

Lecture 1: Build your working environment

COSC 526: Introduction to Data Mining



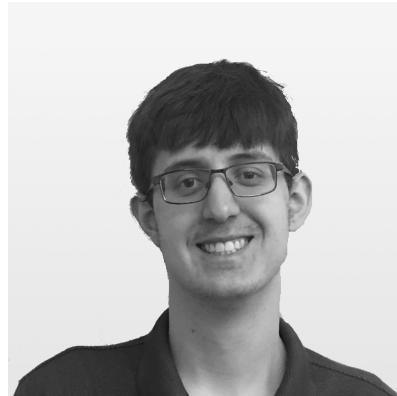
THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

Instructors:

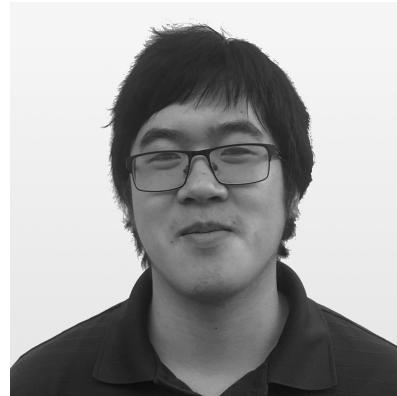


Michela Taufer

Assistants to the Instructor:



Ian Lumsden



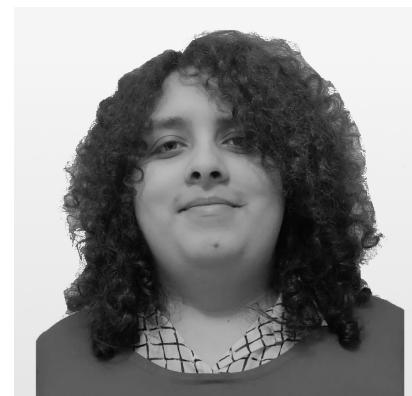
Nigel Tan



Paula Olaya



Leo Valera



Kae Suarez

Build your working environment

- We use **git & GitHub** to distribute & collect assignments as well as other class materials (e.g., slides, code, and datasets)
 - Create a GitHub username (if you have not one)
 - Install git and github desktop on your laptop
- We use **Jupyter notebook** for our assignments and project
 - Follow the steps in the file *Lecture01_student_notes* to install Jupyter
- We use **XSEDE Jetstream** as our cloud platform for assignments and the semester project
 - Introduce its use in the 2nd lecture
 - Create accounts and provide you with access to the cloud in the 4th lecture



GitHub and Git

- **GitHub:** web-based hosting service for **version control**
 - Use it to distribute and collect assignments, to share class materials (e.g., slides, codes, and datasets)
 - Provide us with your GitHub username
- **Git:** software used by GitHub
 - Install git on your laptop
- **Class GitHub Repository:**
 - Clone the course repository
 - <https://github.com/CISC879-BigData/courses-UTK-COSC526-S21>



Git and GitHub



Version Control System

- **Version control system**

“**Version control** is a **system** that records changes to a file or set of files over time so that you can recall specific **versions** later.”

