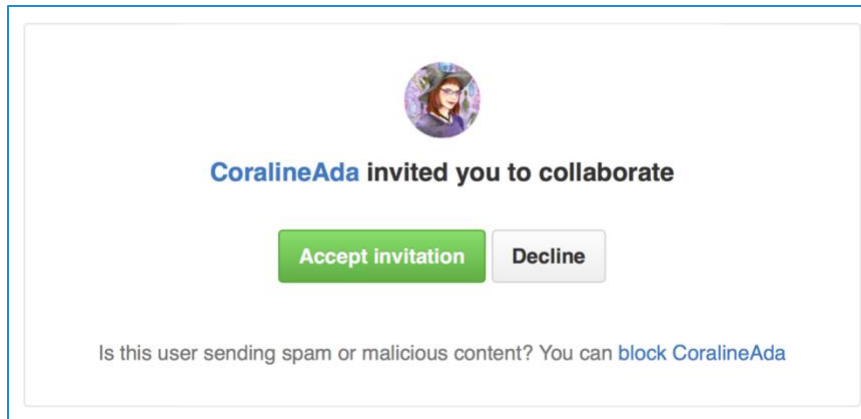


# Applying Best Practices for Sustainable Code: Working on the Workshop Repository

Applying Best Practices for Sustainable Code: Working on the Workshop Repository .....	1
Accept the GitHub Invitation .....	1
Create and Edit Files Directly on GitHub .....	2
Create a file .....	3
Edit an existing file .....	3
Committing Changes .....	3
Run and Edit Code on GitHub Codespaces .....	5
Start a Codespace .....	6
Open and Run Code .....	7
Commit from a Codespace.....	9
Update a Codespace with Changes from the Remote Repository.....	11
You Have No Uncommitted Changes.....	12
You Do Have Uncommitted Changes Outstanding.....	12
I Have a Git Error or Conflict.....	13


## Accept the GitHub Invitation







You should receive an invitation by email to the repository on GitHub that you will share with your team. When you click on the link in the email you will be taken to the repository and see a dialog box like this:





Click the green “Accept invitation” button to access the repository.

The repository should look something like this:

 **Ishahidi** Change line spacing 07907ca · 1 hour ago 🕒 25 Commits

 .devcontainer	Install dependencies with devcontainer.json without rely...	last month
 data	Create test_data.txt	3 days ago
 .gitignore	Initial commit	last month
 README.md	Update README.md	3 days ago
 example_citation.cff	Create example_citation.cff	3 days ago
 text_analysis_notebook.ipynb	Change line spacing	1 hour ago

 **README** 

## Your Project Name

Here is where you should describe your project in detail: the overall objective, the software components, how to install and run the software and any tests or examples.

You should also include information about the project contributors and instructions for how others can contribute, and reference to any resources that are related to the project.

## Create and Edit Files Directly on GitHub

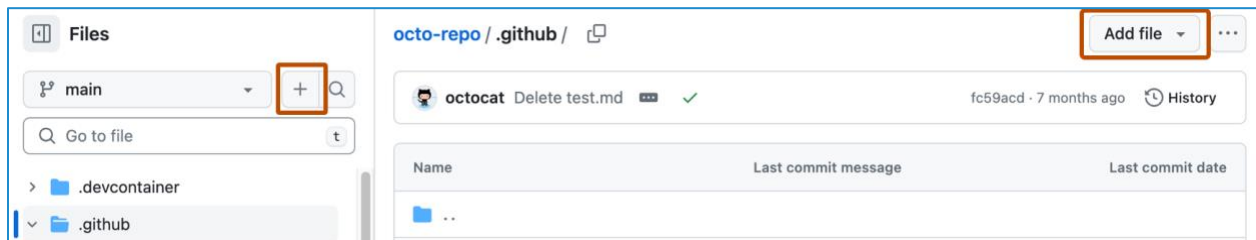
To create new files and edit existing ones in a **simple way** you can use the built-in GitHub editor.

We recommend that you agree with your team which file or files you will be working on to avoid conflicts. Conflicts can be resolved, with some effort, but for this simple workshop it is better to avoid creating them in the first place!

## Create a file

In the repository, browse to the folder where you want to create a file.

