CS 504 Software Engineering

**HOS02A –Git Repositories and Refactoring**

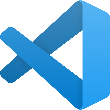
03/10/2023 Developed by Ajay Naik

08/24/2023 Reviewed by Lisa Schultz

09/05/2024 Reviewed by Yogesh Goel

10/08/2024 Reviewed by Naveena Moddu

School of Technology & Computing (STC) @ City University of Seattle (CityU)

**Before You Start**

* Version numbers may not match the most current version at the time of writing. If given the option to choose between the stable release (long-term support) or the most recent, please select the stable release.
* This tutorial targets Windows and MacOS users.
* There might be subtle discrepancies along the steps. Please use your best judgment while going through this tutorial.
* For your working directory, use your course number. This tutorial may use a different course number as an example.

**Learning Outcomes**

* Understanding useful commands of Git
* The Git Pull Request flow
* Features of VS Code
  + - [Debugging](https://code.visualstudio.com/docs/python/debugging)
    - Peer-Programming
    - Git integration
    - Error Resolution
    - [Linting](https://code.visualstudio.com/docs/python/linting)

**Resources**

* <https://cityuseattle.github.io/docs/git>
* <https://git-scm.com/>
* <https://git-scm.com/docs>
* <https://www.atlassian.com/git/glossary>
* <https://cityuseattle.github.io/docs/tools/vscode/>
* <https://www.javatpoint.com/git>
* <https://code.visualstudio.com/docs/python/linting>
* <https://www.w3docs.com/learn-git/introduction1.html>

**Instructions**

1. Complete all the steps described in this document.
2. At the end of the assignment, follow the instructions to commit your work to GitHub.

**Pre-requisites**

Install Git before exploring the features. Git installation instructions are shown here:

<https://cityuseattle.github.io/docs/git/install>

**Introduction**

Git is a distributed version control system allowing users to track repository changes. It is commonly used for coordinating collaborative software development among programmers/users, enabling them to keep track of each other's progress and contributions. We will explore the most used commands with hands-on usage for Pull, Branch, Status, Add, Commit, Push, and other non-frequently used commands.

**Pull Request**: Pull requests let you tell others about changes you've pushed to a branch in a repository on GitHub. Once a pull request is opened, you can discuss and review the potential changes with collaborators and add follow-up commits before your changes are merged into the base branch.

Visual Studio Code is a complete tool the developer needs and provides all necessary plugins and features to work with various programming languages. We can do peer programming with VSC, easily find errors, integrate Git, and many more.

**Part 1: Git Development Cycle**

**Git checkout**

This command will create a new branch on your local machine, and you can work on this branch as part of the development cycle. Run this command in a folder (clone the HOS02A repository in your local machine) that can be used to try git.

**Command**

git checkout -b hello-git-branch

**Output**



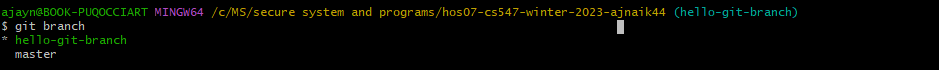
**Git branch**

A Git project can have more than one branch. These branches are a pointer to a snapshot of your changes. When you want to add a new feature or fix a bug, you spawn a new branch to summarize your changes. This command will display the current branches:

**Command**

git branch

**Output**



Note: If you do not see any branches, ensure that you have a git repository set up in your working directory and that you have made at least one commit.

**Adding Files**

Add to the folder a file named hello.txt with the following line in it

Hello World

**Git status**

The git status command shows the state of the working directory and the staging area. It lets you see staged changes and the files that Git is not tracking.

**Command**

git status

**Output**

Text

Description automatically generated

**Git add**

This command will add all new files added in your branch. Add the hello.txt file to git.

**Command**

git add –-all

**Output**



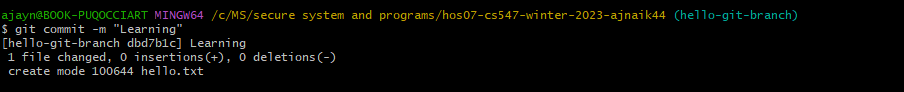
Note: Windows users may not see an output

**Git commit**

The commit command is used to save changes to the local repository. It creates a point to which you can revert files to. Commit your changes to the local repository.

**Command**

git commit -m "Learning"



**Git push**

The push term refers to uploading local repository content to a remote repository. Push your code to the repository.

**Command**

git push -f origin hello-git-branch

**Output**

Text

Description automatically generated

**Pull Request (PR) Process**

Pull requests are a mechanism for a developer to notify team members that they have completed a feature. Once their feature branch is ready, the developer files a pull request via their GitHub account.