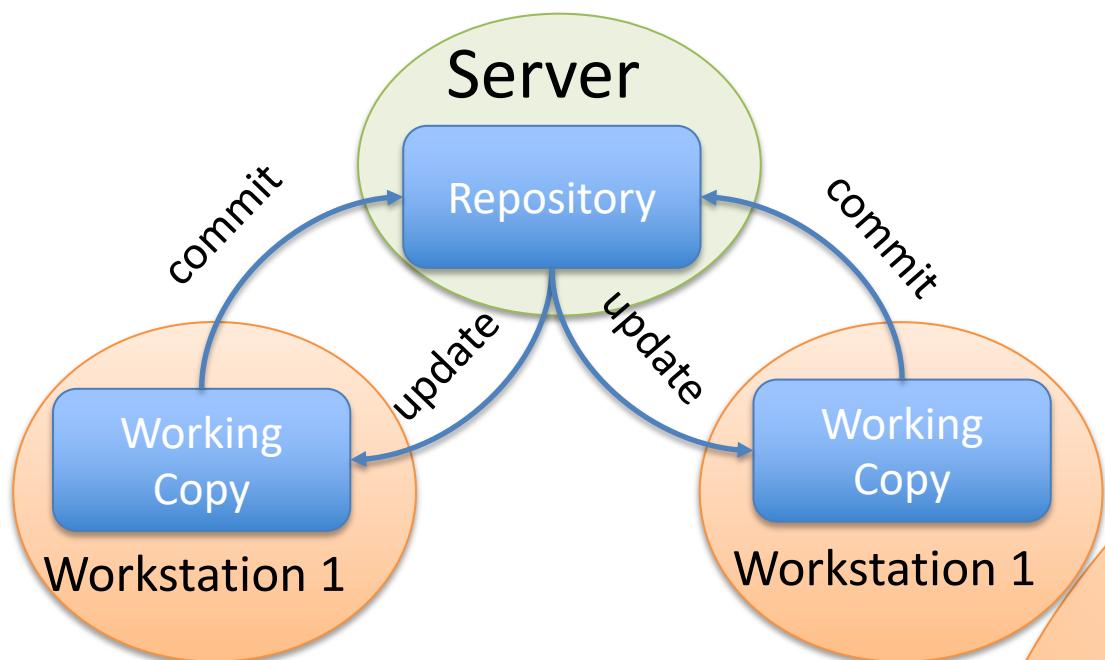
Source: About Version Control – Git; git-scm.com › Getting-Started-About-Version-Control