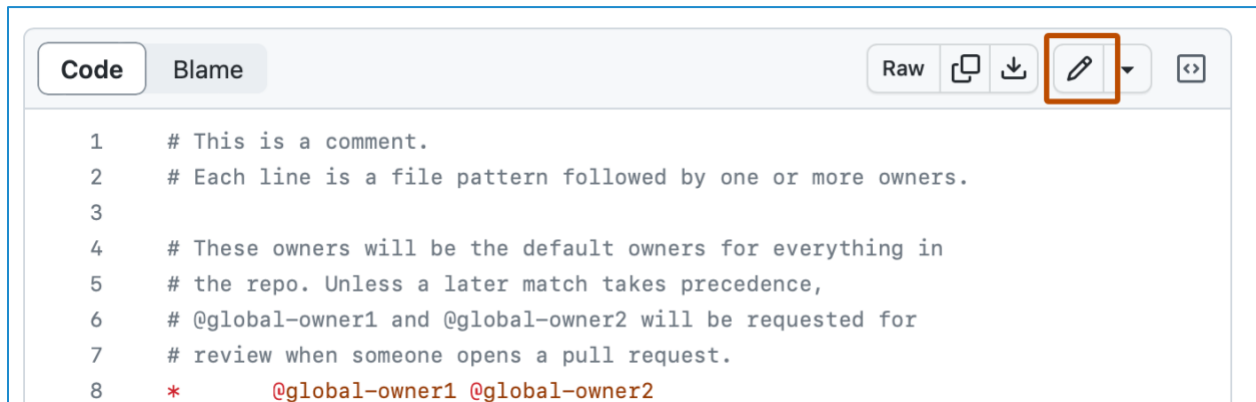
Above the list of files, in the upper right corner, select the **Add file** dropdown menu, then click **Create new file**. Alternatively, you can click the plus sign (+) in the file tree view on the left.



See the GitHub documentation [Creating new files](#) for more guidance.

## Edit an existing file

In your repository, browse to the file you want to edit. In the upper right corner of the file view, click the **pencil icon** to open the file editor.

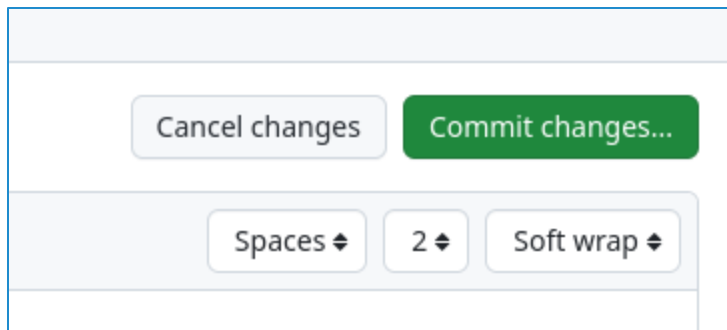


See the GitHub documentation [Editing files in your repository](#) for more guidance.

## Committing Changes

Once you are finished with your edits, you need to commit them to the repository so that your team can see the changes.

In the upper right corner of the file, click the green **Commit changes...** button.



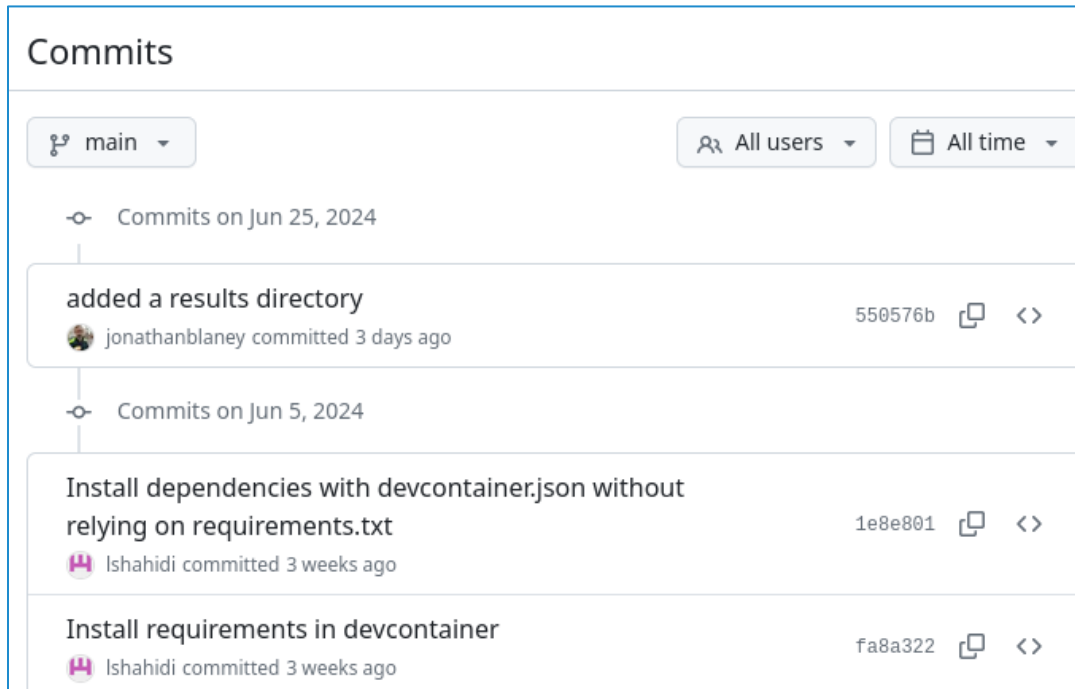
In the dialogue box that appears:

- Add your commit message ([reminder on how to write good commit messages](#))
- Check “Commit directly to the main branch”

Then click the green **Commit changes** button.