Follow the below steps to create the Pull request

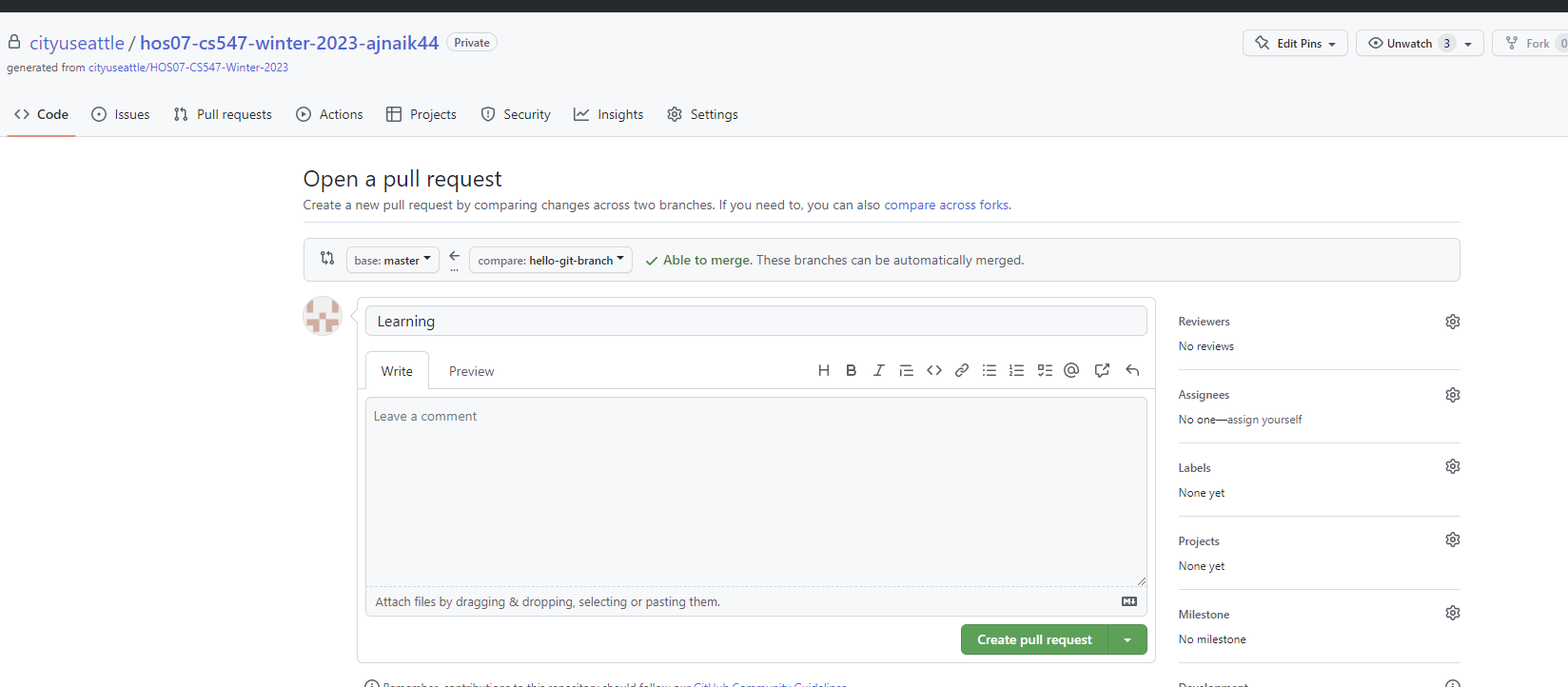
Go to your GitHub account (This is the GitHub branch URL from which you clone the repository.)

Once you log in, you should see this screen:

Graphical user interface, text, application, chat or text message, email

Description automatically generated

Click on compare and pull request option.



Click on Create pull request.

Graphical user interface, text, application, email

Description automatically generated

On the right-hand side, you will see the Reviewers list. Click on setting.

Graphical user interface, text, application, email

Description automatically generated

The Reviewers will get the mail with the Pull Request (PR) URL and can review and gives their feedback.

If the reviewers are ok with the changes, they will accept them, and then you can merge your PR in the main branch. In this case, it is the master branch.

**For this HOS, you do not need the Reviewers Approval**

Click on Merge Pull Request and then Confirm Merge.

Graphical user interface, text, application, email

Description automatically generated

Click on the Delete branch.

The changes are now in the main branch; we can bring them to the folder.

**Command**

git checkout master

git pull

**Output**

Text

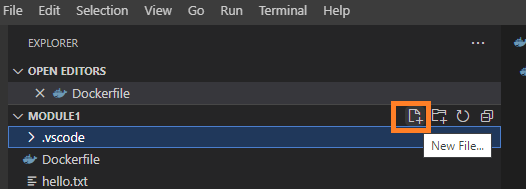
Description automatically generated

**Part 2: Coding Tools in Visual Studio**

**Debugging**

Debugging is very important in programming. Programmers should be able to debug the code and find the root cause of the failing code. Follow and explore the debug features.