- Version control systems can be:
 - Centralized
 - Distributed

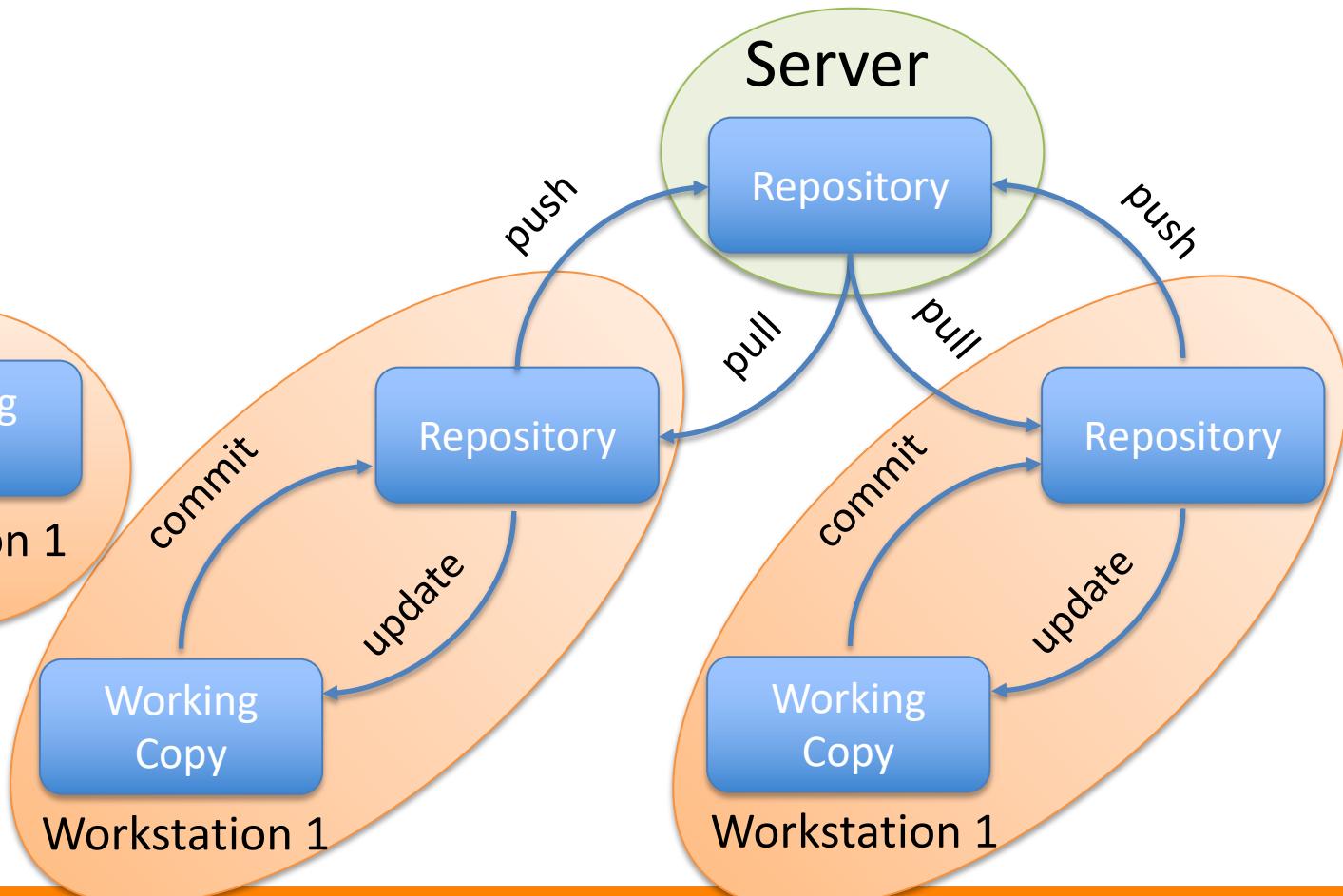


Centralized and Distributed Version Control System

Centralized



Distributed



Why we use Version Control System

- Multiple people can work simultaneously on a single project
 - Everyone works on and edits their own copy of the files and it is up to them when they wish to share the changes made by them with the rest of the team.
- Work done simultaneously by different members of a team can be integrated
 - When conflicting edits are made by two people to the same line of a file, then human assistance is requested by the version control system in deciding what should be done.
- One person can use multiple computers to work on a project
- Users have access to the historical versions of a project
 - This is insurance against computer crashes or data loss
 - If any mistake is made, one can easily roll back to a previous version. It is also possible to undo specific edits that too without losing the work done in the meanwhile
 - It can be easily known when, why, and by whom any part of a file was edited



Git and GitHub

- Git: a free, open source distributed **version control system**
- GitHub: cloud hosted git repositories

Material built from:

<http://rogerdudler.github.io/git-guide/>

<https://marklodato.github.io/visual-git-guide/index-en.html>

https://kbroman.org/github_tutorial/pages/init.html

https://kbroman.org/github_tutorial/pages/routine.html



Install git on your laptop (if you have not yet)

- Windows or Mac
 - Download the GitHub Desktop application:
<https://desktop.github.com>
 - Find GitHub's installation help here:
<https://help.github.com/desktop/guides/getting-started-with-github-desktop/>
- Linux or Mac
 - Install the command-line interface (CLI) of git
 - Find help here:
<https://www.atlassian.com/git/tutorials/install-git>
- Log into github





Overview Release Notes Help

GitHub Desktop

Focus on what matters instead of fighting with Git. Whether you're new to Git or a seasoned user, GitHub Desktop simplifies your development workflow.

[Download for macOS](#)

[Download for Windows](#)

By downloading, you agree to the [Open Source Applications Terms](#).