A 'Commit changes' dialog box. The title bar says 'Commit changes' with a close button (X) on the right. Inside, there's a section 'Commit message' with a text input field containing 'Your commit message here'. Below that is an 'Extended description' section with a larger text area containing the placeholder text 'Add an optional extended description..'. At the bottom, there are two radio buttons: the first is selected and labeled 'Commit directly to the main branch', and the second is labeled 'Create a **new branch** for this commit and start a pull request' with a link '[Learn more about pull requests](#)' below it. At the very bottom are two buttons: a light gray 'Cancel' button and a green 'Commit changes' button.

You should now see your commit in the commit history of your repository:



Commits

main All users All time

Commits on Jun 25, 2024

added a results directory 550576b

jonathanblaney committed 3 days ago

Commits on Jun 5, 2024

Install dependencies with devcontainer.json without relying on requirements.txt 1e8e801

Ishahidi committed 3 weeks ago

Install requirements in devcontainer fa8a322

Ishahidi committed 3 weeks ago

## Run and Edit Code on GitHub Codespaces


If you wish to **run the code** as part of your work, then you can use GitHub Codespaces. Codespaces allows you to run the code in a virtual environment in the cloud without cloning or installing anything on your local machine.

See the GitHub documentation [GitHub Codespaces](#) for full guidance. The information below is the minimum to get you started.

All accounts on GitHub (whether paid or not) have a free quota of Codespaces run time.

Account plan	Storage per month	Core hours per month
GitHub Free for personal accounts	15 GB-month	120
GitHub Pro	20 GB-month	180

You can view your current usage here: <https://github.com/settings/billing/summary>. Scroll down a bit to see your Codespaces usage for the month.

<div>  <b>Codespaces</b> </div> <div>Included quotas reset in 3 days. <a href="#">See billing documentation</a></div>		
> Usage hours	13.51 of 180.00 included core hours used	\$0.00
> Storage	1.05 of 20.00 included GB-month used	\$0.00
\$0.00 monthly spending limit   <a href="#">Set up a spending limit</a>		\$0.00

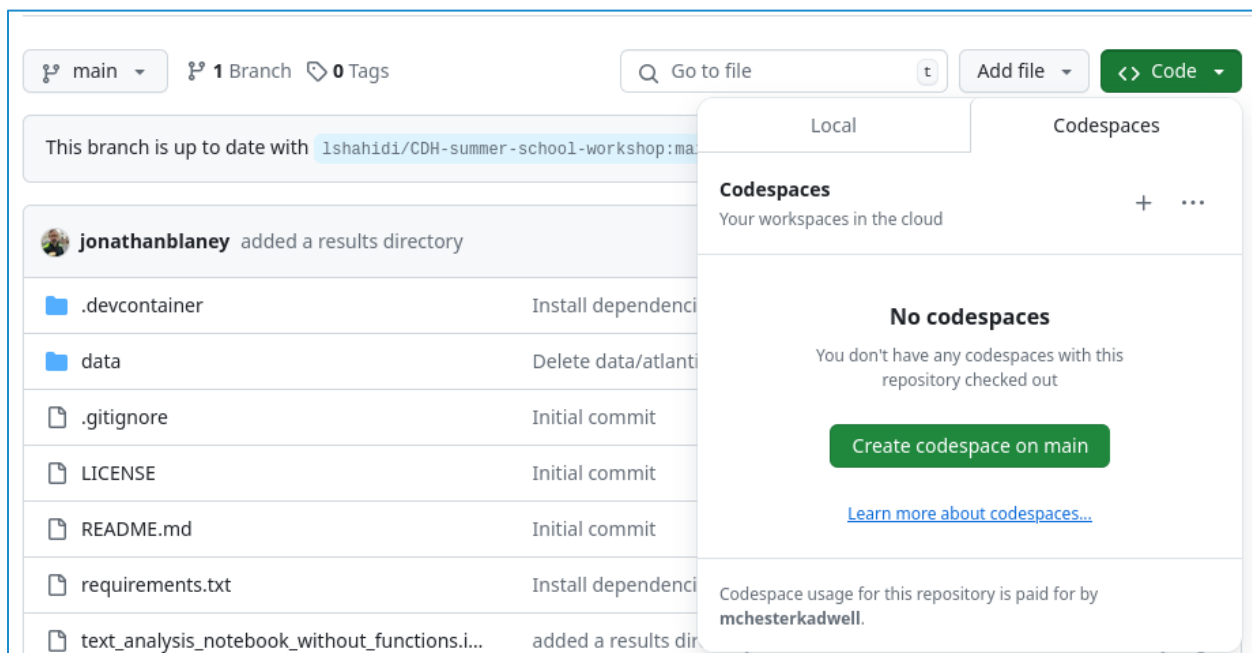
To view all the Codespaces you have and whether they are currently running visit:

<https://github.com/codespaces>

From here you can stop and delete them to ensure you are not using up your quota unnecessarily.

## Start a Codespace

In the top right corner of your repository, click the green **Code** button.



