

# Python Stack

Prof. Calvin

2025-01-09

# Table of contents

<b>Readme</b>	<b>3</b>
<b>1 Python, VS Code, Quarto</b>	<b>4</b>
<b>2 Python, VS Code, Quarto</b>	<b>5</b>
2.1 Install Python . . . . .	5
2.1.1 Search Engine . . . . .	5
2.1.2 Python.org . . . . .	5
2.1.3 Test Python . . . . .	5
2.2 Install VS Code . . . . .	5
2.2.1 Aside . . . . .	5
2.2.2 Install . . . . .	6
2.2.3 VS Code Setup . . . . .	6
2.2.4 Python extension setup . . . . .	6
2.3 Install Quarto . . . . .	7
2.3.1 Search Engine . . . . .	7
2.3.2 Install Quarto . . . . .	7
2.4 Install Jupyter . . . . .	7
2.5 Cleanup . . . . .	9
2.5.1 Quarto Preview . . . . .	9
2.5.2 Render .md . . . . .	9
<b>3 Quarto Books and GitHub Pages</b>	<b>10</b>
<b>4 Quarto Books and GitHub Pages</b>	<b>11</b>
4.0.1 Search Engine . . . . .	11
4.0.2 At the Terminal . . . . .	11
4.0.3 In VS Code . . . . .	11
4.0.4 Creating Content . . . . .	12
<b>5 On GitHub</b>	<b>14</b>
5.0.1 Publish Content to GitHub Pages . . . . .	14
5.0.2 Create a Git repository . . . . .	14
5.0.3 Create a New Repository on the Command Line . . . . .	14
5.0.4 Follow the Instruction from Quarto . . . . .	15

# Readme

This self-documenting book introduces the usage of the Python technology stack with Quarto, VS Code, and GitHub for communicating results within the computational sciences.

# 1 Python, VS Code, Quarto

## 2 Python, VS Code, Quarto

### 2.1 Install Python

#### 2.1.1 Search Engine

1. Search “Python for VS Code”
2. Go to <https://code.visualstudio.com/docs/languages/python>
3. Click “Python Tutorial”
4. Click “Install Python for python.org”

#### 2.1.2 Python.org

1. Click “Download Python 3.XX.X”
2. Open the installer from Downloads
3. Check both boxes when the “Setup” prompt comes up.
4. After the install, select “Disable path length limit”

#### 2.1.3 Test Python

- Windows Key
- Type “terminal”
- In terminal, type “python”
- In Python, type:

```
print("hello world")
```

```
hello world
```

### 2.2 Install VS Code

#### 2.2.1 Aside

- It is of great benefit to us as learners that VS Code includes “Copilot” (LLM) integration.

### 2.2.2 Install

1. Go to <https://code.visualstudio.com/>
2. Click “Download for Windows”
3. Open the installer from Downloads
4. Check all four boxes when the “Setup” prompt comes up.
5. After the install, select “Launch Visual Studio Code”

### 2.2.3 VS Code Setup

1. Use Copilot, or don’t.
2. Choose your theme
3. Select “Browse Language Extensions”
  - Select “Python”
    - Keep this window open
  - Select “Jupyter”
  - Optionally, look into [R in Visual Studio Code](#)
4. Click through the remaining options.

### 2.2.4 Python extension setup

1. Create a new Python project
2. Create a new folder, likely Documents/DATA505
3. Create a file “hello.py”:

---

**Listing 2.1** hello.py

---

```
# hello.py  
print("hello world")
```

---

4. Click the [**>**] button to run the file, which should print out to the terminal:

```
hello world
```

5. Return to the Python tutorial.
  - Click “Select Python Interpreter”
  - Select the version of Python you installed earlier.

## 2.3 Install Quarto

### 2.3.1 Search Engine

1. Search “Quarto”
2. Go to <https://quarto.org/>
3. Click “Get Started”
4. Click “Download Quarto CLI”

### 2.3.2 Install Quarto

1. Run the Installer
  - All default options are fine.
2. Install the VS Code Extension.
  - Go to <https://marketplace.visualstudio.com/items?itemName=quarto.quarto>
  - Click the “Install” button.
  - Allow the installation in VS Code and your browser.
3. Setup Python for Quarto

## 2.4 Install Jupyter

\*To work with .qmd files, Python needs a package akin to R “languageserver”

1. Go to <https://quarto.org/docs/get-started/hello/vscode.html>
2. Run the following terminal command:
  - If you lost your terminal, create a new one from the menu.

```
py -m pip install jupyter matplotlib plotly
```

- This will take a moment.
3. Following the Quarto tutorial, make `hello.qmd`
  4. Click “Run Cell” in `hello.qmd` on this cell:

```

import numpy as np
import matplotlib.pyplot as plt

r = np.arange(0, 2, 0.01)
theta = 2 * np.pi * r
fig, ax = plt.subplots(
    subplot_kw = {'projection': 'polar'}
)
ax.plot(theta, r)
ax.set_rticks([0.5, 1, 1.5, 2])
ax.grid(True)
plt.show()