Welcome to GitHub Desktop

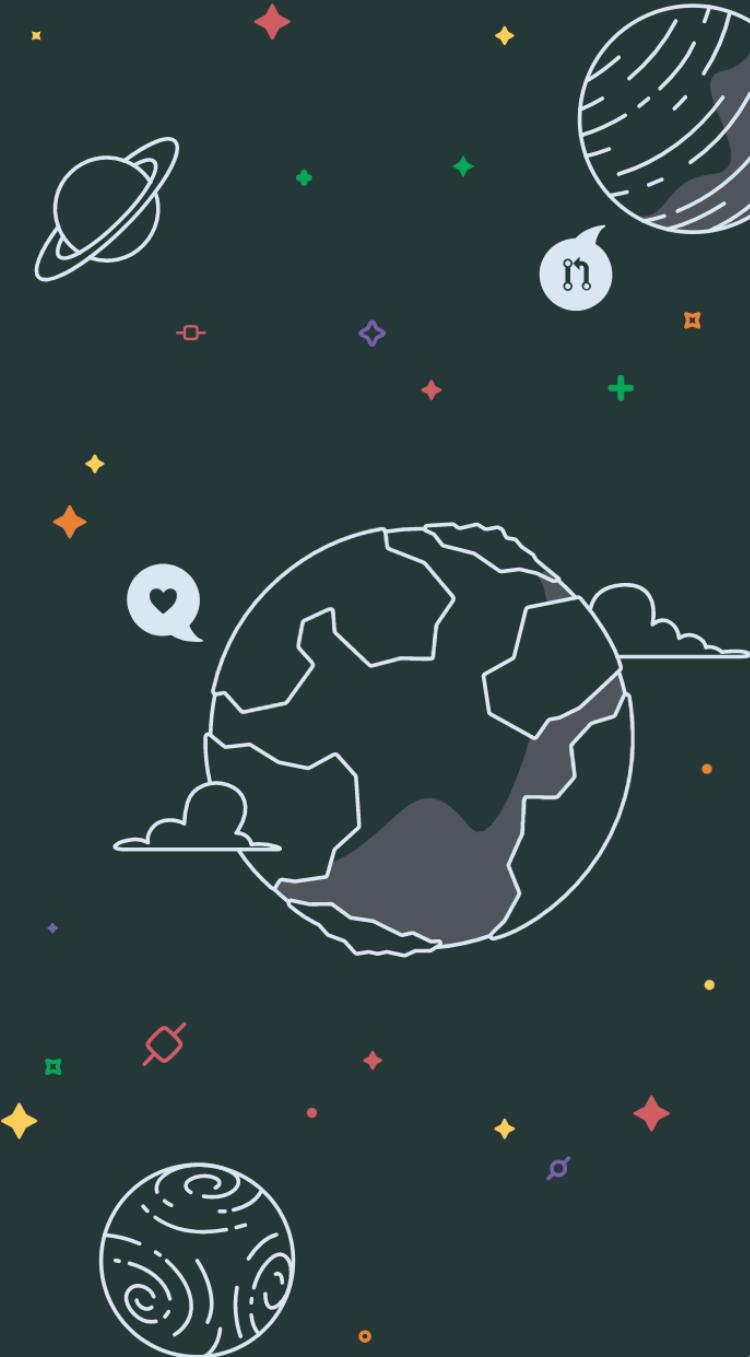
GitHub Desktop is a seamless way to contribute to projects on GitHub and GitHub Enterprise Server. Sign in below to get started with your existing projects.

New to GitHub? [Create your free account.](#)

[Sign in to GitHub.com](#)

[Sign in to GitHub Enterprise Server](#)

[Skip this step](#)



Connect new repos to github

- Go to [github](#)
- Log in to your account

You don't have a github account?



Connect new repos to github

- Go to [github](#)
- Log in to your account

You don't have a github account?

- Get a [github](#) account
 - Go to the GitHub Sign Up page <https://github.com/join>
 - Create a free account



Connect new repos to github

- Go to [github](#)
- Log in to your account

You don't have a github account?

- Get a [github](#) account
 - Go to the GitHub Sign Up page <https://github.com/join>
 - Create a free account
- Share your github account with us:
 - Complete form <https://forms.gle/BBdtpSQeYaeut3ru9>



Use git and GitHub

- The routine use of git involves just a few commands:
 - initiate a repository
 - add and commit
 - push and pull
 - status
 - diff
- You can deal with git and github via:
 - GitHub Desktop (GUI)
 - Command Line (CLI)