Create a file addNumbers.py



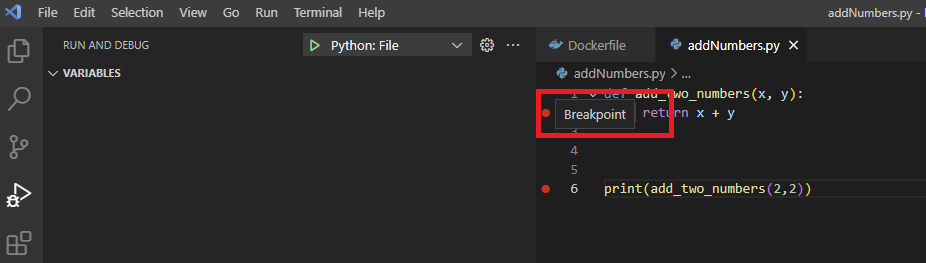
Include the following code in the file:

def add\_two\_numbers(x, y):

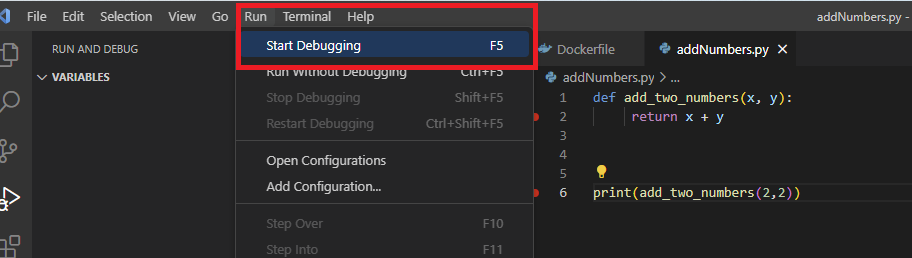
  return x + y

print(add\_two\_numbers(2,2))

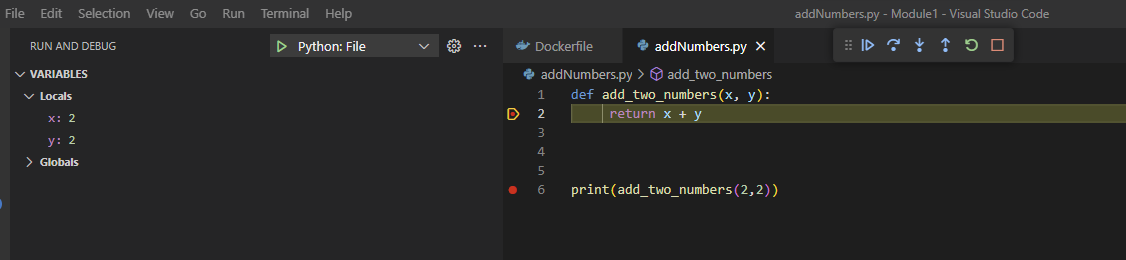
1. Add a breakpoint on the print statement



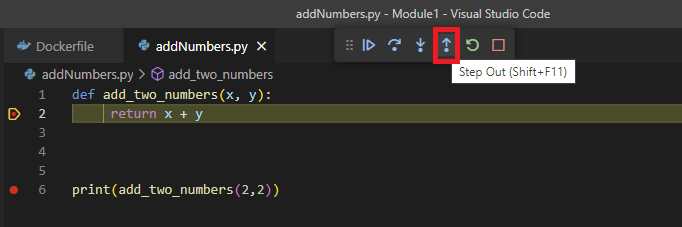
Click on Run -> Start Debugging.



Use F11 (step into) to start debugging the add\_two\_numbers function



Use F10 (step over) to run each step in the function

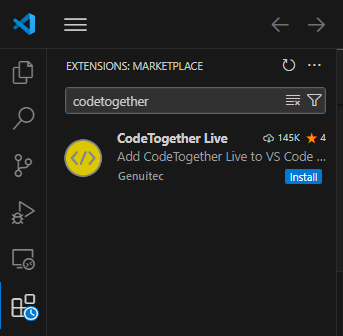


The following link <https://cityuseattle.github.io/docs/debugging> includes a video explaining these steps.