The screenshot shows a GitHub repository interface. At the top, there's a navigation bar with 'main' branch selected, '1 Branch', and '0 Tags'. A search bar 'Go to file' and an 'Add file' button are also present. The 'Code' button is highlighted in green. A dropdown menu is open, showing 'Local' and 'Codespaces' tabs. Under the 'Codespaces' tab, it says 'Your workspaces in the cloud' and 'No codespaces'. A green button 'Create codespace on main' is visible, along with a link 'Learn more about codespaces...'. At the bottom of the dropdown, it states 'Codespace usage for this repository is paid for by mchesterkadwell.'

NB: There is a note that says 'Codespace usage for this repository is paid for by <useraccount>'. If you have a Free account, you will not be charged. If you have a Pro account that you pay for with a registered credit card, you would only be charged for usage if you exceed the free monthly quota.

In the Codespaces tab, click the green button **Create codespace on main**.

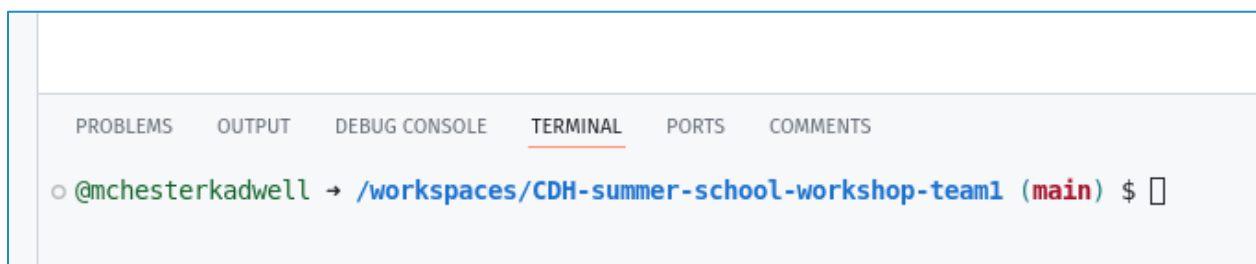
A new browser tab will open and some setting up will happen. The full build will take around 65 seconds. You need to wait until the `postCreateCommand` has finished:



The screenshot shows a VS Code terminal window with the 'TERMINAL' tab selected. The output text is as follows:

```
Use Cmd/Ctrl + Shift + P -> View Creation Log to see full logs
✓ Finishing up...
✓ Running updateContentCommand...
⚙ Running postCreateCommand...
  > python -m spacy download en_core_web_sm
```

Finally, the command prompt appears at the bottom of the window and your working space is ready!

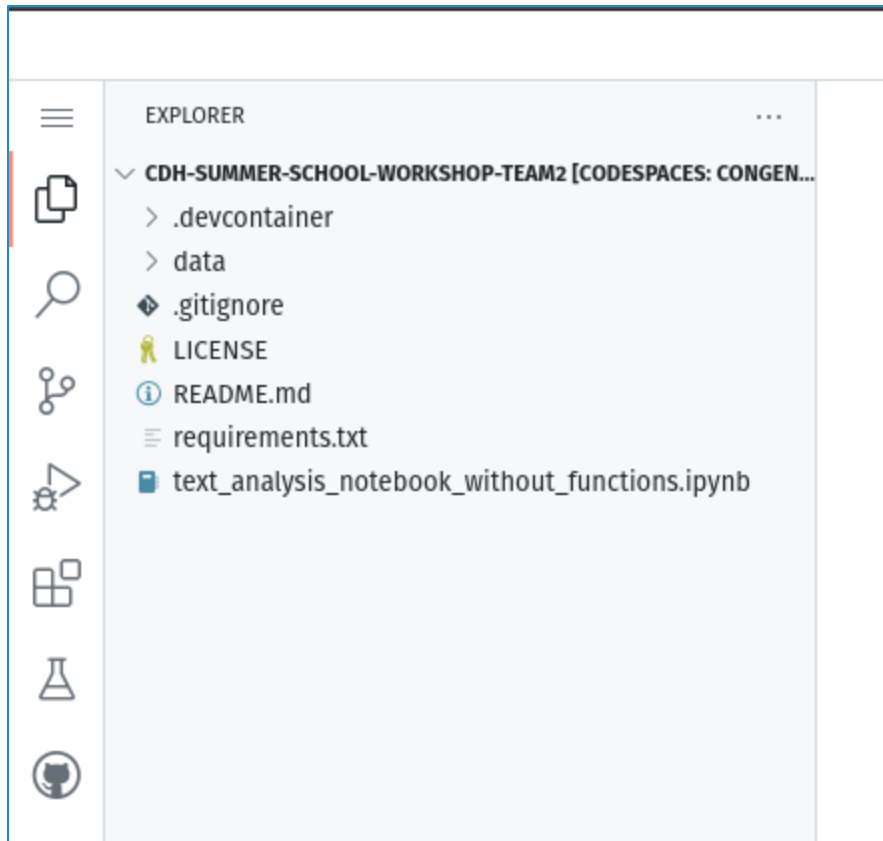


The screenshot shows the same VS Code terminal window, now displaying a command prompt. The prompt text is:

```
@mchesterkadwell → /workspaces/CDH-summer-school-workshop-team1 (main) $ █
```

