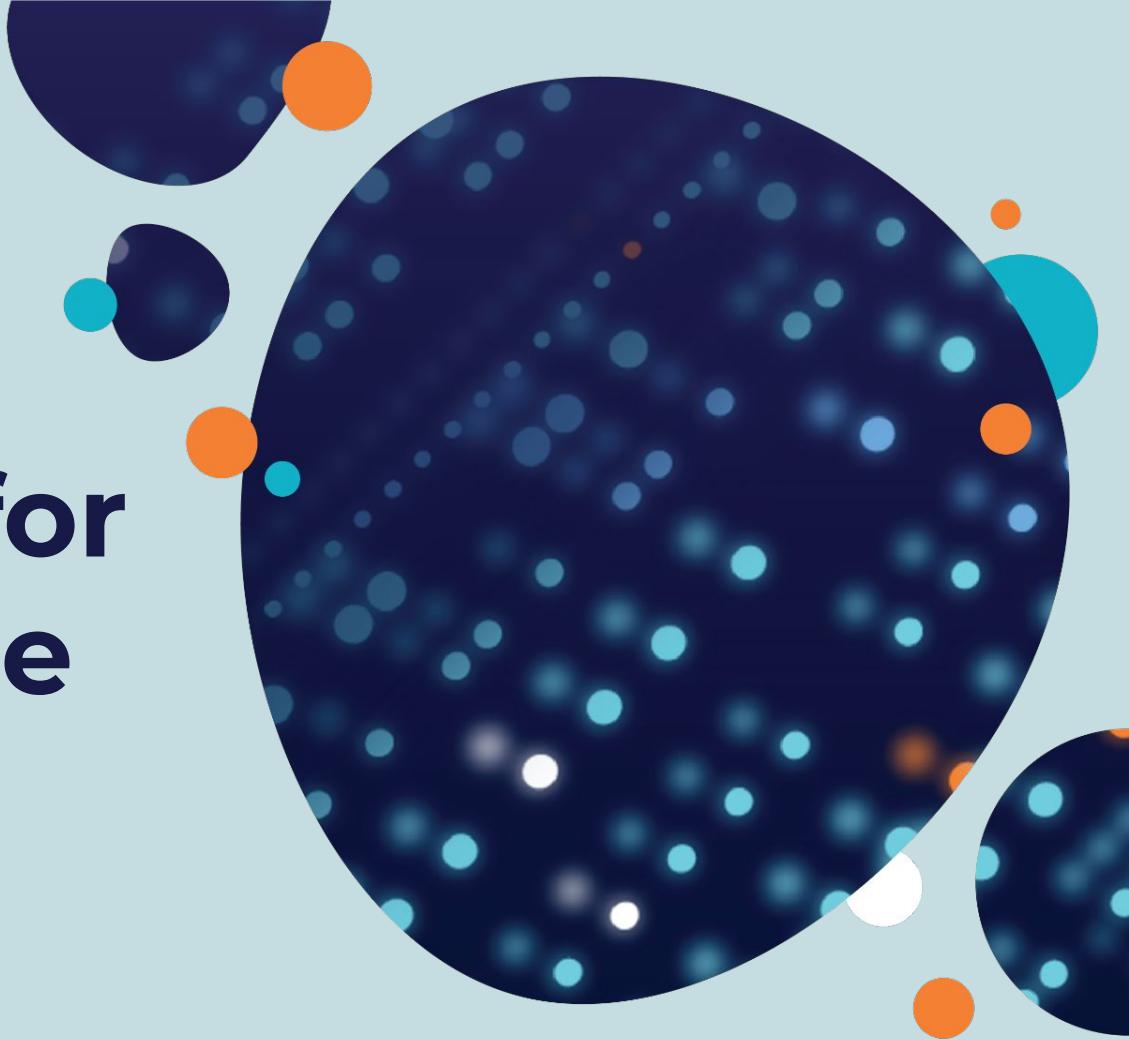




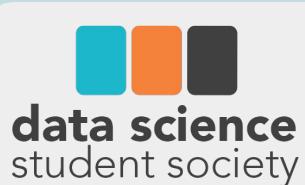
Git/GitHub for Data Science

Software Engineering Domain

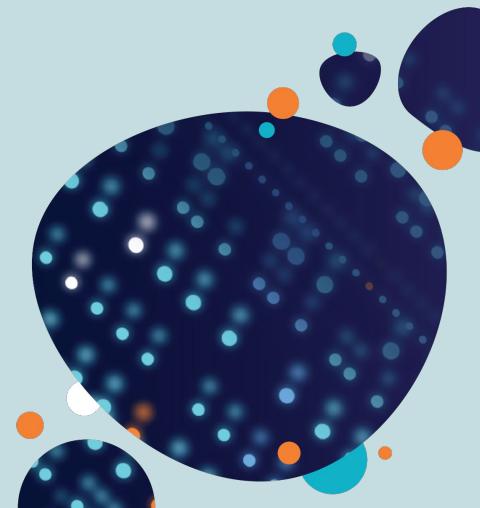
Garrett Lam and Mihir Joshi



Overview



- **Version Control**
- **Git/Github**
- **VS Code**
- **Github Desktop**
- **Git Clone**
- **Git Add/Commit/Push**
- **Git Fetch/Status/Pull**
- **Git Branch/Checkout/Merge**
- **Git Fork/Pull Requests**
- **Git Issues**
- **Scenarios for Data Science**
- **Activity**



What is version control?

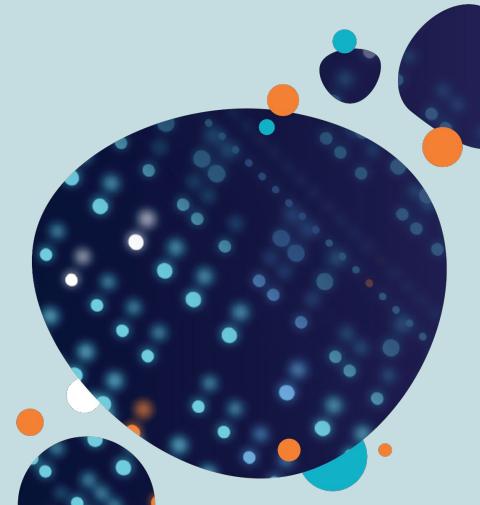
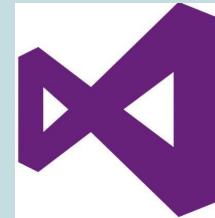
A practice of tracking and managing changes to software code (files)

- **Repository (repo):** a set of files

E.g. Version Control Systems:



Team Foundation Control (TFC)

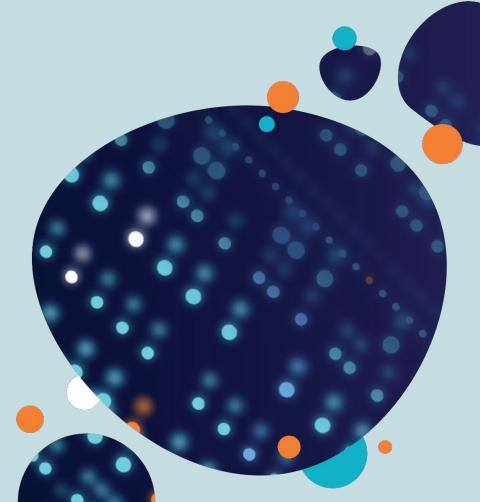


What is Git/Github?



Git: a version control system

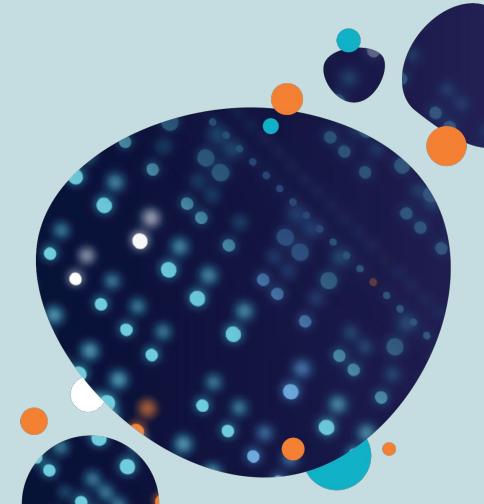
Github: a cloud-based platform where git-based projects are stored (google drive for code)



Importance of Github for Data Science



- Programming happens collaboratively, so it is important to be able to easily share and edit code collectively
- Continuous Integration
- Continuous Deployment
- People can edit files without interference

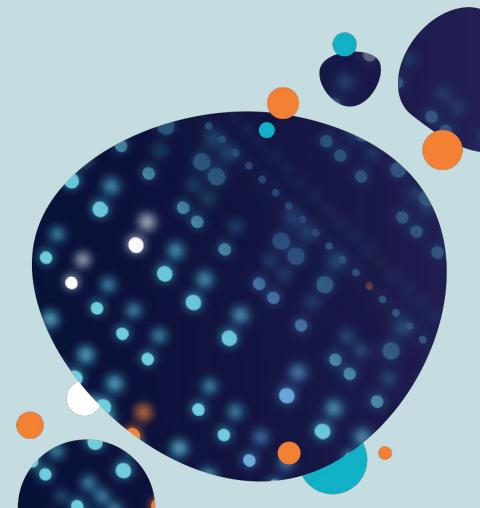


Making a Github Account



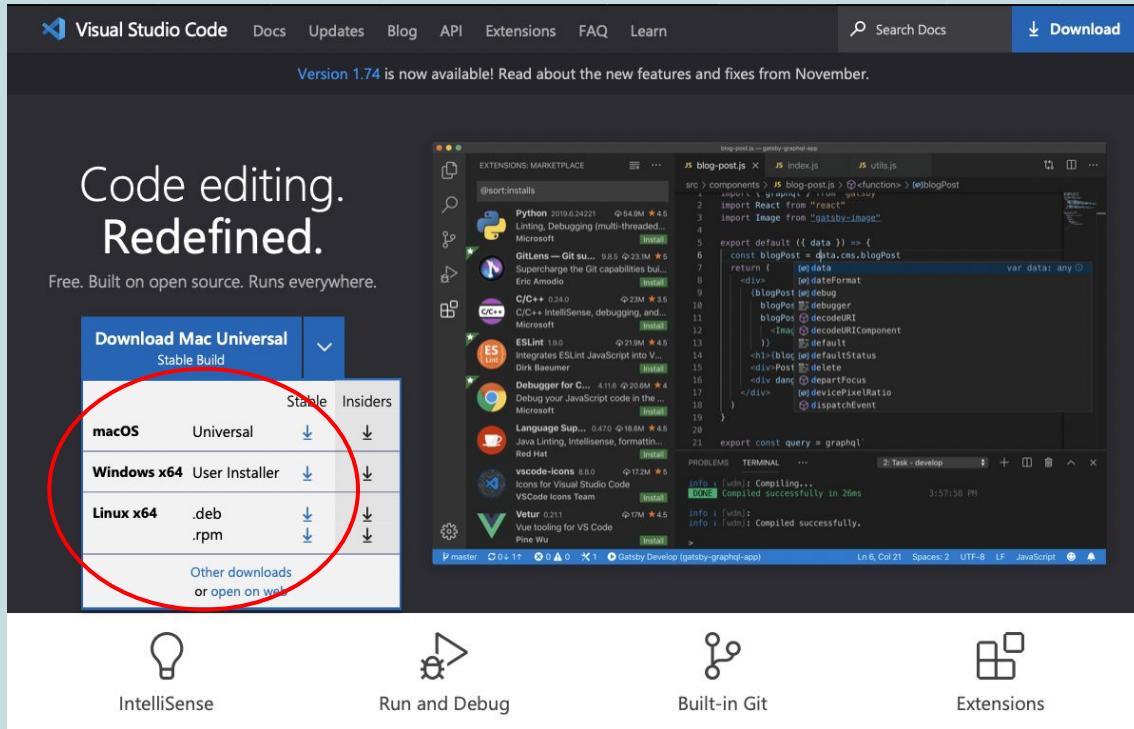
<https://github.com>

The screenshot shows the GitHub homepage with a Mars-themed background image. At the top, there is a navigation bar with links for Product, Solutions, Open Source, and Pricing. To the right of the navigation bar are search and sign-in/sign-up buttons. A red arrow points to the "Sign up" button. Below the navigation bar, the text "Let's build from here" is prominently displayed, followed by a subtitle: "Harnessed for productivity. Designed for collaboration. Celebrated for built-in security. Welcome to the platform developers love." At the bottom of the page, there is a form for entering an email address and a "Sign up for GitHub" button, along with a link to "Start a free enterprise trial". The footer contains a section titled "Trusted by the world's leading organizations" with logos for Stripe, Pinterest, KPMG, Mercedes-Benz, P&G, and TELUS.