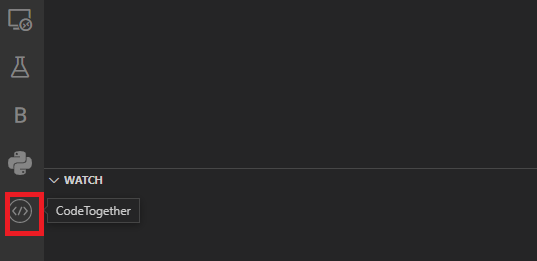
**Peer Programming**

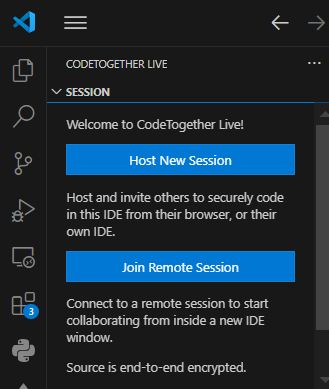
Peer programming can help programmers learn new skills and technologies they may not be familiar with. Through collaboration, programmers can learn from each other and expand their knowledge.

Click on Extensions in VSC on the left side and type CodeTogether. Install it.



Click on CodeTogether from the left panel.





Click on Host New Session and Start.

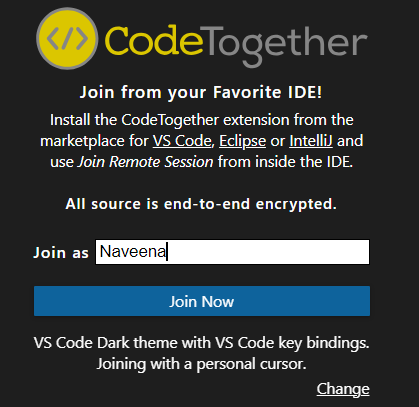
Graphical user interface, text, application

Description automatically generated

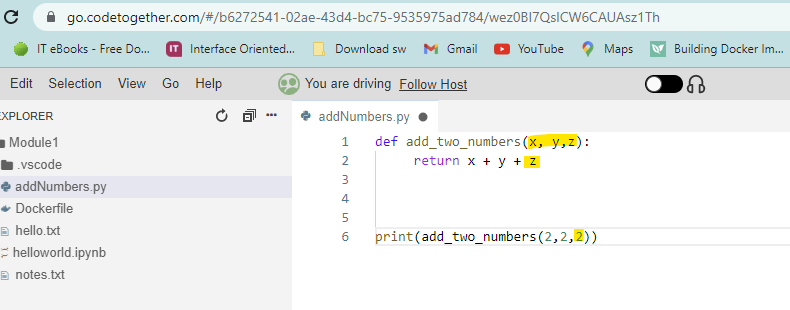
Click on Invite Others and open a browser and paste the URL.

Enter your name and click on Join Now.

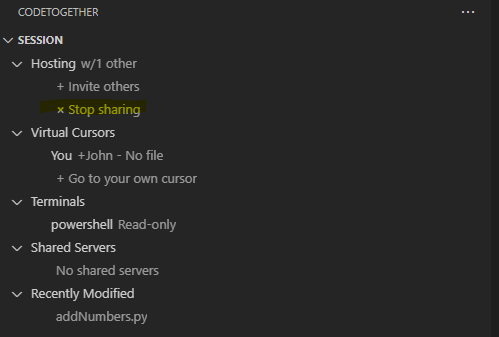




Modify the code, which will be reflected in your VSC code.