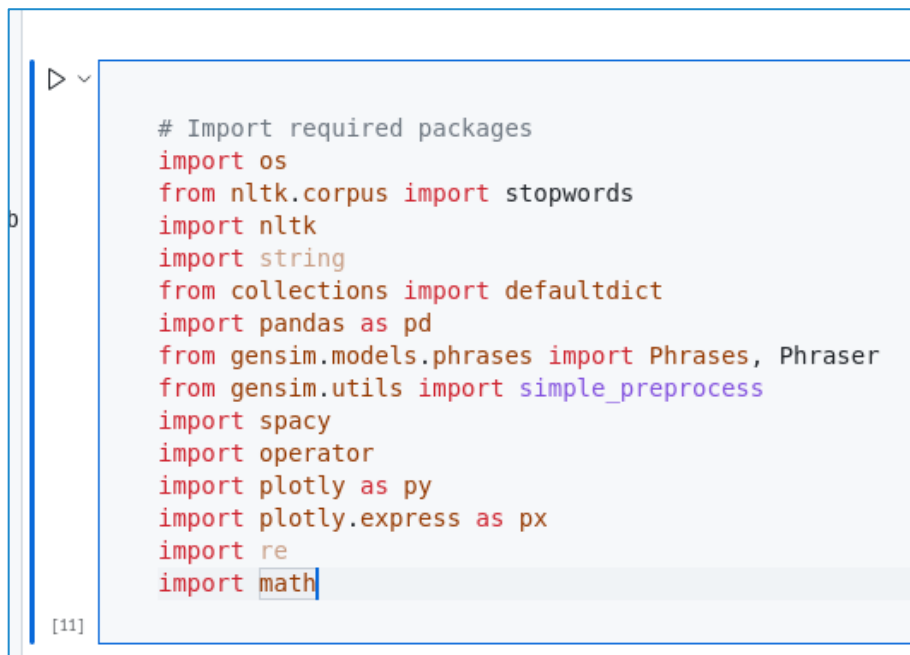
## Open and Run Code

You can see the list of files in the repository on the left-hand side (the actual contents may vary from the example listing below).



To run the Jupyter Notebook, click on the file with the **.ipynb** extension.

The notebook opens and shows a series of code cells. To get started, in the first cell containing all the imports, click the **small arrow button** in the top left of the cell.



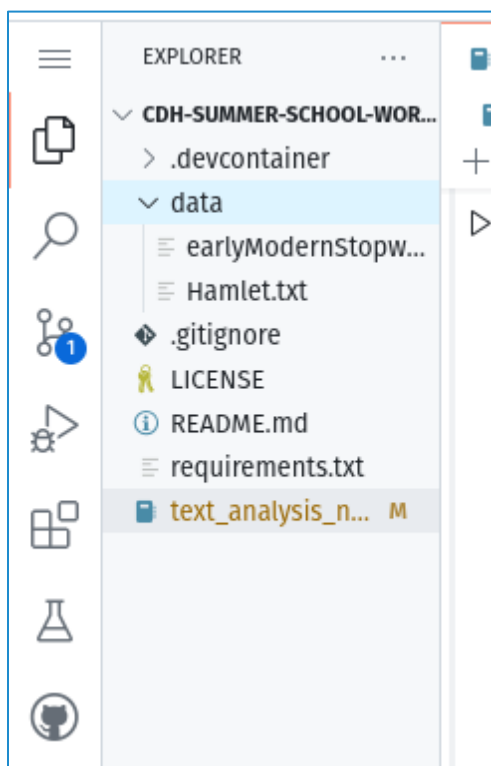


The command pallet pops up at the top of the page with the heading ‘Select kernel’. In the dropdown menu click on **Python Environments...** and then the recommended option of **Python 3.10.13**.

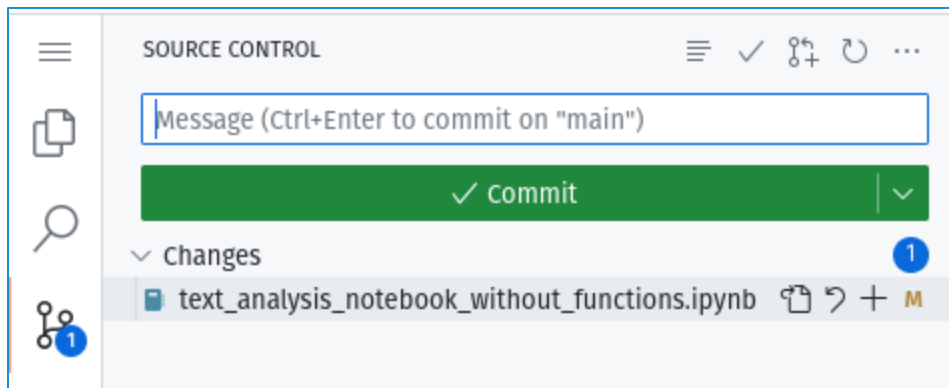
After this your code cell will run and import all the packages. After a cell has finished running, the time it took is displayed in seconds in the bottom left-hand corner of the cell. You can continue down the notebook to run all the cells.