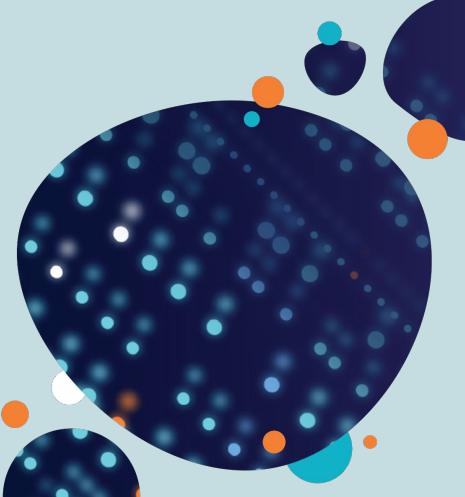


Downloading VS Code

<https://code.visualstudio.com>

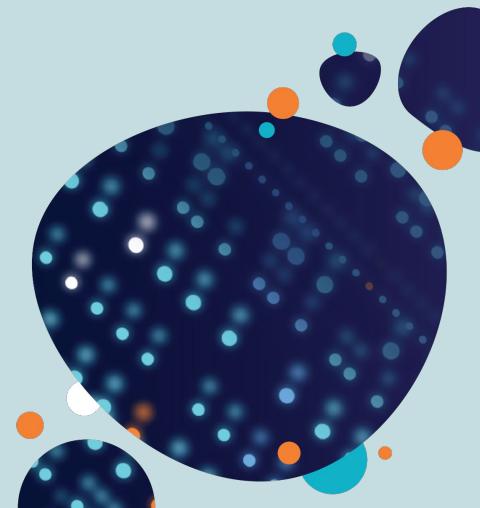
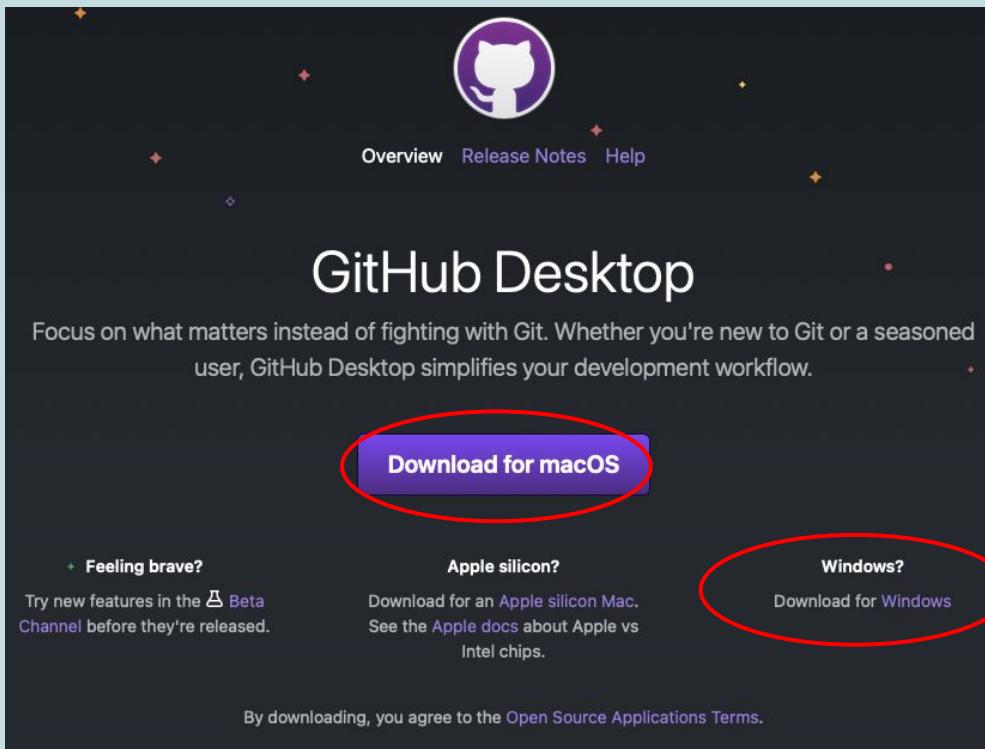


The screenshot shows the official Visual Studio Code website at <https://code.visualstudio.com>. The top navigation bar includes links for Visual Studio Code, Docs, Updates, Blog, API, Extensions, FAQ, and Learn. A prominent "Download" button is visible. A message at the top states, "Version 1.74 is now available! Read about the new features and fixes from November." Below this, a large heading reads "Code editing. Redefined." with the subtitle "Free. Built on open source. Runs everywhere." A red circle highlights the "macOS" download section, which offers "Universal" (Stable and Insiders) and "Windows x64" (User Installer) options. The main content area displays the VS Code interface with code editor, extensions marketplace, terminal, and status bar. At the bottom, there are four icons: "IntelliSense", "Run and Debug", "Built-in Git", and "Extensions".



Downloading GitHub Desktop

<https://desktop.github.com>



Creating a Repository

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * Repository name *

MihirYJoshi / myRepository ✓

Great repository names are short and memorable. Need inspiration? How about [curly-dollop?](#)

Description (optional)

Public Anyone on the internet can see this repository. You choose who can commit.
 Private You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

Add a README file This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None ▾

Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

License: None ▾

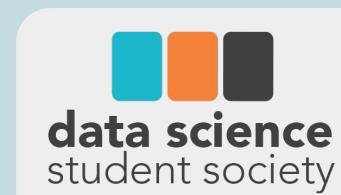
This will set `main` as the default branch. Change the default name in your [settings](#).

You are creating a public repository in your personal account.

Create repository

- **Private:** Access is Restricted to shared people
- **Public:** Anyone can access
- Add a README file
 - High-level description of the Repository
- **Gitignore:** Tells Github to ignore those files when committing

Adding Collaborators



Screenshot of the GitHub repository settings page showing the "Who has access" section.

Who has access

PRIVATE REPOSITORY Only those with access to this repository can view it. [Manage](#)

DIRECT ACCESS 1 has access to this repository. 1 collaborator.

Manage access

Select all [Type](#)

Find a collaborator...

Garrett Lam [garrett-lam](#) • Collaborator [Remove](#)

< Previous Next >

GitHub

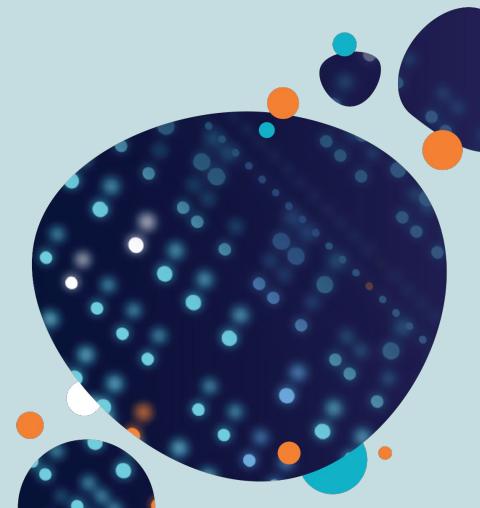
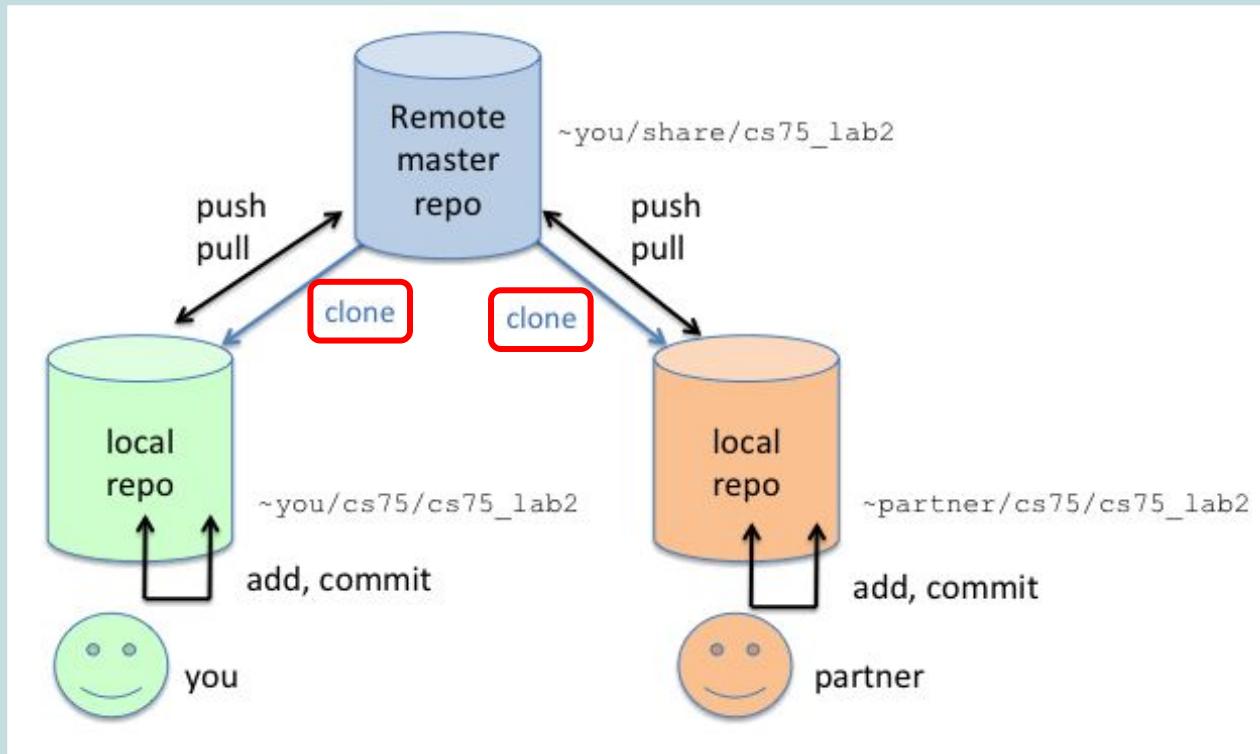
+

@MihirYJoshi has invited you to collaborate on the **MihirYJoshi/myRepository** repository

You can accept or decline this invitation. You can also head over to <https://github.com/MihirYJoshi/myRepository> to check out the repository or visit @MihirYJoshi to learn a bit more about them.

