

# Babson Coding Environment Setup

Welcome to your Babson Python coding environment!

This guide walks you through setting up GitHub and Codespaces - your cloud-based coding workspace. By the end, you'll have your own repository, a working Python IDE, and the tools needed to write and submit assignments.

Follow each step carefully. If something doesn't look right, ask for help!

Step 1: Log into your GitHub account. This is where your code will live.

Step 2: Open the Canvas assignment link and click 'Accept'.

This creates your own personal copy of the class repository - like getting your own digital notebook.

Step 3: Your new repo will be named:

`python-YOUR_GITHUB_USER_NAME`

This is your private workspace for the course.

Step 4: Go back to github.com, refresh, and click on your new repo.

Step 5: Click Code -> Codespaces -> Create codespace on main.

This launches a cloud-based coding environment with your files already loaded.

Step 6: Open Settings: hamburger menu -> File -> Preferences -> Settings.

Step 7: Search for 'git.untracked', change MIXED to HIDDEN, then close settings.

This hides unnecessary Git files so you can focus on your code.

**Visual reference for Settings panel:**



Step 9: Click Extensions in the left sidebar, search 'Live Server', and install the one by Ritwick Dey.

This lets you preview HTML files in your browser.

Step 10: In the terminal (just once for the course), enter:

```
git remote add upstream https://github.com/babson-org/classroom-week00-python_class.git
```

This connects your repo to the teacher's master copy so you can pull updates.

**Git Source Control view in Codespaces shows a commit icon on the upper left. Clicking it adds a suggested commit message:**

## What's Going On?

GitHub is your code's home base. When you accept the assignment, GitHub creates a private copy of the teacher's repository just for you.

Codespaces is your coding workspace - like a virtual computer in the cloud. It opens your repo in a full-featured Python editor.

Git tracks changes to your files. When you 'commit', you're saving a snapshot of your work. You'll write a short message each time to describe what you changed.

To keep your repo up to date with the teacher's version, you'll use:

```
git pull --no-edit upstream main
```

This command does three things:

- \* Connects to the teacher's repo (called 'upstream')
- \* Pulls in the latest changes from the main branch
- \* Merges them into your repo without asking for a commit message

**\*\*** Before running this command, you must commit your changes and sync. If you haven't committed and synced, Git won't let you pull - and you risk losing unsaved work.

\*\* You'll use this command often:

\* After your instructor announces updates

Make it a habit:

Commit -> Sync -> Pull -> Code

**IMPORTANT: Always commit and sync your work BEFORE updating from the master repository.**

### **Setup Checklist**

# Accepted the GitHub Classroom assignment

# Created a Codespace on main

# Installed Live Server extension

# Set git.untracked to HIDDEN

# Added upstream remote

# Committed at least once