```

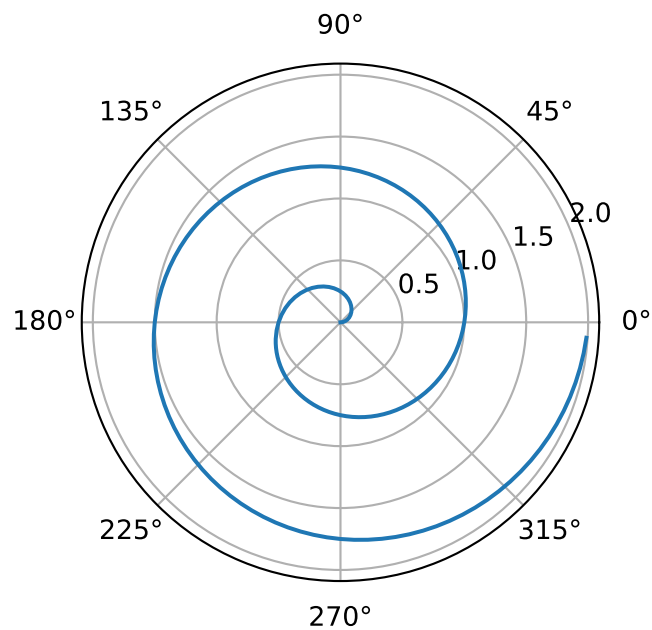


Figure 2.1: A line plot on a polar axis

5. When prompted, install the `ipykernel`
6. I restarted VS Code.



## 2.5 Cleanup

### 2.5.1 Quarto Preview

1. To test whether the process worked, open `hello.qmd` in VS Code.
2. Quarto preview `hello.qmd` either by pressing [`>`] or by using `ctrl+shift+k`.
3. After a moment, a document will be displayed, in VS Code and if you wish, in browser.

### 2.5.2 Render .md

*To make this document.*

1. Prefix the .md document with a .qmd header:

```
---  
title: "Python, VS Code, Quarto"  
format:  
  html:  
    code-fold: true  
jupyter: python3  
---
```

2. Rename from `fname.md` to `fname.qmd`
  - I did this by creating a new file and using copy+paste
3. Use `quarto render fname.qmd` either:
  - In the VS Code integrated terminal, or
  - At the command line, e.g. Windows Key+“terminal”

## **3 Quarto Books and GitHub Pages**

## 4 Quarto Books and GitHub Pages

### 4.0.1 Search Engine

1. Search “Quarto Books”
2. Go to <https://quarto.org/docs/books/>

### 4.0.2 At the Terminal

1. Created a folder called “work”.
2. I ran the following in the terminal:

---

**Listing 4.1** Terminal

---

```
quarto create project book mybook
```

---

3. I will be prompted for a title
  - I used “Python Stack”
4. I will be prompted if I want to open in `vscode`
  - I accepted.

### 4.0.3 In VS Code

1. Generating the book project automatically generated a `_quarto.yml` file.
2. I updated the file to be as follows, with only two chapters:

---

**Listing 4.2** \_quarto.yml

---

```
project:
  type: book

book:
  title: "mybook"
  author: "Jane Doe"
  date: "8/18/2021"
  chapters:
    - index.qmd
    - intro.qmd
    - summary.qmd
    - references.qmd

bibliography: references.bib

format:
  html:
    theme: cosmo
  pdf:
    documentclass: scrreprt
  epub:
    cover-image: cover.png
```

---

#### 4.0.4 Creating Content

1. I view the initial book by opening a .qmd file and using **Preview**
  - By default on Windows, **Ctrl+Shift+K**
2. I include some content in my new .qmd files.
  - For me, this was a prior lecture “Python, VS Code, Quarto”
  - For me, this was *this* .qmd on books & pages.
3. I use a full **render** after creating content to ensure everything is up-to-date.

---

**Listing 4.3** \_quarto.yml

---

```
project:
  type: book

book:
  title: "Python Stack"
  author: "Prof. Calvin"
  date: "1/9/2025"
  chapters:
    - index.qmd
    - VSCode.qmd
    - Books.qmd

format:
  html:
    theme: cosmo
  pdf:
    documentclass: scrreprt
```

---

---

**Listing 4.4** Terminal

---

```
quarto render
```

---

# 5 On GitHub

Follow *this guide*

## 5.0.1 Publish Content to GitHub Pages

1. Search “Quarto Publish”
2. Go to <https://quarto.org/docs/publishing/github-pages.html>
3. Read the following:

an important pre-requisite: you need to have a Git repository on your local machine that is synced to GitHub.

## 5.0.2 Create a Git repository

1. Navigate to <https://github.com/>
2. Select the Green Button **New**
  - It will take you to this url: <https://github.com/new>
3. Do some setup:
  - Select an owner
    - I am @cd-public
  - Name the repository
    - I’ll do “python-stack-book”
  - Add a description.
    - “Demo quarto book on the Python tech stack.”
4. Select the Green Button **Create repository**

## 5.0.3 Create a New Repository on the Command Line

1. Navigate in the file system to your book directory.
  - You will need to set up Git at the command line.

#### 5.0.4 Follow the Instruction from Quarto

1. Return to <https://quarto.org/docs/publishing/github-pages.html>
2. Follow instructions from [Render to docs](#) up to `Publish Command`