This invitation will expire in 7 days.

[View invitation](#)

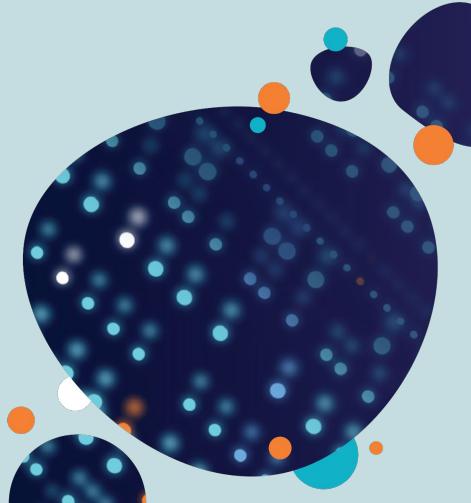


Cloning Repo in VS Code



A screenshot of a GitHub repository page for "myRepository". The page shows one branch ("main") and zero tags. The repository contains two files: "MihirYJoshi Create README.md" and "README.md". A button "Create README.md" is visible. In the bottom right corner of the repository card, the text "myRepository" is displayed.

On the right side of the screen, a "Clone" modal is open. It has tabs for "Local" and "Codespaces" (with "New" in parentheses). Below these are three cloning options: "HTTPS", "SSH", and "GitHub CLI". The "HTTPS" option is selected and highlighted with a red oval. The URL "https://github.com/MihirYJoshi/myRepository" is shown in the input field, with a green checkmark icon indicating it has been copied. A "Copied!" message is displayed next to the checkmark. Below the URL, there is a note: "Use Git or checkout with SVN using the web URL." At the bottom of the modal are buttons for "Open with GitHub Desktop" and "Download ZIP".



SOURCE CONTROL

In order to use git features, you can open a folder containing a git repository or clone from a URL.

 Open Folder

 Clone Repository

To learn more about how to use git and source control in VS Code [read our docs](#).

Get Started

<https://github.com/MihirYJoshi/myRepository.git>

- Clone from URL <https://github.com/MihirYJoshi/myRepository.git>
- Clone from GitHub

Visual Studio Code

Editing evolved

Start

-  New File...
-  Open...
-  Clone Git Repository...

SOURCE CONTROL

In order to use git features, you can open a folder containing a git repository or clone from a URL.

Open Folder

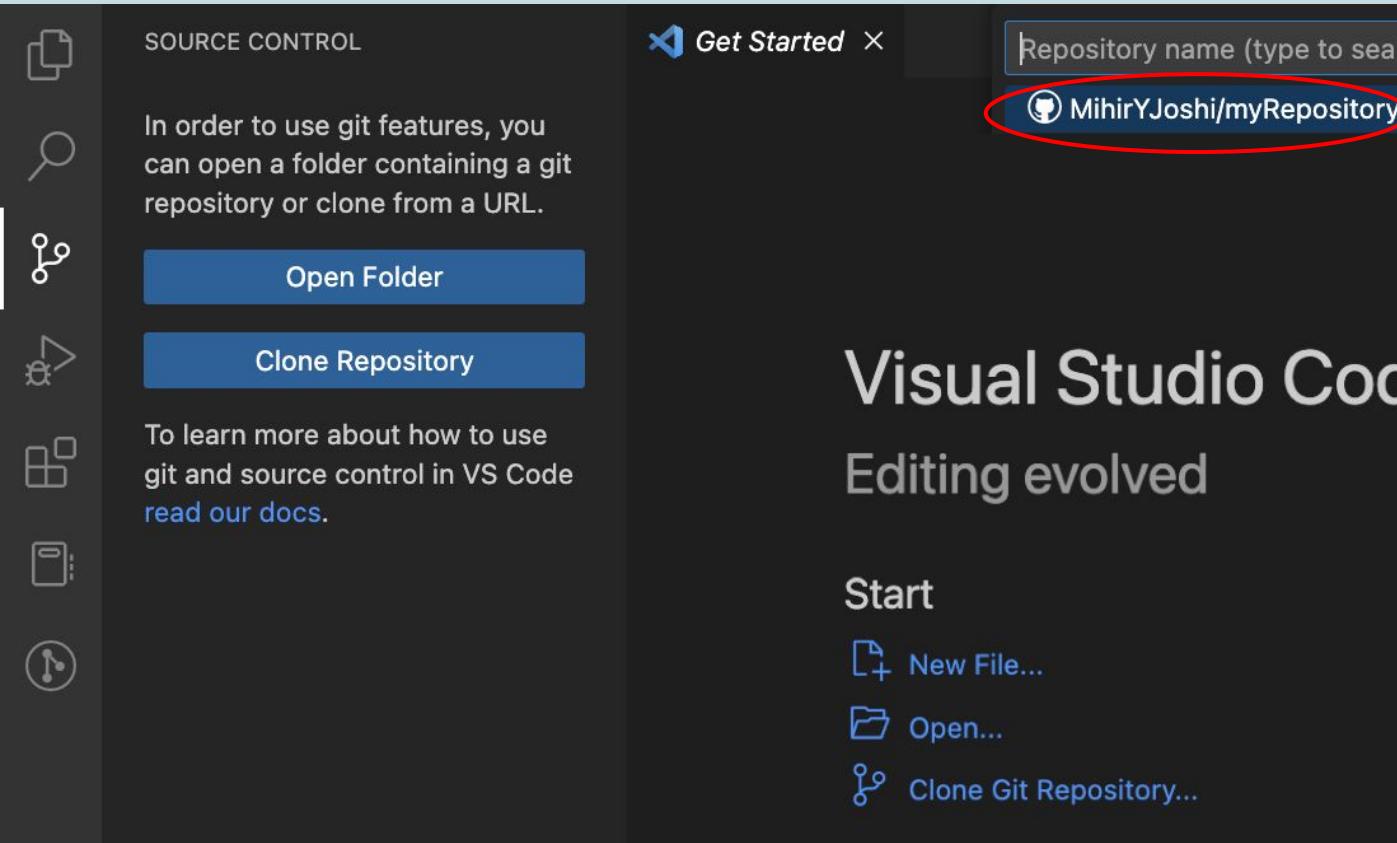
Clone Repository

To learn more about how to use git and source control in VS Code read our docs.

Get Started ×

Repository name (type to search)

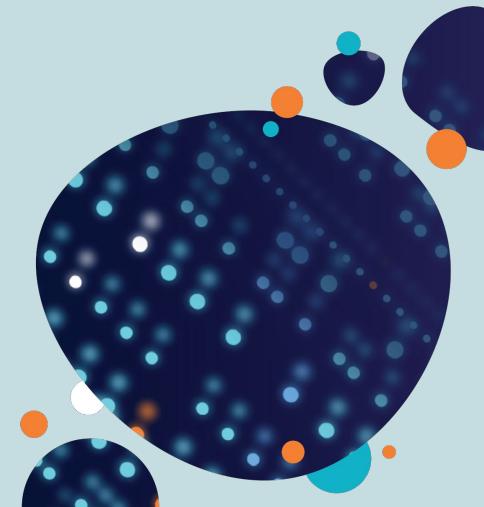
MihirYJoshi/myRepository

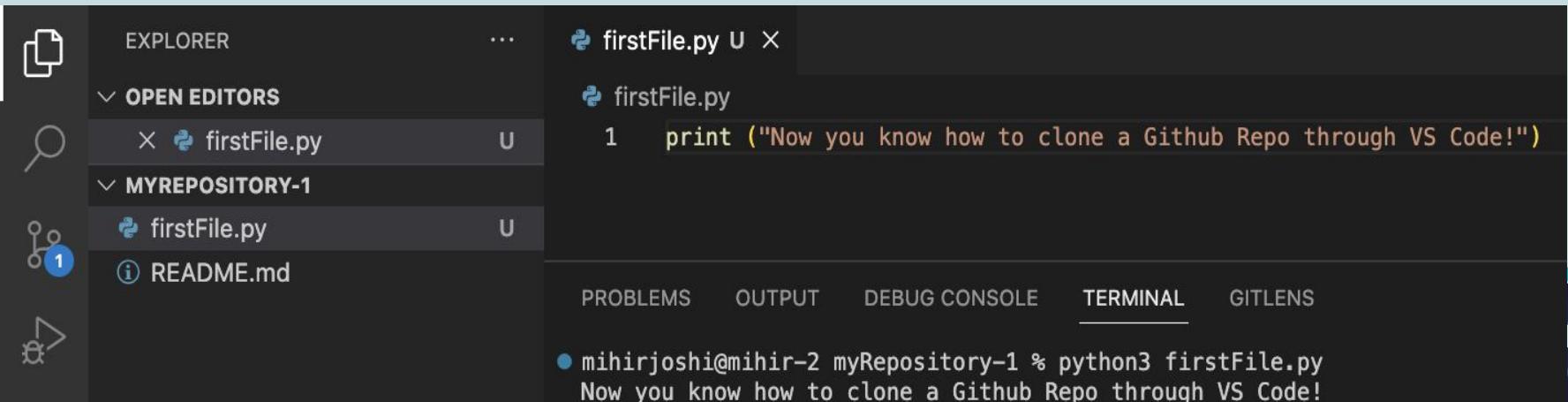


Visual Studio Code
Editing evolved

Start

- New File...
- Open...
- Clone Git Repository...