## Commit from a Codespace

Once you have made any change, a **(1)** appears in the Source Control section in the left-hand side of the window. Every file you change will increase this number by one.

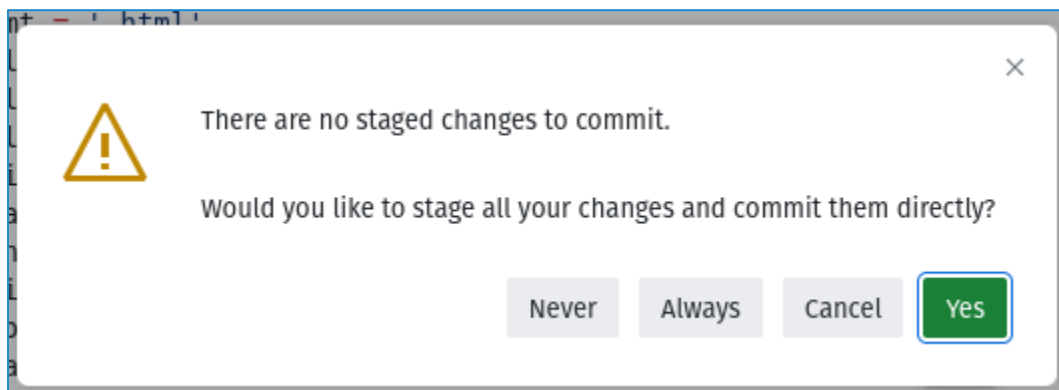


Click on the **branch icon with the number** to go to Source Control and you are presented with a list of changes and a green Commit button.



Enter your commit message in the **Message** text box, and click the **Commit** button.

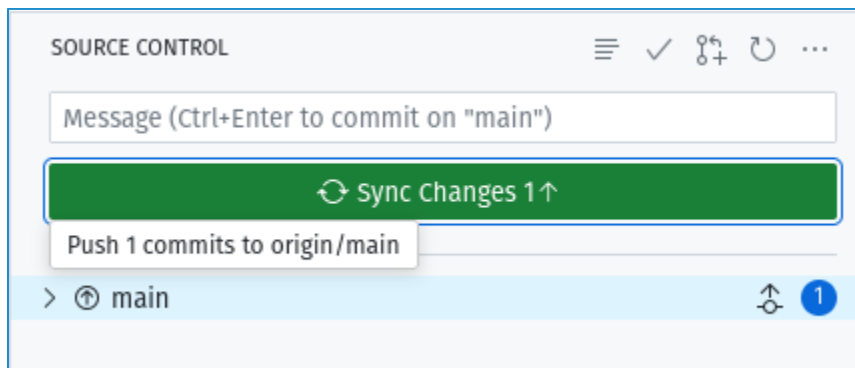
The first time you do this you will get a popup about 'no staged changes to commit'.



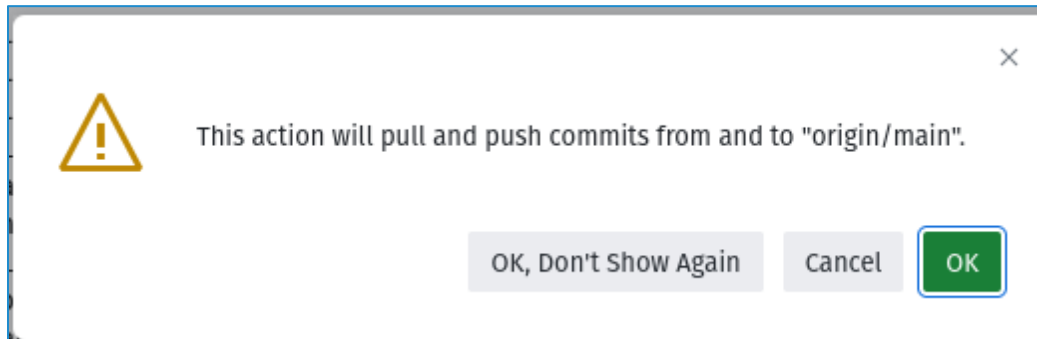
Unless there are changes you do not wish to commit, it is safe to click the **Yes** button and have all your changes both staged and committed at the same time.

Now you have made a commit, but as yet your changes have not been pushed to the remote repository.

