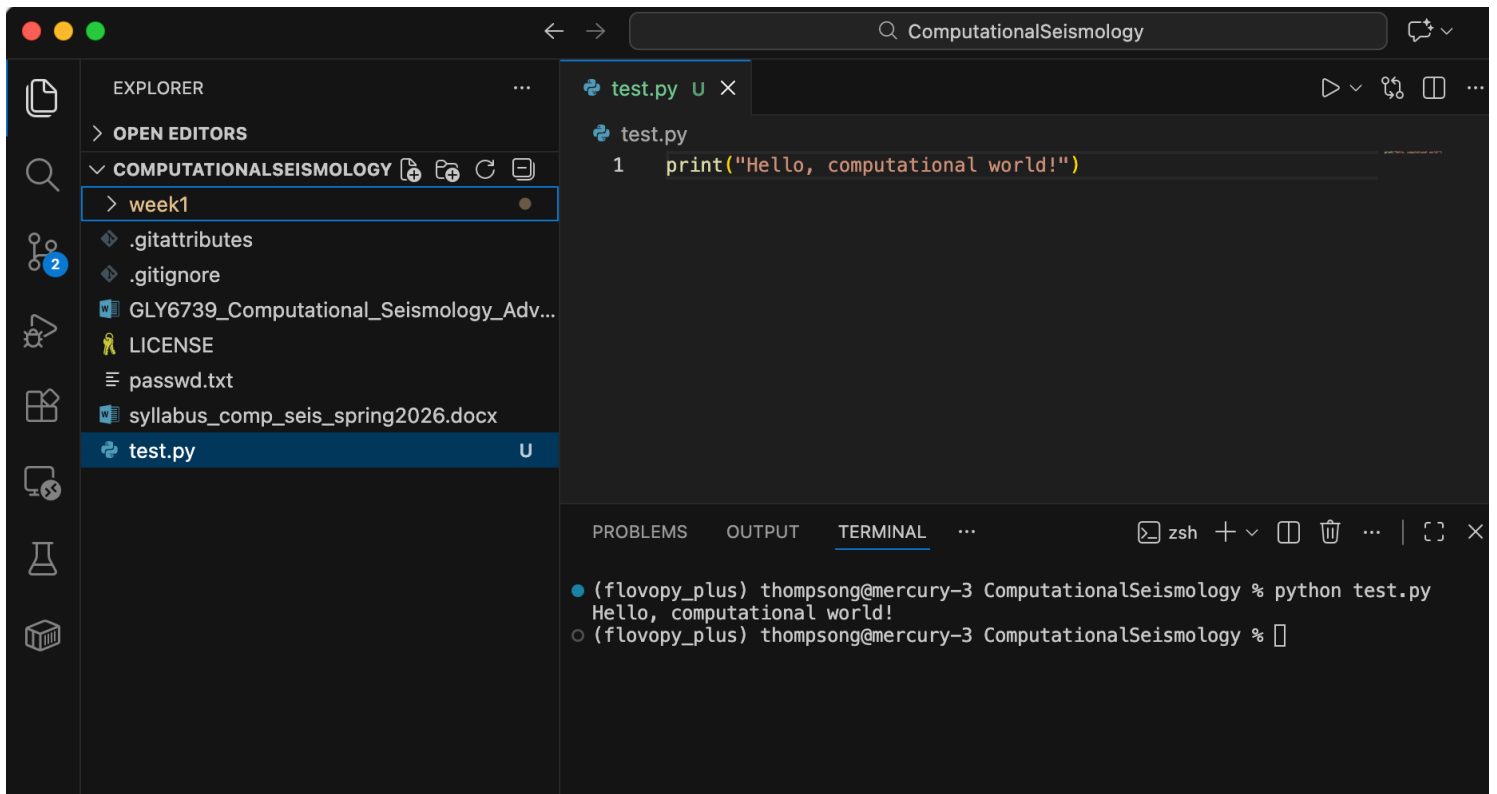
Create a file called `test.py` containing:

```
print("Hello, computational world!")
```

Run it directly from the Visual Studio Code terminal:

```
python test.py
```

Here is how it should look (approximately):



Commit the file using GitHub Desktop with a suitable comment. Then "Push origin" (upper right on GitHub Desktop - you might have to click on "Fetch origin" and then "Commit" before this appears). Here is how GitHub Desktop should look (approximately):

The screenshot shows the GitHub web interface for the repository **GLY6739.017S26\_ComputationalSeismology**. The current branch is **main**. The **Changes** tab is active, showing 2 changed files: **test.py** and **week1/01\_reading\_computing\_1936\_1972.ipynb**. The **test.py** file is selected, and its content is displayed in the right pane:

```
@@ -0,0 +1 @@
+ print("Hello, computational world!")
```

Below the file list, there is a commit message input field with the text "Created test.py" and a description field. At the bottom, a blue button reads "Commit 1 file to main".

Finally, navigate to your test.py file on github.com. It should look like this (approximately):

The screenshot shows the GitHub web interface for the repository **GLY6739.017S26\_ComputationalSeismology**. The **Code** tab is active, and the **test.py** file is selected. The file content is displayed in the right pane:

```
1 print("Hello, computational world!")
```

The left sidebar shows the file explorer with the following structure:

- week1
  - images
    - 01\_reading\_computing\_1936\_19...
    - 03\_lab\_compiling.ipynb
  - rms\_large.c
  - rms\_large.f90
  - rms\_large.py
  - rms\_large\_numpy.py
  - .DS\_Store
  - .gitattributes
  - .gitignore
  - GLY6739\_Computational\_Seismol...
  - LICENSE
  - passwd.txt
  - syllabus\_comp\_seis\_spring2026....
  - test.py

## ✓ Final Checkpoint

- Python runs
  - The file executes correctly
  - The repository has at least one commit
- 

## Submission Instructions

Submit the following to Canvas:

- A screenshot showing:
    - VS Code open
    - `test.py`
    - Terminal output from running `test.py`.
  - A screenshot of your `test.py` program on github.com.
- 

## Troubleshooting

If something doesn't work:

- Copy the exact error message
  - Take screenshots
  - Get help from other students, or email me at [thompson@usf.edu](mailto:thompson@usf.edu)
  - Bring the issue to Week 2 — debugging is part of computational science
- 

## One-sentence takeaway

*This week is about making your computer a predictable, reproducible scientific tool.*

Points 5

Submitting a file upload