Click on Stop sharing Session

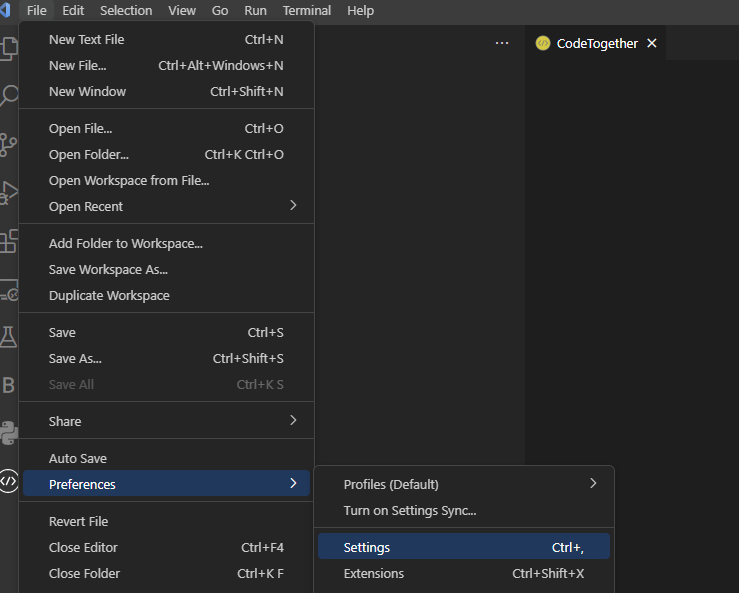


**Git in VS Code**

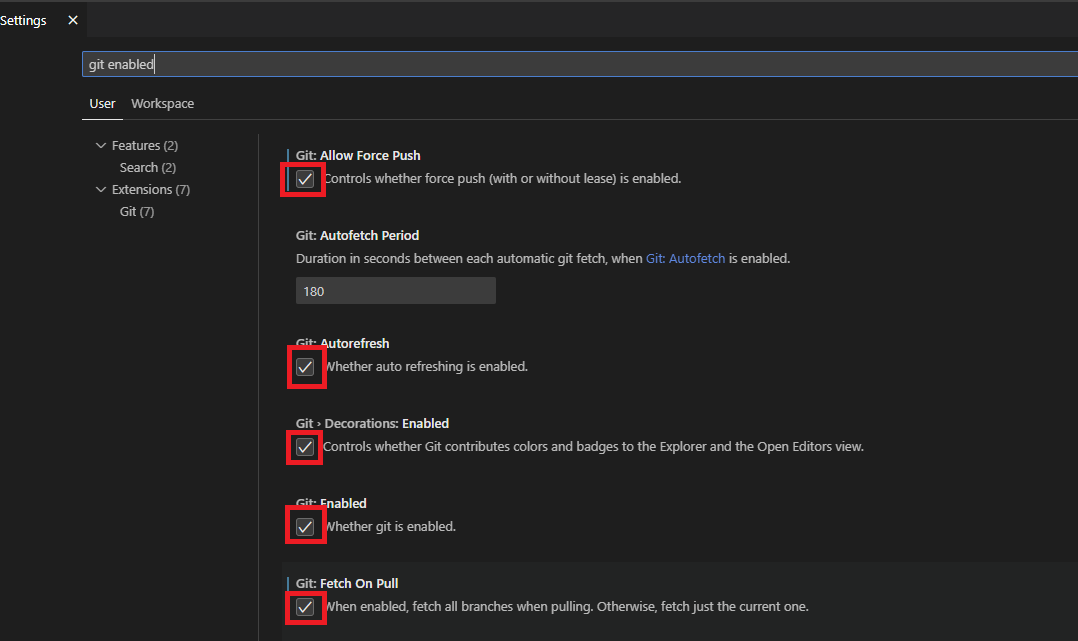
Git integration with Visual Studio Code allows developers to manage and track changes to their source code quickly, collaborate with other developers, and deploy their applications faster.