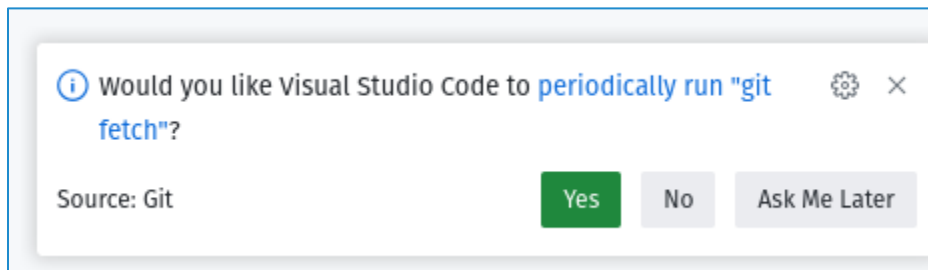
The green Commit button has changed into a Sync Changes button:



Click the **Sync Changes** button. This creates a popup about ‘pull and push commits from and to “origin/main”’. Click **OK**.



Finally, a popup may appear in the bottom left corner of the window about ‘git fetch’. Click **Yes** to get any changes from your team members automatically fetched from the repository.

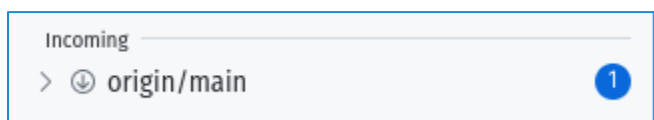


## Update a Codespace with Changes from the Remote Repository

While you are busy working away, your team may also be busy adding changes to your shared repository. As long as you have set your Codespace to periodically run ‘git fetch’ for you (as above), your team’s changes will be made available to you in your Codespace automatically.

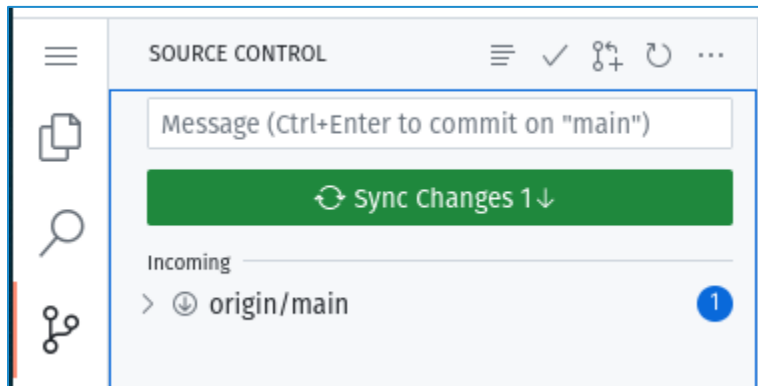
However, to access their commits and incorporate them into your copy of the repository, you need to take action.

When changes have been made in the remote repository, the incoming changes will be shown with a number under the heading ‘Incoming’. (1) indicates one incoming change; (2) indicates two changes, and so on.

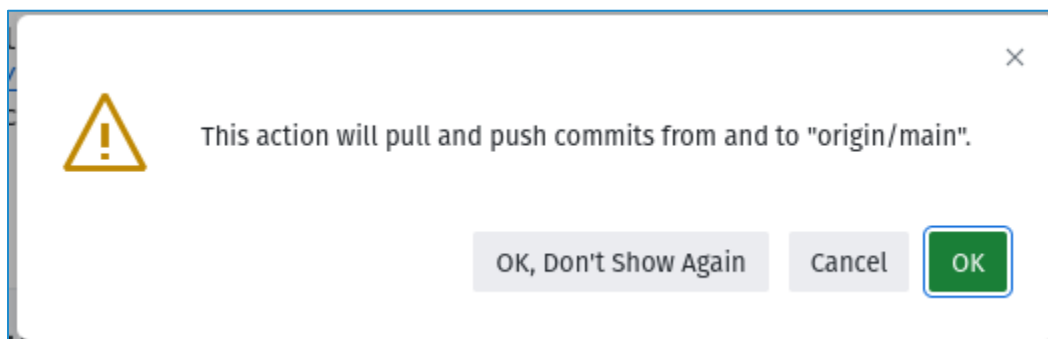


### ***You Have No Uncommitted Changes***

If you have no uncommitted changes yourself, the green button will show **Sync Changes** with a down arrow and a small number that indicates the number of incoming changes.

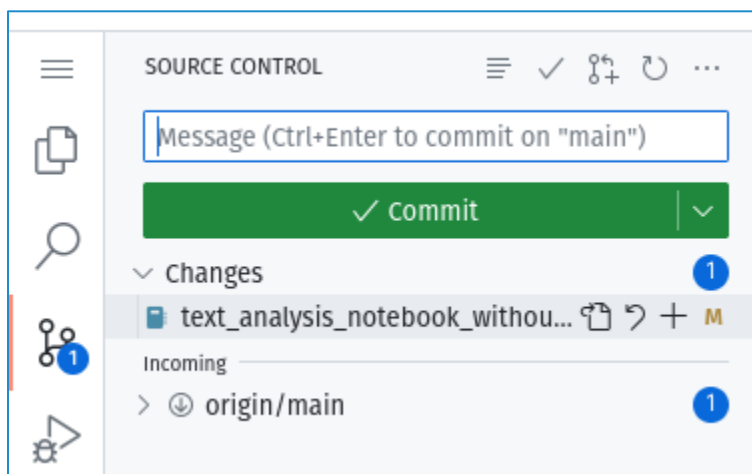


In which case, click on **Sync Changes** and then **OK** in the popup window to pull all the changes.