The image shows a screenshot of the Visual Studio Code (VS Code) interface. On the left is the Explorer sidebar with icons for files, folders, and git. The 'OPEN EDITORS' section shows two files: 'firstFile.py' and 'README.md'. The 'MYREPOSITORY-1' folder is expanded, showing its contents. The main editor area displays the code for 'firstFile.py':

```
1 print ("Now you know how to clone a Github Repo through VS Code!")
```

Below the editor are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'GITLENS'. The 'TERMINAL' tab is active, showing the output of running the script:

```
mihirjoshi@mihir-2 myRepository-1 % python3 firstFile.py  
Now you know how to clone a Github Repo through VS Code!
```

Cloning Repo in Terminal

A screenshot of a GitHub repository page for "myRepository". The page shows one branch ("main") and zero tags. The repository contains two files: "MihirYJoshi Create README.md" and "README.md". A button "Create README.md" is visible. In the top right corner, there is a "Code" button which is highlighted. A modal window titled "Clone" is open, showing three cloning options: "HTTPS", "SSH", and "GitHub CLI". The "HTTPS" option is selected and its URL, "https://github.com/MihirYJoshi/myRepository", is highlighted with a red oval and has a green checkmark icon next to it. A "Copied!" message is displayed above the URL. Below the URL, there is a note: "Use Git or checkout with SVN using the web URL." At the bottom of the modal, there are buttons for "Open with GitHub Desktop" and "Download ZIP".

main ▾ 1 branch 0 tags

MihirYJoshi Create README.md

README.md Create README.md

README.md

myRepository

Local Codespaces New

Clone

HTTPS SSH GitHub CLI

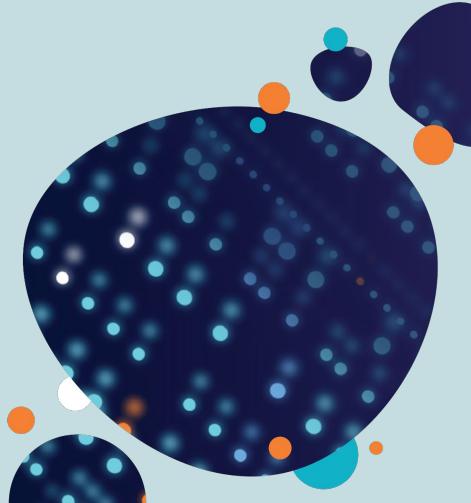
Copied!

https://github.com/MihirYJoshi/myRepository ✓

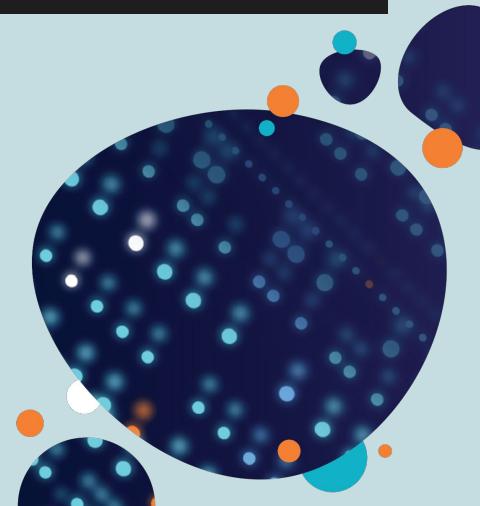
Use Git or checkout with SVN using the web URL.

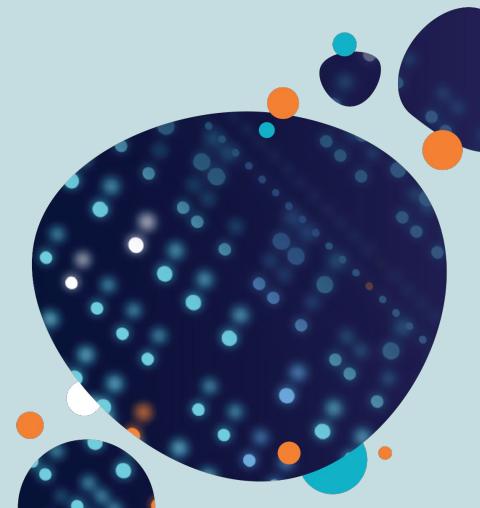
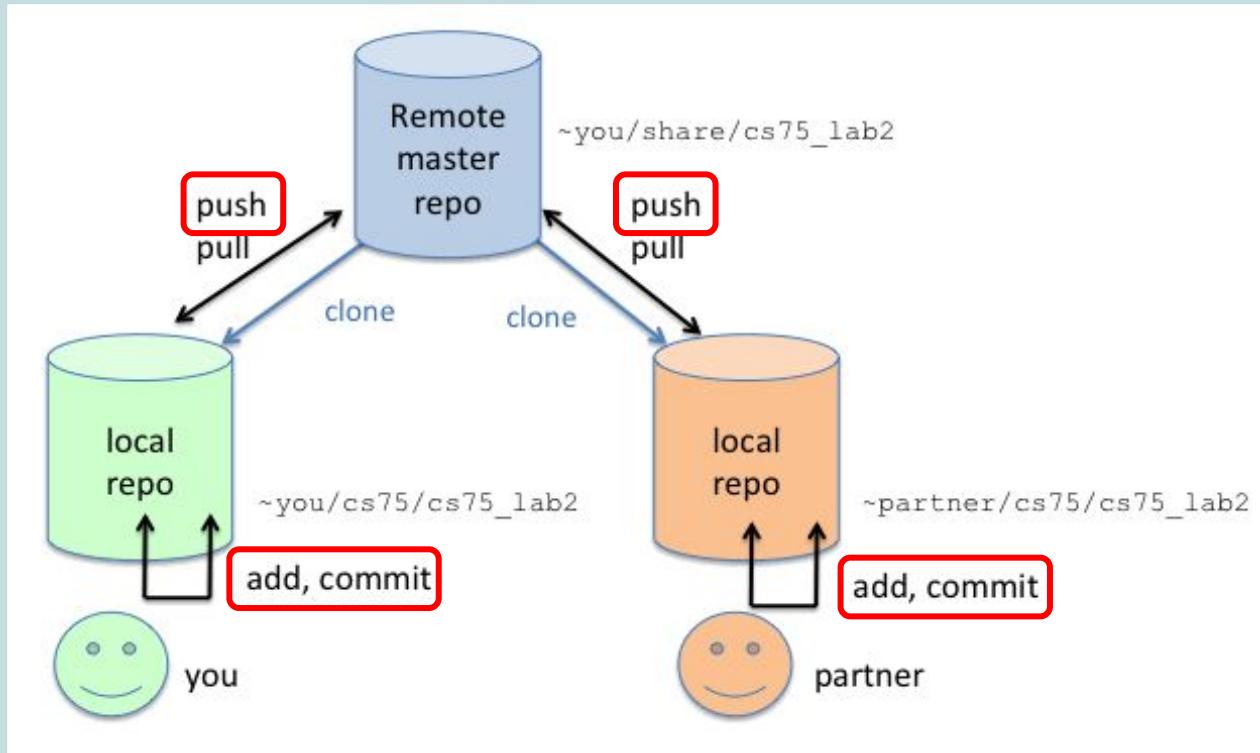
Open with GitHub Desktop

Download ZIP



```
[mihirjoshi@mihir-2 Downloads % git clone https://github.com/MihirYJoshi/repo2.git
Cloning into 'repo2'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
mihirjoshi@mihir-2 Downloads %
```





Git Add/Commit/Push: Terminal

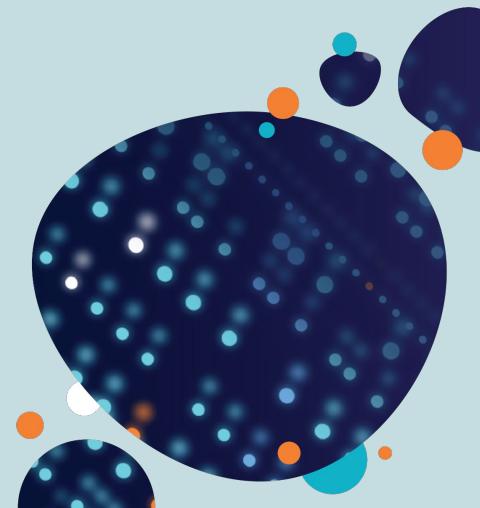


```
thirdFile.py — myRepository

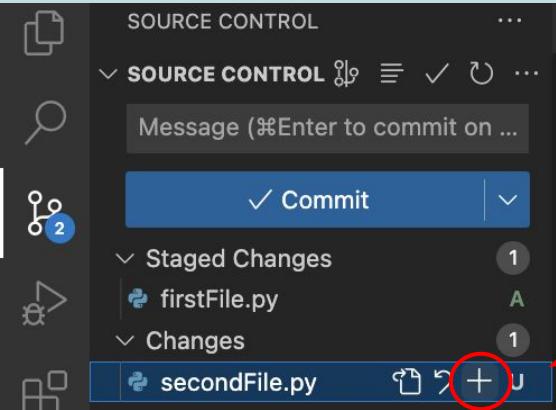
❶ thirdFile.py ×
❷ thirdFile.py
1   print("Third File!")

PROBLEMS    OUTPUT    TERMINAL    GITLENS    DEBUG CONSOLE

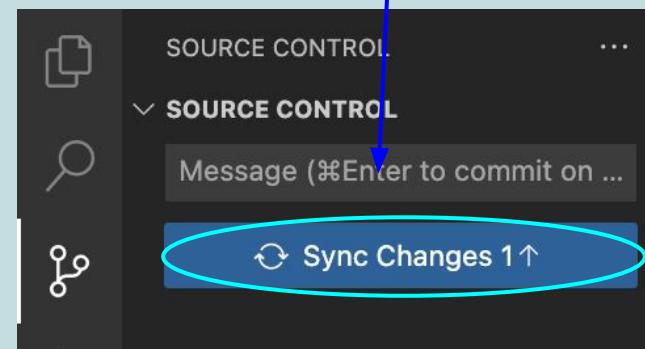
❸ garrettlam@x86_64-apple-darwin13 myRepository % git add thirdFile.py ←
❹ garrettlam@x86_64-apple-darwin13 myRepository % git commit -m "created third file" ←
[main 04e3234] created third file
 1 file changed, 1 insertion(+)
  create mode 100644 thirdFile.py
❺ garrettlam@x86_64-apple-darwin13 myRepository % git push ←
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 316 bytes | 316.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/MihirYJoshi/myRepository.git
  25738cf..04e3234  main -> main
```



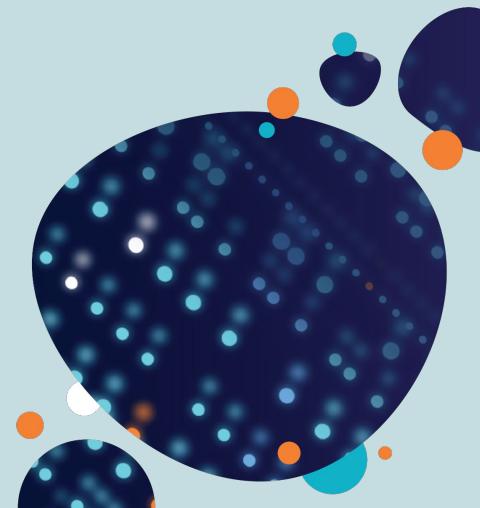
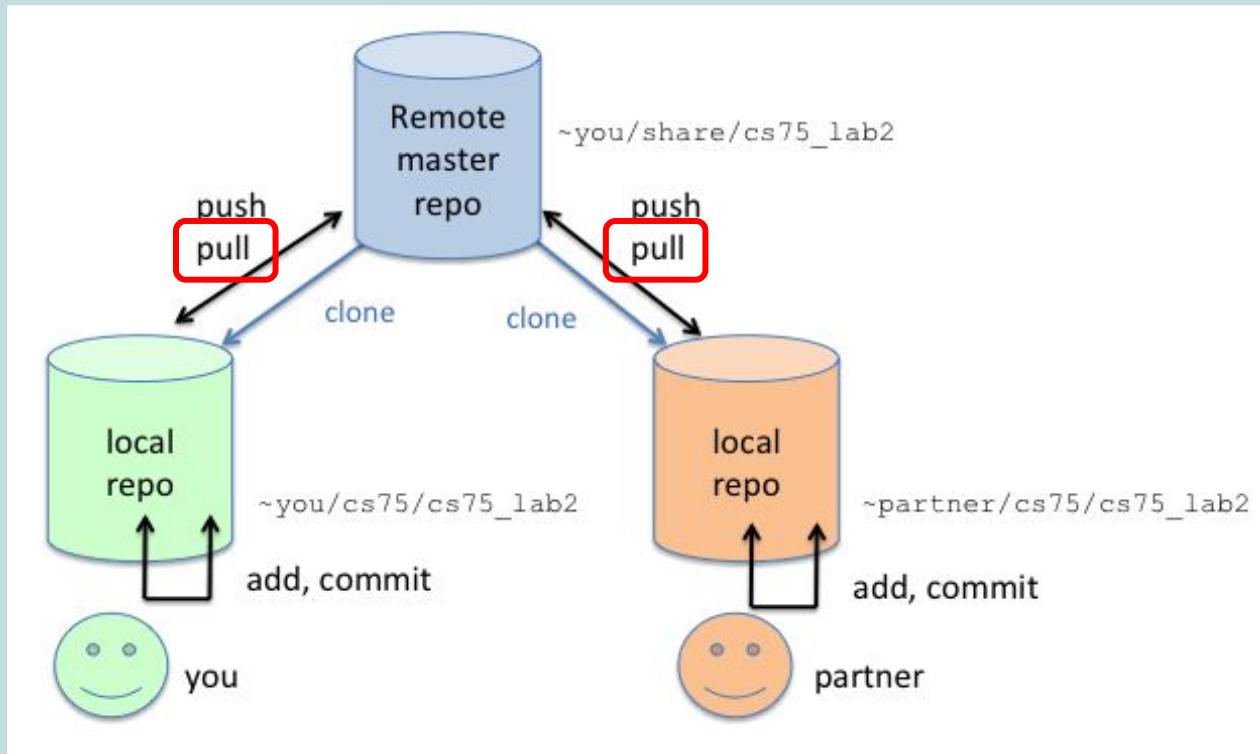
Git Add/Commit/Push: VS Code



```
git > COMMIT_EDITMSG X
git > COMMIT_EDITMSG
      Adding the First an
      1
      2 # Please enter the com
      3 # with '#' will be ign
      4 #
      5 # On branch main
      6 # Your branch is up to
      7 #
      8 # Changes to be committ
      9 # new file: firstF
     10 # new file: secondF
     11 #
```