**Enable Git in VS Code on Windows:**

Go to file > Preferences > Settings

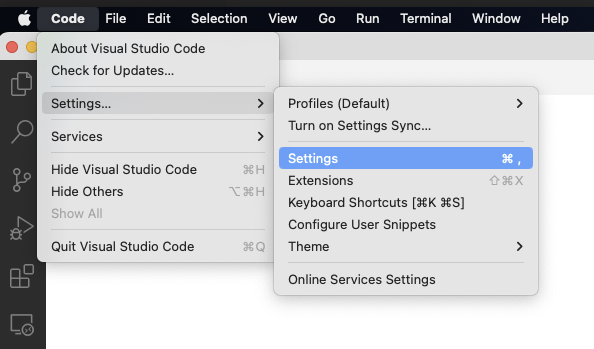


Type Git: Enabled in the search bar. Make sure that all box is ticked

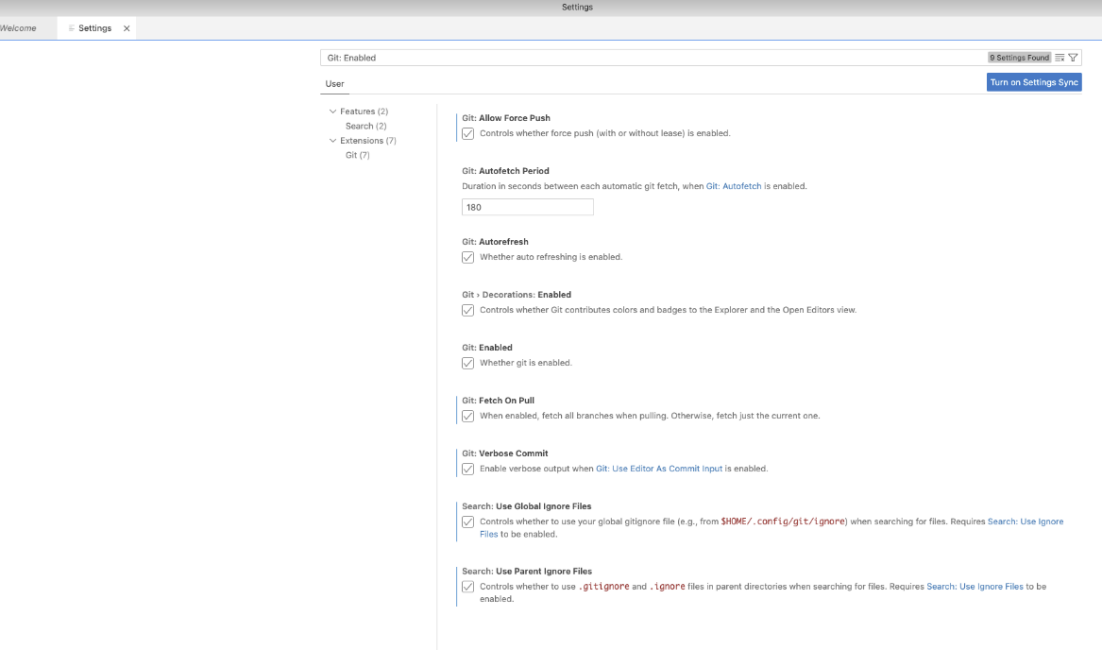


**To enable Git in VS Code on Mac:**

Got to code > Settings... > Settings



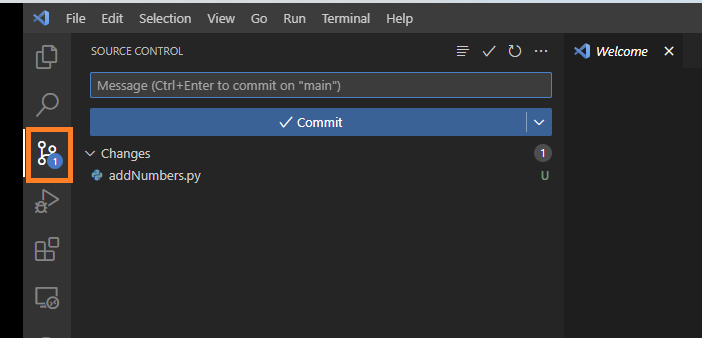
Type Git: Enabled in the search bar. Make sure that the box is ticked

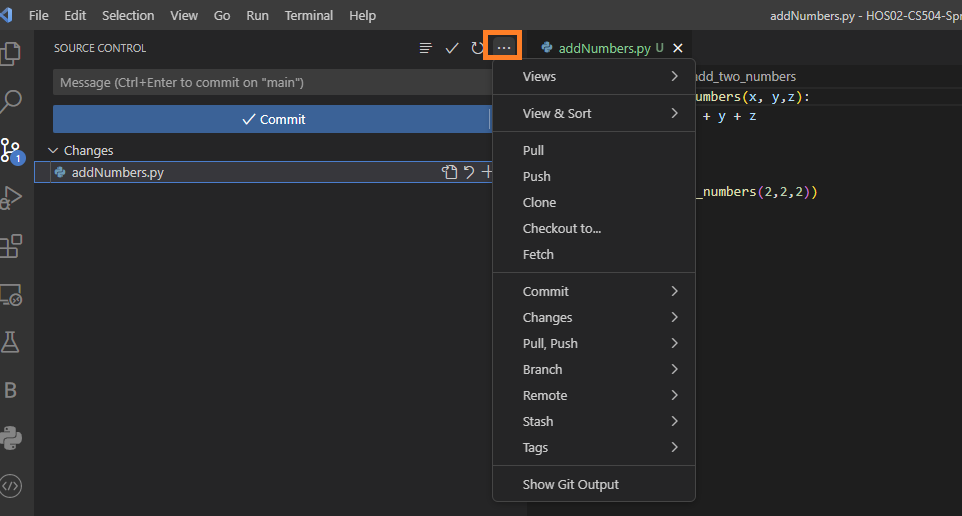


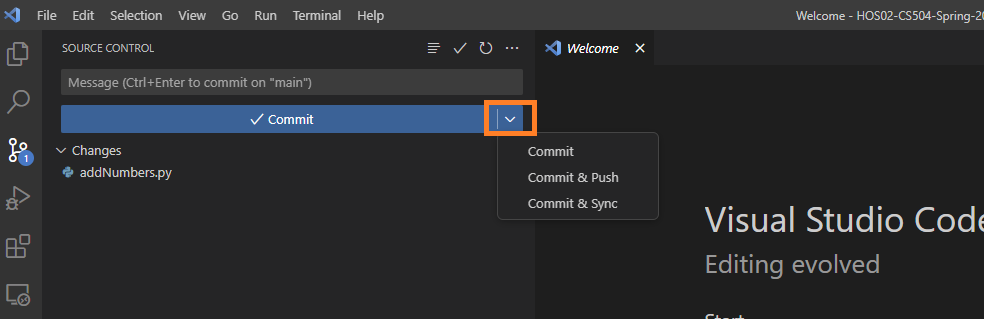
**Opening a Project**

Open a git project (e.g., your HOS02 git repo folder).