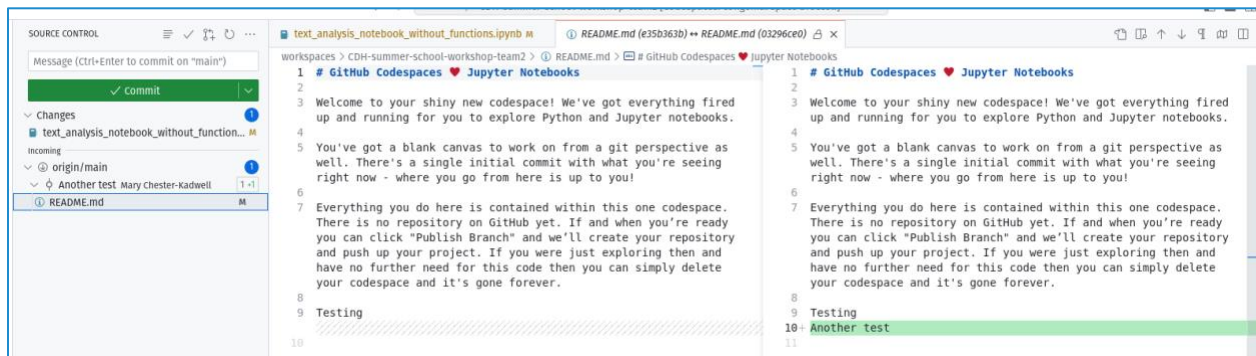
### ***You Do Have Uncommitted Changes Outstanding***

If you do have uncommitted changes, then instead the source control section will show your changes under 'Changes' and the incoming changes under 'Incoming'.



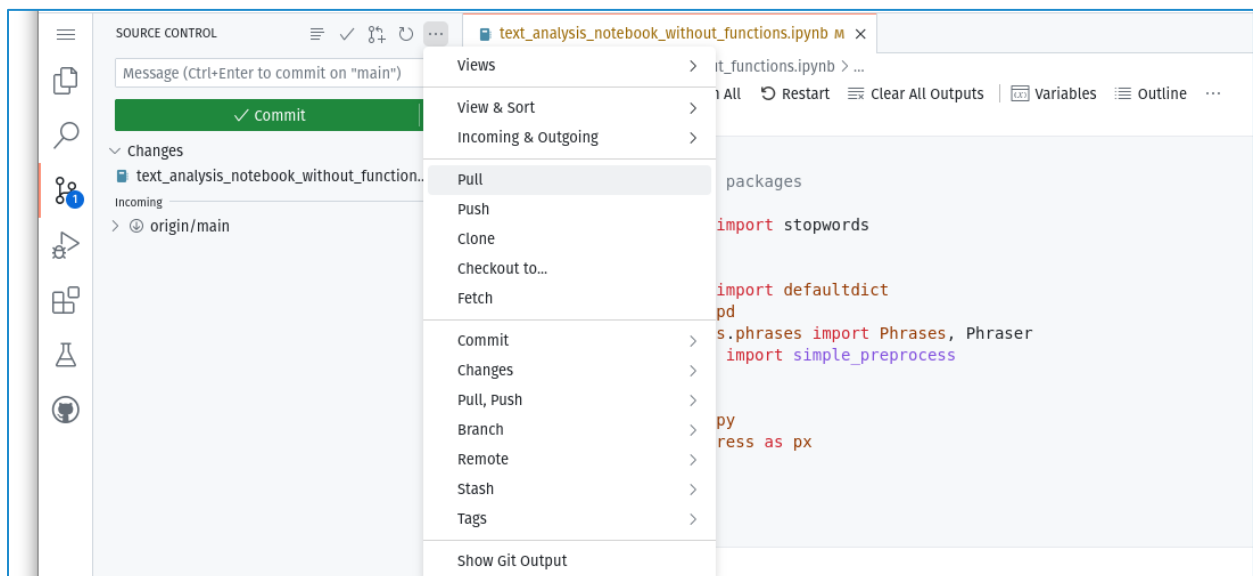
At this point it is wise to double-check that the incoming changes do not conflict with your changes.

Click on the incoming changes list to open a side-by-side comparison between your copy and the incoming version:



If you are satisfied there is no problem, then close the comparison view.

In the top right corner of the Source Control side panel, click on the **3 small dots**, and in the dropdown menu choose **Pull**.



Now all the changes from the remote repository will be incorporated into your copy.

### ***I Have a Git Error or Conflict***

If you find you have an error, however, and you do not know what to do – contact a helper for assistance.