A screenshot of the Git commit message editor window titled "COMMIT_EDITMSG X". It contains a single line of text: "Adding the First an". Below the editor, the terminal shows the command "git > COMMIT_EDITMSG".

- Click + to Add/Stage Changes
- Click Commit and Write Commit Comments
 - Close Commit File after writing comment
- Sync Changes to Push



Git Fetch/Status/Pull: Terminal

```
● garrettlam@x86_64-apple-darwin13 myRepository % git fetch
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0
Unpacking objects: 100% (3/3), 295 bytes | 295.00 KiB/s, done.
From https://github.com/MihirYJoshi/myRepository
  04e3234..af4ca47  main      -> origin/main
```

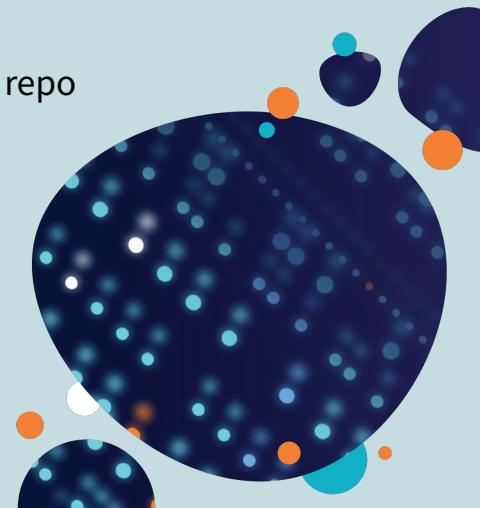
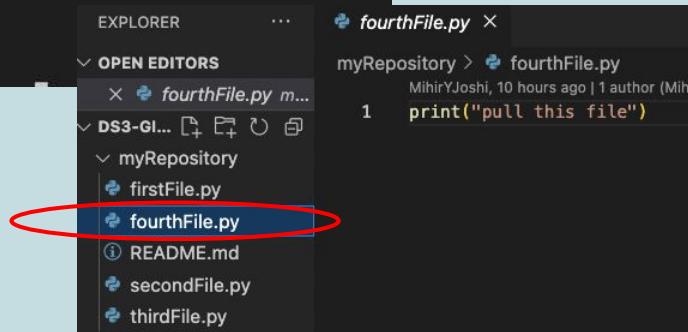
- Downloads contents from remote repo → local repo

```
● garrettlam@x86_64-apple-darwin13 myRepository % git status
On branch main
Your branch is behind 'origin/main' by 1 commit, and can be fast-forwarded.
  (use "git pull" to update your local branch)
```

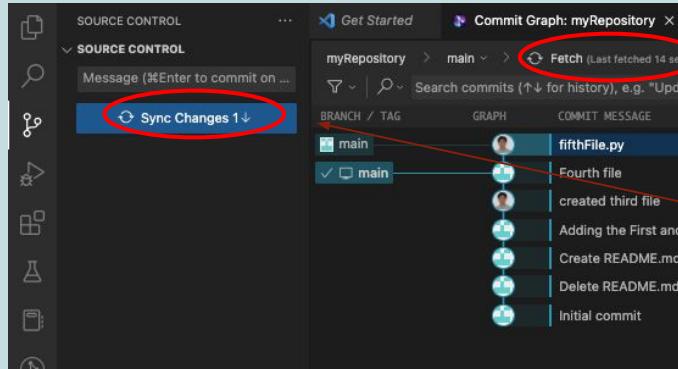
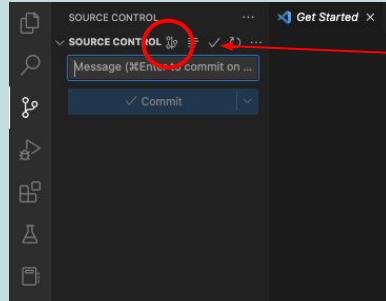
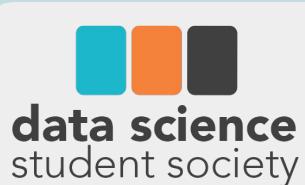
- Checks for differences between remote and local repo

```
nothing to commit, working tree clean
● garrettlam@x86_64-apple-darwin13 myRepository % git pull
Updating 04e3234..af4ca47
Fast-forward
  fourthFile.py | 1 +
  1 file changed, 1 insertion(+)
  create mode 100644 fourthFile.py
```

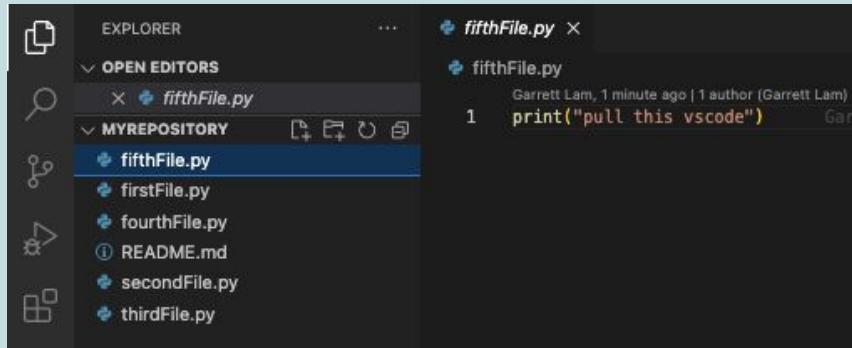
- Update local repo



Git Fetch/Status/Pull : VS Code



- Click to Display Commit Graph
- Click Fetch to “Refresh”
- Click Sync Changes to pull the file(s)



```
fifthFile.py
```

Garrett Lam, 1 minute ago | 1 author (Garrett Lam)

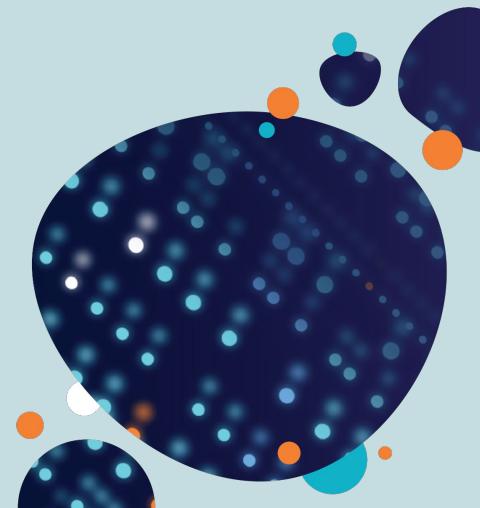
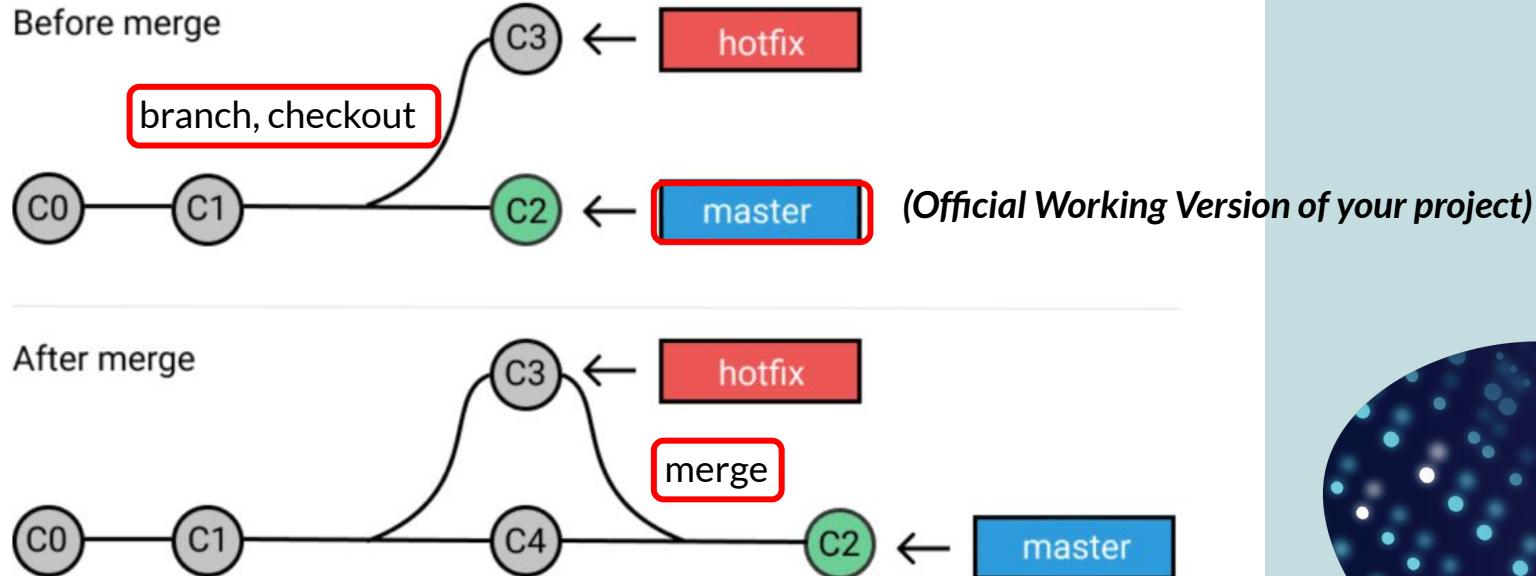
```
1 print("pull this vscode") Gar
```