Click on Source Control Option and click on '….'







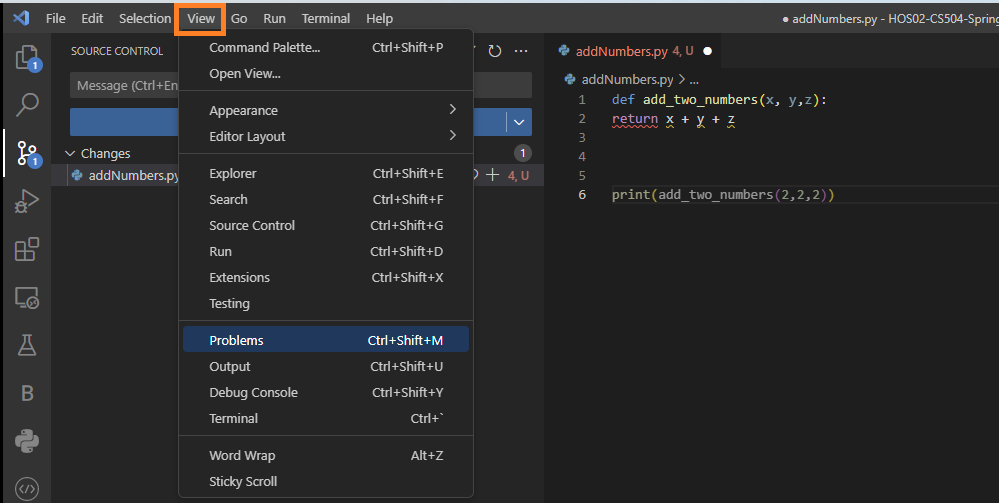
**Error Analysis**

Syntax errors occur when a programmer writes code that does not follow the programming language's rules. This can be due to a missing semicolon, a misspelled word, or an unmatched brace or parenthesis. Syntax errors can prevent the code from running, and the programmer must go back and fix the errors before the code can be executed.

**Finding errors**

Write the following code

Click on View > Problems



The following window displaying the errors will display, add the numbers near the return statement, and it will solve the error.

Text

Description automatically generated

**Linting**

Linting highlights syntactical and stylistic problems in your Python source code, which often helps you identify and correct subtle programming errors or unconventional coding practices that can lead to errors.

Open addNumbers.py and add the code below.

def add\_two\_numbers(x, y,z):

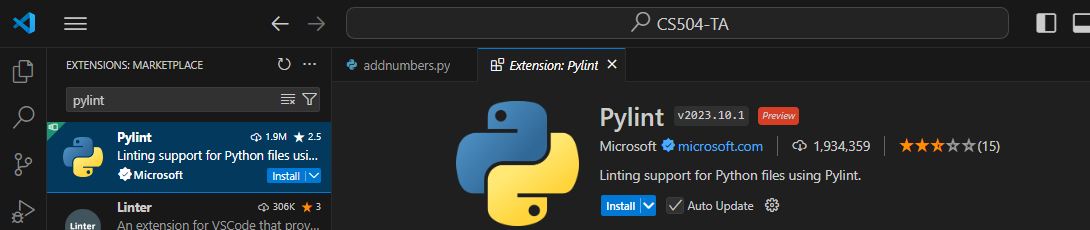
     arr = [0 for i in range(11115)]

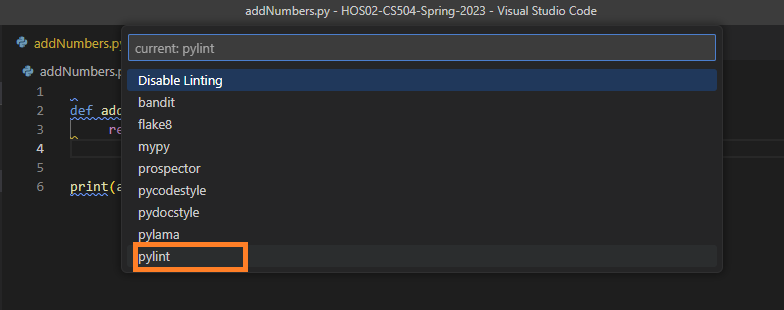
     return x + y + z

print(add\_two\_numbers(2,2,2))

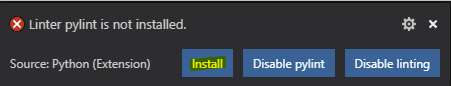
**Enable Linting**: Keep your addNumbers.py python file open in VS code. To enable linters, open the Command Palette (Ctrl+Shift+P) and select the Python: Select Linter command.

