Step-by-Step Environment Set-Up

Install Visual Studio Code

What it is? A code editor.
Why do I need it? You don't *need* it, but it makes coding easier by colour-coding code, auto-completions, and highlighting errors.
Where do I get it? Here

Install Python

What it is? The programming language.
Why do I need it? To tell your computer how to execute programs written in Python syntax.
Where to get it? Here

Install Python Extension

What it is? Additional helpful add-ons for the coding editor.
Why do I need it? To get Python-specific help while you code.
Where to get it? VSCode → ctrl + shift + x → search 'Python Extension Pack' → install

Get a GitHub Account

What it is? An open-source online platform for storing and sharing code.
Why do I need it? To back up your code, collaborate with others, and showcase your projects.
Where to get it? Here

Install Git

What it is? A local version control system.
Why do I need it? To track changes to your code and to upload your changes to GitHub.
Where to get it? Here

Install Git Extension & Sign In

What it is? Code editor add-ons for working with Git.
Why do I need it? To use Git through VSCode interface rather than the terminal.
Where to get it? VSCode → ctrl + shift + x → search 'GitHub Pull Requests' → install; sign into your GitHub account in VSCode → Accounts (bottom left corner)

Configure Git in VSCode

What it is? Telling your VSCode to communicate with *your* GitHub.
Why do I need it? To identify whose changes are being saved on GitHub.
How do I do this? VSCode → ctrl + ` → run `git config --global user.name "[your username]"` → run `git config --global user.email "[your email]"`

Now you're ready to start!

Starting a Project

Create a Project

1. In VSCode, go to Explorer (paper icon or ctrl + shift + E)
2. Hit 'Open Folder' and navigate to your project's folder (or create a new one)
3. Select the desired folder and hit 'Open'
4. In the Explorer, navigate to the folder title on the left-hand side and hit 'New File'
5. Name your file and add `.py` at the end to let VSCode know this is a Python file

Upload Your Project Online

1. In VSCode, go to Source Control (branch icon or ctrl + shift + G)
2. Hit 'Initialize Repository'
3. In the text field on the left-hand side, type a quick message to describe your file and hit 'Commit'
4. Once committed, hit 'Publish Branch'
5. In the dropdown menu, select 'Publish to GitHub public repository'
6. Double-check on GitHub that there is a repository with your project's name containing your file

Propose Changes to Others' Projects

1. On GitHub, go to a repository you want to modify, hit the green 'Code' button, and copy the URL
2. In VSCode, go to Source Control (ctrl + shift + G) and hit 'Clone Repository'
3. Paste the copied URL into the revealed text field and hit enter
4. Navigate to where you want to save the repository and hit 'Select as Repository Destination'
5. Source Control → 'More Actions' (3 dots next to 'CHANGES') → 'Branch' → 'Create Branch'
6. Name your branch and hit enter
7. Make changes to the code
8. In the VSCode GitHub tab, hit 'Create Pull Request' (next to 'PULL REQUESTS') and write a message

Review Others' Change Proposals

1. In your VSCode, hit the 'sync changes' icon (bottom left bar)
2. In the VS Code GitHub Pull Requests tab, find your unresolved pull requests
3. Select a pull request and hit 'Open Changes' (icon next to the request title)
4. Review changes and add comments by hitting '+' on the changed lines
5. Hit the request's title to open a tab with details of the request
6. Select 'Approve' for incorporating the change, 'Comment' for resolving the change by commenting on it, or 'Request Changes' to request changes to the proposed change

Primitive Data Types

Name	Stores	Examples
`int`	integer value	`3, 7, 42`
`float`	float value	`3.14, 2.0`
`str`	text	`'Sherlock', "Holmes"`
`bool`	true or false values	`True, False`

You can print the data type of `x` by running `print(type(x))`. Learn more about other data types here.

Operators

Assignment Operators

Operator	Function	Example	Equivalent
`=`	assigns a value to name	`x = 2`	
`+=`	increases assigned value by new value	`x += 3`	`x = x + 3`
`-=`	decreases assigned value by new value	`x -= 3`	`x = x - 3`
`**=`	raises assigned value to the power of new value	`x **= 3`	`x = x**3`

Arithmetic Operators

Operator	Returns	Example
`+`	addition of two values	`x + y`
`-`	subtraction of two values	`x - y - z`
`*`	multiplication of two values	`x * y`
`/`	division of two values	`x / y`
`**`	exponentiation (power) of two values	`x ** y`
`%`	remainder after division (modulo)	`x % y`

Logical Operators

Operator	Returns	Example
`and`	True if both conditions are true	`x > 5 and y < 10`
`or`	True if at least one condition is true	`x > 5 or y < 10`
`not`	True if condition is false (negation)	`not x > 5`

Comparison Operators

Operator	Returns	Example
`==`	True if values are equal	`x == 5`
`!=`	True if values are not equal	`x != 5`
`>`	True if left value is greater	`x > 5`
`<`	True if left value is smaller	`x < 5`
`>=`	True if left value is greater or equal	`x >= 5`
`<=`	True if left value is smaller or equal	`x >= 5`

Learn more about other operators here.

Couple General Tips

1. **Many issues can be resolved by reading notifications and manuals closely.** When the computer is trying to communicate, take a moment to read what it's saying—it's usually trying to help.

2. **Most programmers Google (and LLM!) stuff all the time.** Whatever you're struggling with, someone has probably struggled with before; go out there and see what's helpful.

3. Thanks to the Jupyter extension you installed as a part of the Python Extension Pack, **typing `#%%` will split your code into 'cells' that you can run independently** (no need to make a new file for each task).

4. **Always document your code.** This is not difficult, just type `##` next to specific lines and write a brief description of what your code does. (You can also document questions and limitations.)

5. **Print stuff for debugging and/or better understanding.** In Python, type `print()` with whatever you want to display inside the parentheses to see it printed in the terminal.