Scenario:

You are working on a data science project or application as a **developer** and you wanted to create a new feature but you are not sure if it will work

- **Branch** → work on feature → test out feature → **Merge** to main branch



Git Branch/Checkout/Merge: Terminal



```
garrettlam@x86_64-apple-darwin13 myRepository % git branch feature
garrettlam@x86_64-apple-darwin13 myRepository % git branch -a
  feature
* main
  remotes/origin/HEAD -> origin/main
  remotes/origin/main
garrettlam@x86_64-apple-darwin13 myRepository % git checkout feature
Switched to branch 'feature'
garrettlam@x86_64-apple-darwin13 myRepository % git branch -a
* feature
  main
  remotes/origin/HEAD -> origin/main
  remotes/origin/main
```

The screenshot shows the VS Code interface with the following details:

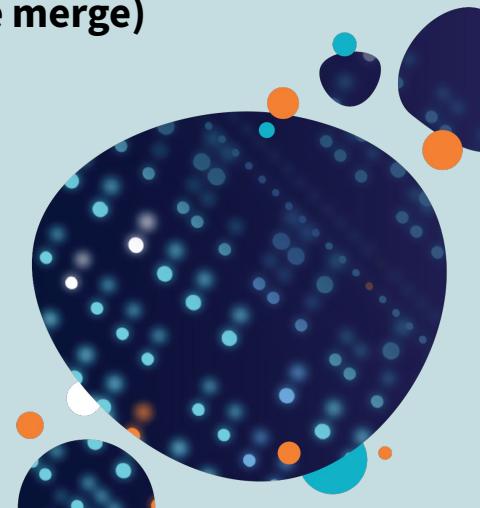
- File Explorer:** Shows files in the repository: fifthFile.py, feature.py (highlighted with a red circle), fourthFile.py, README.md, secondFile.py, and thirdFile.py.
- Terminal:** Shows the command `git status` and its output:

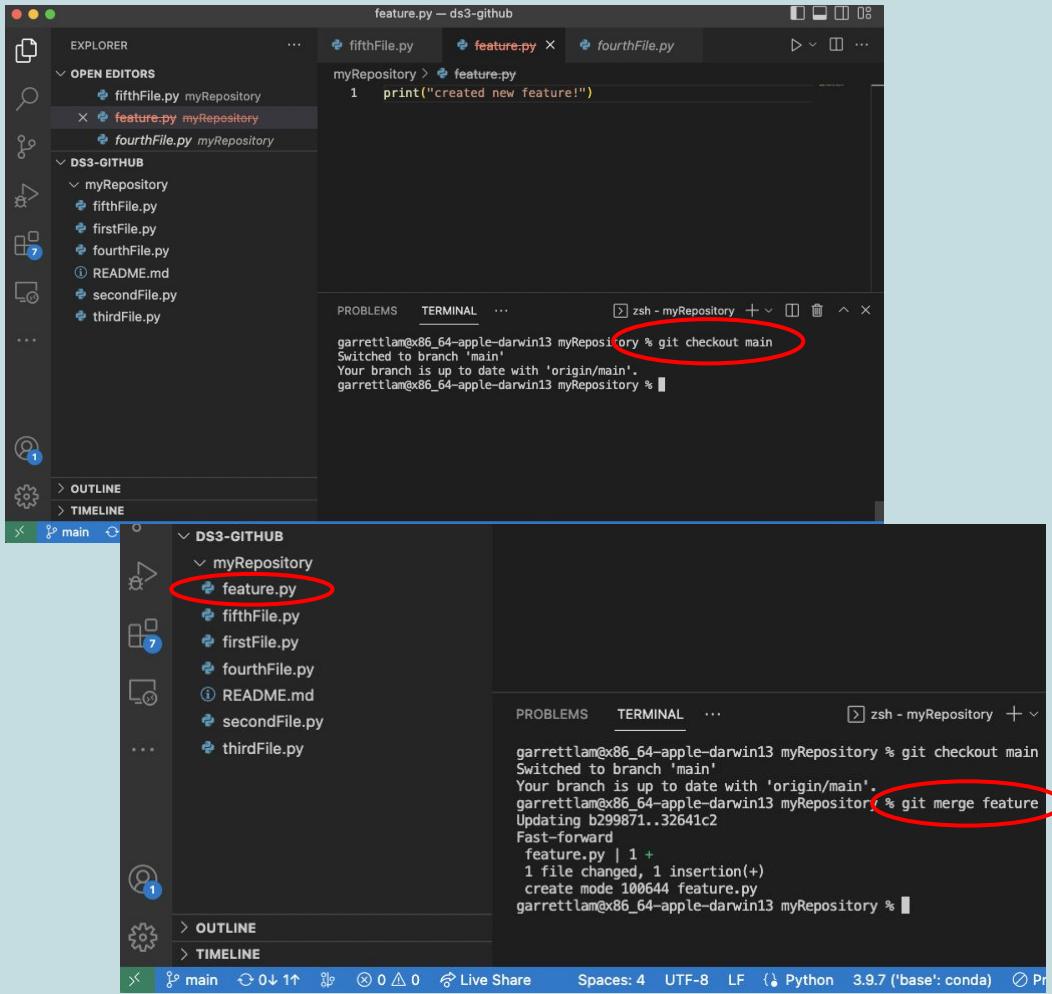
```
garrettlam@x86_64-apple-darwin13 myRepository % git status
On branch feature
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    feature.py

nothing added to commit but untracked files present (use "git add" to track)
```

- **Git checkout:** used to switch between branches

**Creating files on feature branch
will NOT show up on main
branch (before merge)**





The image shows two screenshots of a VS Code interface. Both screenshots have a red circle highlighting specific parts of the terminal output.

Screenshot 1 (Top): The terminal shows the command `git checkout main` being run, which switches the branch to 'main'. The output indicates the branch is up-to-date with 'origin/main'.

```
garrettlam@x86_64-apple-darwin13 myRepository % git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
garrettlam@x86_64-apple-darwin13 myRepository %
```

Screenshot 2 (Bottom): The terminal shows the command `git merge feature` being run. The output shows a fast-forward merge where `feature.py` is updated. The status bar at the bottom indicates the merge was successful with 1 file changed and 1 insertion(+).

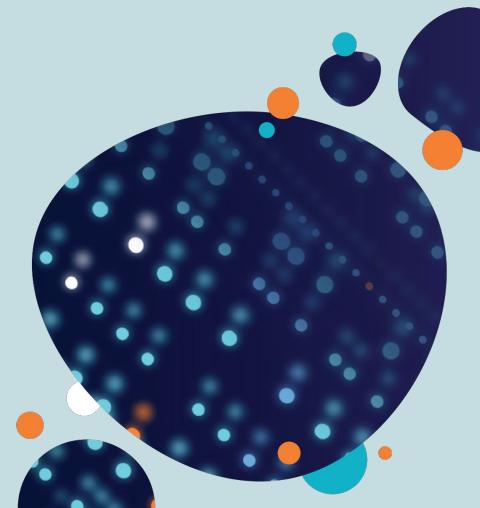
```
garrettlam@x86_64-apple-darwin13 myRepository % git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
garrettlam@x86_64-apple-darwin13 myRepository % git merge feature
Updating b299871..32641c2
Fast-forward
 feature.py | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 feature.py
garrettlam@x86_64-apple-darwin13 myRepository %
```

VS Code UI Elements: The left sidebar shows a tree view of files and repositories. The top editor shows the contents of `feature.py`. The bottom editor shows the terminal output. The status bar at the bottom provides information about the workspace, including the number of changes (0), the live share icon, and the Python version (3.9.7).

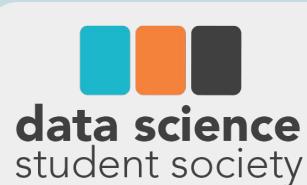
Switch back to main branch to merge!

Feature file not on main branch

- Feature.py not in repo (when switched back to main branch)



Git Branch: VS Code



A screenshot of the Visual Studio Code interface focusing on the Source Control sidebar. A red circle highlights the three-dot menu icon in the top right corner of the sidebar. A green arrow points from the "Create Branch..." option in the "Branch" submenu (also highlighted with a green oval) to a separate input dialog window.

I Studio Code
evolved

SOURCE CONTROL REPOSITORIES

repo2... bran ⌂ ⌂ ✓ 0

SOURCE CONTROL

Message (⌘Enter to commit on ...)

✓ Commit

View & Sort >

- Pull
- Push
- Clone
- Checkout to...
- Fetch
- Commit >
- Changes >
- Pull, Push >
- Branch >**
- Remote >
- Stash >
- Tags >
- Add Co-authors
- Show Git Output

Merge Branch...

Rebase Branch...

Create Branch...

Create Branch From...

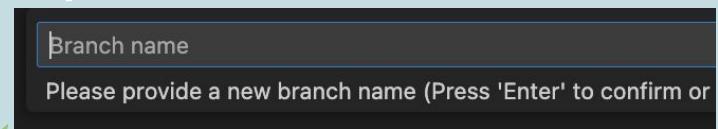
Rename Branch...

Delete Branch...

Publish Branch...

TCG-Tech-Final-Project-2022 ~/Downloads

More...



Publish branch just uploads to Github

A screenshot of the Visual Studio Code interface focusing on the Source Control sidebar. A green arrow points from the "Publish Branch" button in the bottom right corner of the sidebar to the same button in the main interface.

SOURCE CONTROL

SOURCE CONTROL REPOSITORIES

repo2... bran ⌂ ⌂ ✓ 0

SOURCE CONTROL

Message (⌘Enter to commit on ...)

Publish Branch

Git Checkout/Merge: VS Code



Git Merge

The screenshot shows the VS Code interface with the Source Control sidebar open. A yellow arrow points from the "Merge Branch..." option in the context menu to the "Select a branch to merge from" dropdown in the top right.

SOURCE CONTROL REPOSITORIES

repo2... main origin/main origin/main

SOURCE CONTROL

Message (%Enter to commit on ...)

✓ Commit

View & Sort

Pull

Push

Clone

Checkout to...

Fetch

Commit

Changes

Pull, Push

Branch

Remote

Stash

Tags

Add Co-authors

Show Git Output

Merge Branch...

Rebase Branch...

Create Branch...

Create Branch From...

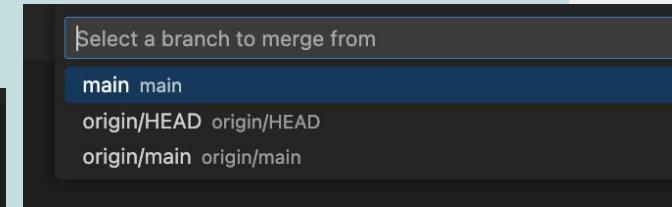
Rename Branch...

Delete Branch...

Publish Branch...

TCG-Tech-Final-Project-2022 ~/Downloads

More...



Git Checkout

feature.py — ds3-github

EXPLORE Select a branch or tag to checkout

OPEN E + Create new branch...

+ Create new branch from...