[Note: If you cannot find it, you need to install pylint from extensions menu manually]

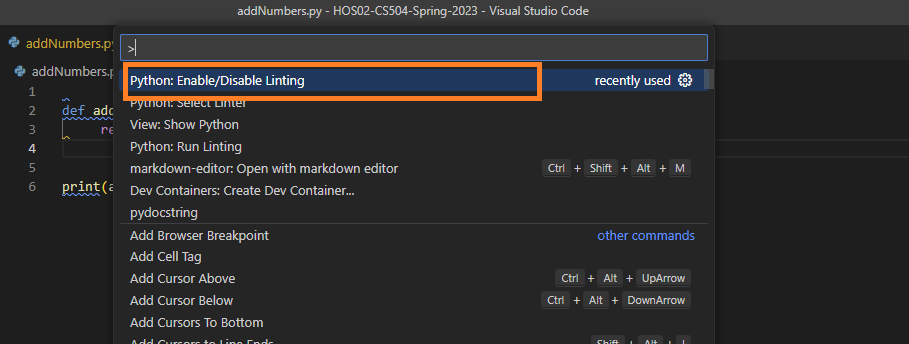


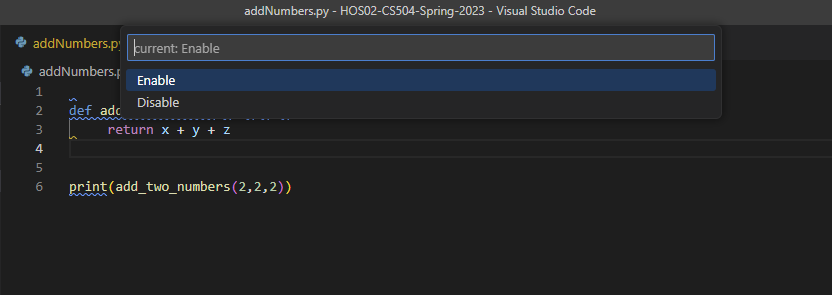


Enabling a linter prompts you to install the required packages in your selected environment for the chosen linter.



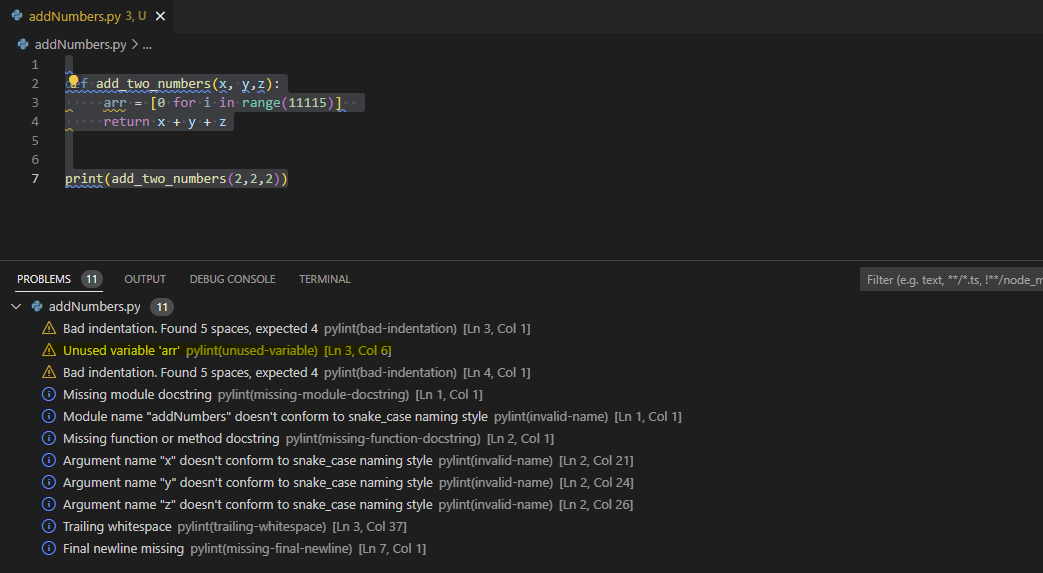
**Enable/Disable Linting:** Linting can easily toggle between enabling and disabling your linter. To switch, open the Command Palette (Ctrl+Shift+P) and select the Python: Enable/Disable Linting command.





To perform linting, open the Command Palette (Ctrl+Shift+P), filter on "linting," and select **Python: Run Linting**. Linting will run automatically when you save a file.

Click on View > Problems



Linting will give all the suggestions related to naming standards, docstring, potential bugs, memory leaks, and any other checks that may be useful. In this case, we have initialized an array and not used it, it is an example of unused variables, and we are wasting memory.

**Push your work to GitHub**

Open the terminal from the VS Code by hitting the control + ~ key and typing the following command:

Run the following commands to push your work to the GitHub repository:

git add

git commit -m "Submission for Module 2"

git push

If you cannot remember, run the command "git status" to check