Checkout detached...

i feature 32641c20

DS3-GH feature 32641c20

myR main 32641c20

origin/HEAD Remote branch at 32641c20

Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/MihirYJoshi/myRepository
y.git
b299871..32641c2 main -> main
garrettlam@x86_64-apple-darwin13 myRepository

OUTLINE

TIMELINE

main

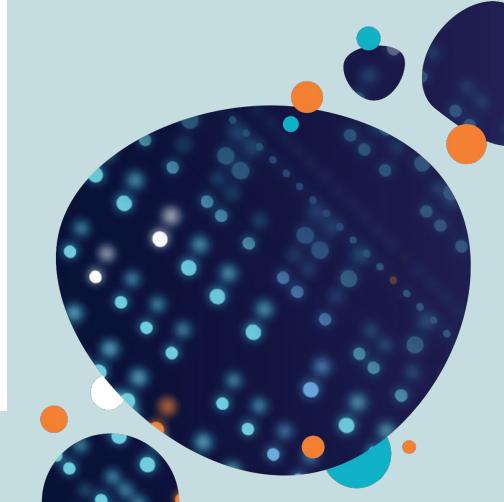
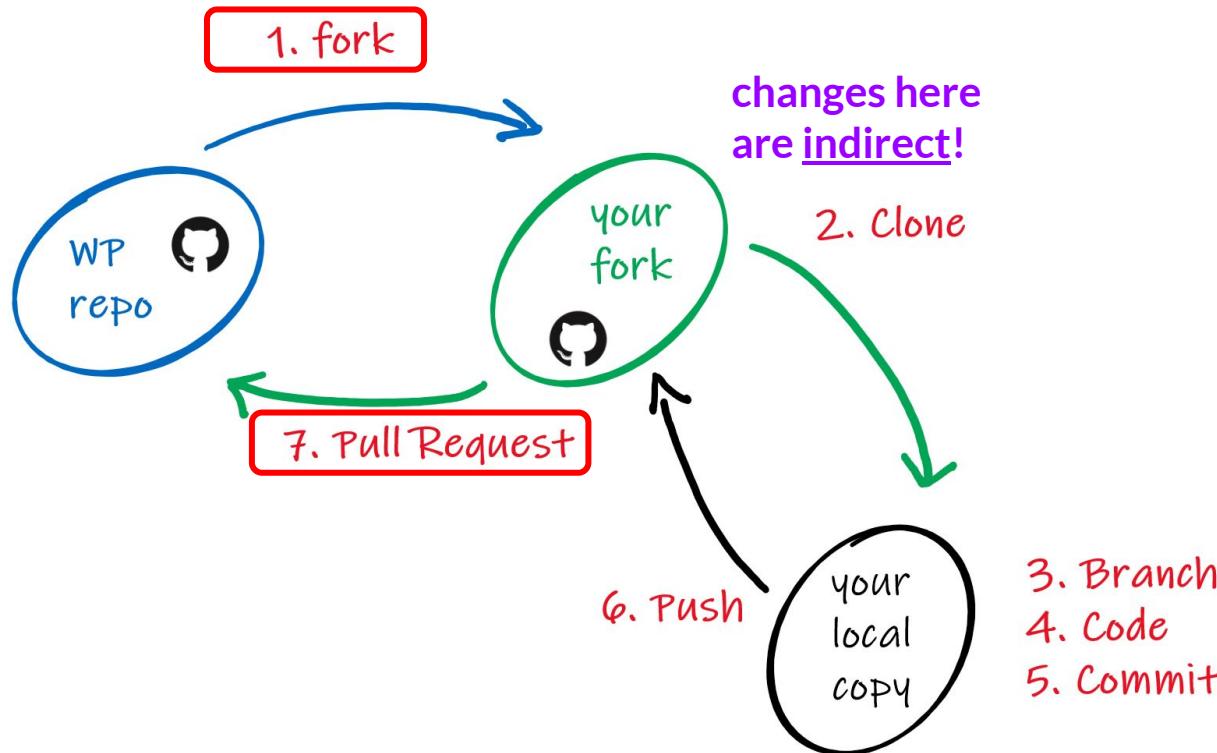
Live Share

Prettier

Scenario:

Mihir: created an AI chatbot (like ChatGPT) to be used by public

Garrett (random person on internet): wanted to improve it



Git Fork



Garrett POV:

A screenshot of a GitHub repository page for the user 'MihirYJoshi' named 'forkMe'. The page shows a single commit from 'Create hi.txt' made 1 hour ago. The top navigation bar includes 'Watch 1', 'Fork 0', and 'Star 0'. A red circle highlights the 'Fork 0' button, which has an arrow pointing to it from the 'Garrett POV:' text. The repository details include an 'About' section with 'No description, website, or topics provided.', '0 stars', '1 watching', and '0 forks'. There are sections for 'Releases' (No releases published) and 'Packages' (No packages published).

Git Fork: making a copy of someone's repository (that you are **NOT** a collaborator on)

A screenshot of the GitHub 'Create a new fork' dialog. It starts with a heading 'Create a new fork' and a description: 'A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.' The form fields include 'Owner' (garrett-lam) and 'Repository name' (forkMe). Below the form, a note says: 'By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.' Under 'Description (optional)', there is a text input with 'This is a copy of Mihir's repo'. A checked checkbox says 'Copy the main branch only' with the note: 'Contribute back to MihirYJoshi/forkMe by adding your own branch. Learn more.' At the bottom, a red circle highlights the 'Create fork' button.

Garrett POV: write code to improve Mihir's code on forked repo



Git Clone

A screenshot of a GitHub repository page for 'garrett-lam/forkMe'. The repository is public and was forked from 'MihirYJoshi/forkMe'. The main branch is up-to-date with 'MihirYJoshi/forkMe:main'. A context menu is open over a file named 'hi.txt', showing options like 'Clone', 'HTTPS', 'SSH', 'GitHub CLI', 'Open with GitHub Desktop', and 'Download ZIP'.

Git Add/Commit/Push

A screenshot of a GitHub Codespace interface titled 'hi.txt — forkMe'. The terminal window shows the following command sequence:

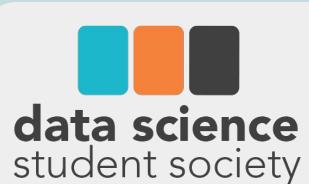
```
garrettlam@x86_64-apple-darwin13 forkMe % git add .
garrettlam@x86_64-apple-darwin13 forkMe % git commit -m "editing on forked repo"
[main 9652a64] editing on forked repo
 1 file changed, 2 insertions(+)
garrettlam@x86_64-apple-darwin13 forkMe % git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 308 bytes | 308.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/garrett-lam/forkMe.git
 31d8bb5..9652a64 main -> main
garrettlam@x86_64-apple-darwin13 forkMe %
```

Git Pull Request

A screenshot of a GitHub repository page for 'garrett-lam / forkMe'. The URL 'garrett-lam / forkMe' is highlighted with a red circle. Below it, the repository is described as 'forked from MihirYJoshi/forkMe'. The main navigation bar includes 'Pin', 'Watch 0', 'Fork 1', 'Star 0', 'Code', 'Pull requests' (which is underlined), 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. Below the navigation is a search bar with the query 'is:pr is:open'. A green 'New pull request' button is also circled in red.

A screenshot of a GitHub repository page for 'MihirYJoshi / forkMe'. The URL 'MihirYJoshi / forkMe' is highlighted with a red circle. The main navigation bar includes 'Code', 'Issues 1', 'Pull requests' (underlined), 'Actions', 'Projects', 'Security', and 'Insights'. Below the navigation is a section titled 'Comparing changes' with the instruction 'Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks.' A dropdown menu for selecting branches is circled in red. A green 'Create pull request' button is located at the bottom right of this section.

"I want my improvements to show up on the original repo!"



Pull Request: similar to Git merge but needs to be approved by original repo owner

A screenshot of a GitHub repository page for 'MihirYJoshi / forkMe'. The URL 'MihirYJoshi / forkMe' is highlighted with a red circle. The main navigation bar includes 'Code', 'Issues 1', 'Pull requests' (underlined), 'Actions', 'Projects', 'Security', and 'Insights'. Below the navigation is a section titled 'Open a pull request' with the instruction 'Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.' A dropdown menu for selecting branches is shown. A green 'Create pull request' button is circled in red at the bottom right.

Mihir's POV: review changes → accepting pull request



ce
ety

Pull requests 1

Label issues and pull requests for new contributors

Now, GitHub will help potential first-time contributors discover issues labeled with

Filters is:pr is:open

1 Open 0 Closed

Author Label Projects

editing on forked repo #2 opened now by garrett-lam

ProTip! Notify someone on an issue with a mention, like: @MihirYJoshi

A screenshot of the GitHub Pull Requests page. A red circle highlights the 'Pull requests' tab at the top. Another red circle highlights the 'editing on forked repo' label for a specific pull request.

editing on forked repo #2

garrett-lam wants to merge 1 commit into MihirYJoshi:main from garrett-lam:main

Conversation 0 Commits 1 Checks 0 Files changed 1

garrett-lam commented 1 minute ago
Replied with Hi Mihir!

editing on forked repo

Looks like you got your first contribution! Visit your community profile to learn more about recommended open source practices.

Add more commits by pushing to the main branch on garrett-lam/forkMe.

Require approval from specific reviewers before merging
Branch protection rules ensure specific people approve pull requests before they're merged.

Continuous integration has not been set up
GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch
Merge can be performed automatically.

Merge pull request

You can also open this in GitHub Desktop or view command line instructions.

A screenshot of the GitHub pull request details page. A red circle highlights the 'Merge pull request' button. Below it, a message indicates the pull request has been merged.

Conversation 0 Commits 1 Checks 0 Files changed 1

garrett-lam commented 1 minute ago
Replied with Hi Mihir!

editing on forked repo

MihirYJoshi merged commit 4ba34aa into MihirYJoshi:main now

9652a64

Revert

A screenshot of the GitHub commit history page. A red circle highlights the message 'MihirYJoshi merged commit 4ba34aa into MihirYJoshi:main now'. A large red swoosh highlights the entire commit message area.

Git Issues

Way for non-collaborators to let **original repo owner** they have issues with using their tool OR if there are any bugs

This screenshot shows the GitHub Issues interface. At the top, there's a navigation bar with links for Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The 'Issues' link is highlighted with a red circle. Below the navigation is a header with a user icon and the word 'Issue'. There are two tabs: 'Write' (selected) and 'Preview'. The main area contains the text 'This is an Issue' and an illustration of a hand holding a sign that says 'ISSUE' next to a traffic cone. A note at the bottom says 'Styling with Markdown is supported'. At the bottom right is a green button labeled 'Submit new issue'.

This screenshot shows the GitHub Issues interface with several annotations:

- A red circle highlights the 'Issues' link in the top navigation bar.
- A red circle highlights the 'Submit new issue' button at the bottom right of the issue creation form.
- A red circle highlights the 'Comment' button at the bottom right of the issue detail page.
- A red circle highlights the 'Comment' section of the issue detail page, which shows a comment from 'garrett-lam' with the text 'How to Fix Bug:'
- A red circle highlights the 'Close with comment' button at the bottom right of the issue detail page.
- A red circle highlights the 'Collaborator' status of the user 'garrett-lam' in the issue detail page.
- A red circle highlights the 'garrett-lam closed this as completed now' message at the bottom of the issue detail page.

The main content of the page includes:

- The title 'This is an Issues #1'.
- The status 'Open' (now 'Closed') and the author 'MihirYJoshi'.
- A comment from 'MihirYJoshi' stating 'commented 5 minutes ago - 0 comments'.
- An illustration of a hand holding a sign that says 'ISSUE' next to a traffic cone.
- A comment from 'garrett-lam' with the text 'How to Fix Bug:'.
- A note to attach files by dragging & dropping, selecting or pasting them.
- A 'Close with comment' button.
- A 'Comment' button.
- The status 'Owner' and a three-dot menu.
- The title 'This is an Issues #1' again.
- The status 'Closed' (now 'Open') and the author 'MihirYJoshi'.
- A comment from 'MihirYJoshi' stating 'commented 7 minutes ago'.
- An illustration of a hand holding a sign that says 'ISSUE' next to a traffic cone.
- A comment from 'garrett-lam' with the text 'commented now'.
- A note to attach files by dragging & dropping, selecting or pasting them.
- The status 'Collaborator' and a three-dot menu.
- The message 'garrett-lam closed this as completed now'.

Scenarios for Data Science



The screenshot shows a Jupyter Notebook interface with the title "Final Project". The notebook contains two sections: "Comparing Demographics" and "DC vs. Marvel". The "DC vs. Marvel" section includes a large "MARVEL VERSUS DC" graphic featuring various superhero and supervillain characters from both universes.

Final Project
DSC 10, Fall 2022

Checkpoint due **Thursday, November 17th at 11:59PM**
Final submission due **Tuesday, November 29th at 11:59PM**

Welcome to the Final Project!

Deadlines This assignment has two deadlines:

- The first deadline is **Thursday, November 17th at 11:59PM**. This is a checkpoint by which you must submit Sections 6 and 7 of the project. You may not use any days to extend the deadline.
- The second deadline is **Tuesday, November 29th at 11:59PM**. This is the final submission date. If you have not submitted the checkpoint by the first deadline and using skip days, skip days will be deducted from each person's skip days available. If you have not submitted the checkpoint by the first deadline and using skip days, skip days will be deducted from each person's skip days available.

10% of your grade will come from your score on the checkpoint, and 90% of your grade will come from your score on the final submission. See the Edition post titled "Final Project Release" for more details.

Partners You are very much encouraged to form a team to work through the project with. If you work with a partner, you are required to contribute to both parts of the project. [Laptop guidelines](#) are available on the course website.

Rules Don't share your code with anyone but your partner. You are welcome to ask clarifying questions with your student, but don't share the answers. The experience of solving the problems in this project will prepare you for the final exams and your future in data science.

Support You are not alone! Come to our office hours, post on EdStem, and talk to your classmates. If you want to add details of your solution to a problem, make a private EdStem post and staff will try to assist. All of the concepts necessary for this project lecture and can be found in the [lectures](#) and [background notes](#). If you are stuck on a particular problem, reading through the relevant textbook section or referring the Jupyter notebook from lecture will often clarify the concept.

Tests The tests won't actually tell you what your answer is correct. More often, they help catch basic mistakes. It is up to you to ensure that your answer is correct. Additional tests will be applied to verify the correctness of your submission in order to assign your final grade.

Advice First, start early. As you know from the Midterm Project, projects are complex and time-consuming. Second, develop your answers incrementally. To perform a complicated task, break it into steps, perform each step on its own, give a new name to each step that each intermediate result is what you expect. You can add any additional name or function you want to the provided cells, and you can add additional cells as needed. If you run into memory issues, consider using smaller test resources. Then only ever code to tests. For `for` loops to repeat `i`, similarly, for defining functions, first write the code and then use the `assert` command for a single `for` loop. That way, you can put that code inside a function and change the input to be whatever.

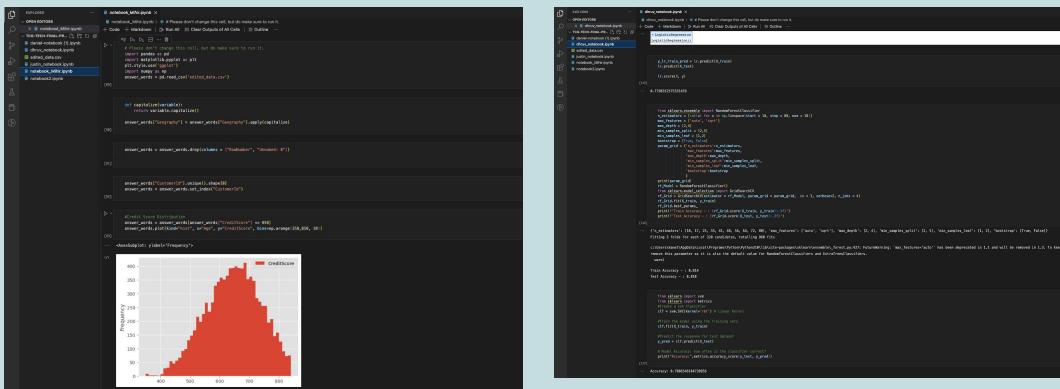
Random Seeds This project uses random seeds, as described in [Lecture 8](#). When we provide a random seed, we set the internal configuration of Python's random number generator so that it produces the same results every time. This makes it easier to grade your assignments.

To understand how random seeds work, just do something like this to see a `random` value:

```
Don't change  
print(random)  
print("The same result each time you run the cell")
```

Long Simulations If you are doing tasks that take more than a few minutes to run, you are probably doing something wrong. You can sometimes speed things up by making sure you have a Dataframe of only the rows and columns you need to do your analysis, which should be significantly faster. If you are doing simulations, think possibly, as to avoid using additional `for` loops or creating many iterations. If you do a `for` loop, make it a `list comprehension` or, if possible, use a `map` function. If you have already nested the loops on one level, consider nesting them again.

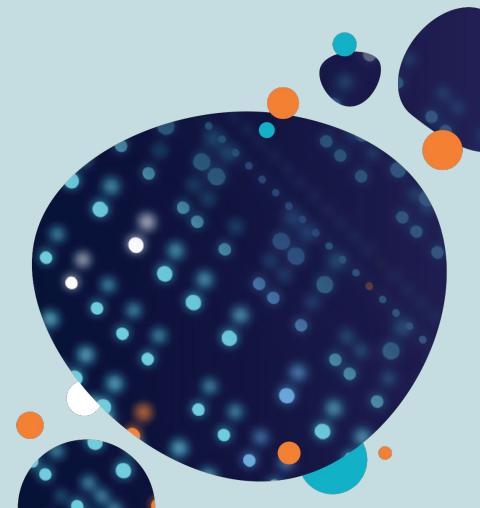
- Work on Python Notebooks in VS Code
 - .ipynb files are Jupiter Notebooks



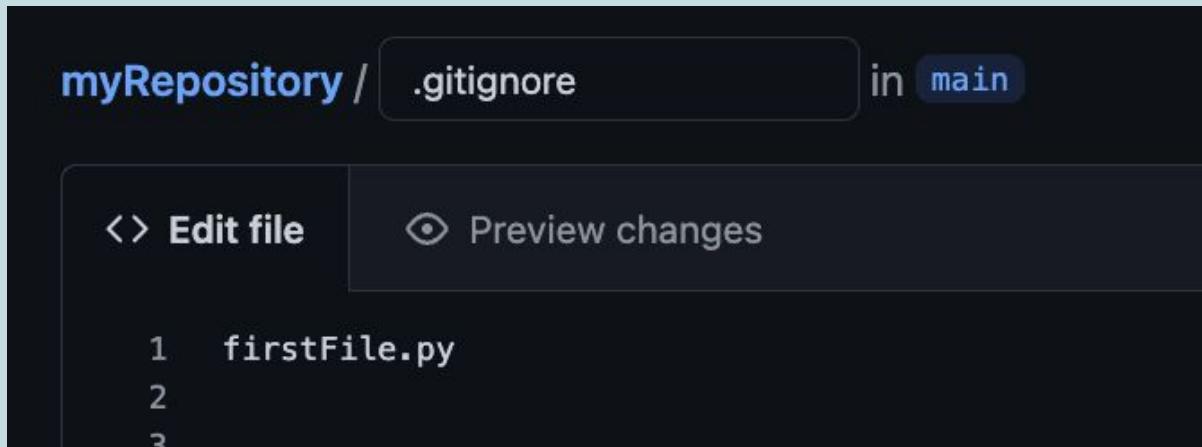
Create an Empty Directory

- Mkdir to create a folder
- Cd to move into it
- Use .gitkeep

```
gitkeep@example:~$ mkdir empty-directory
gitkeep@example:~$ cd empty-directory
gitkeep@example:~$ touch .gitkeep
gitkeep@example:~$ git add .
gitkeep@example:~$ git commit -m "Commit empty fo
gitkeep@example:~$ git push origin
```



.gitignore

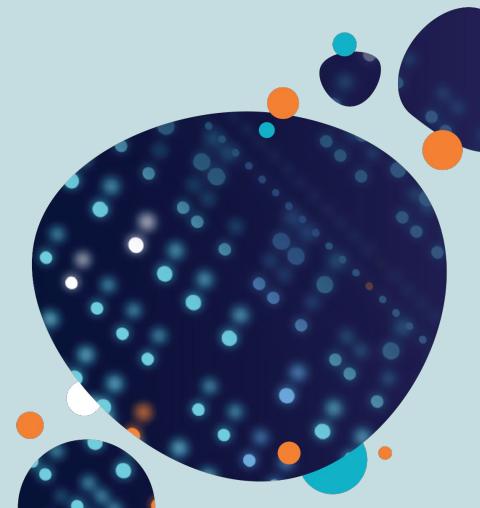


A screenshot of a GitHub repository interface. The top navigation bar shows "myRepository / .gitignore" and "in main". Below the navigation, there are two buttons: "<> Edit file" and "👁 Preview changes". The preview area shows the content of the .gitignore file:

```
1 firstFile.py  
2  
3
```



- Add file to gitignore
 - Ignores file

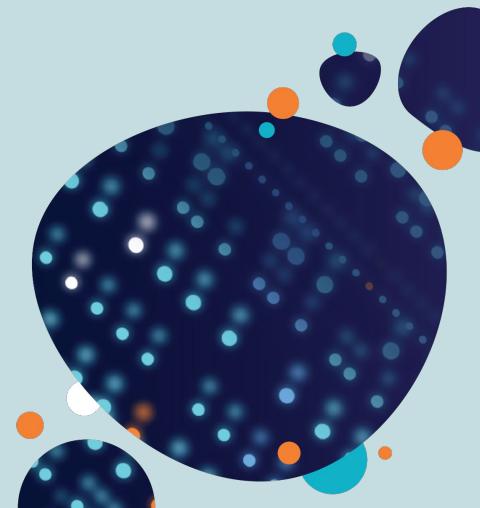


Now it's *your* turn!

1. Create Github Account
2. Download VS Code and Github Desktop
3. Pair up and make a repo
4. Add your teammates as collaborators
5. Start off with:
 - a. **Person A:** Git Add/Commit/Push a file
 - b. **Person B:** Git Fetch/Status/Pull the file
6. Then try out Git Branch/Checkout/Merge individually

If you have time (**Git Fork/Pull Requests**):

1. **Person A:** Make a brand new repo
2. **Person B:** Fork the repo, make changes, make pull request
3. **Person A:** Review pull request, and merge to original repo!



Feedback



data science
student society