GitHub Desktop Application





Welcome to GitHub Desktop

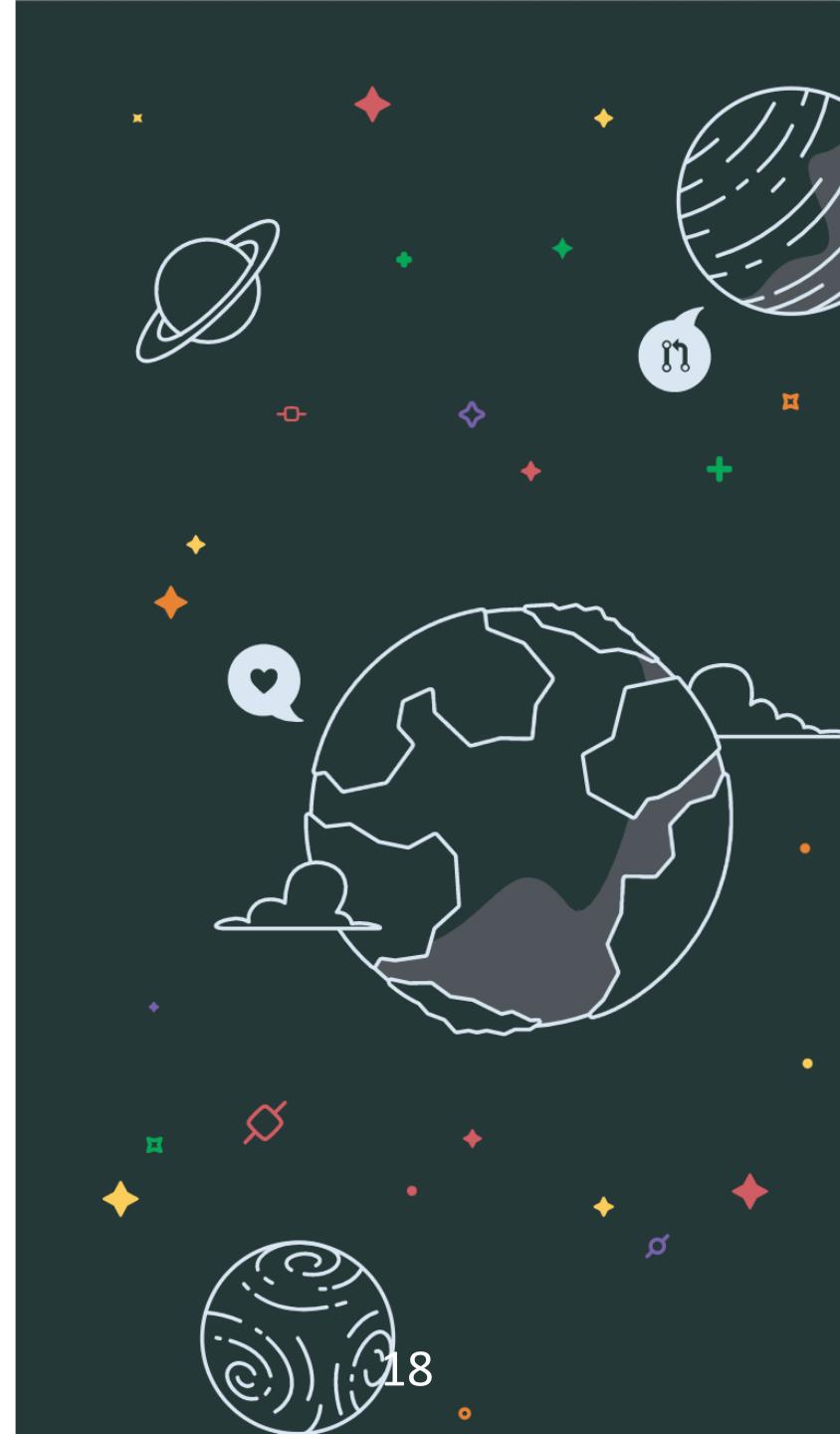
GitHub Desktop is a seamless way to contribute to projects on GitHub and GitHub Enterprise Server. Sign in below to get started with your existing projects.

New to GitHub? [Create your free account.](#)

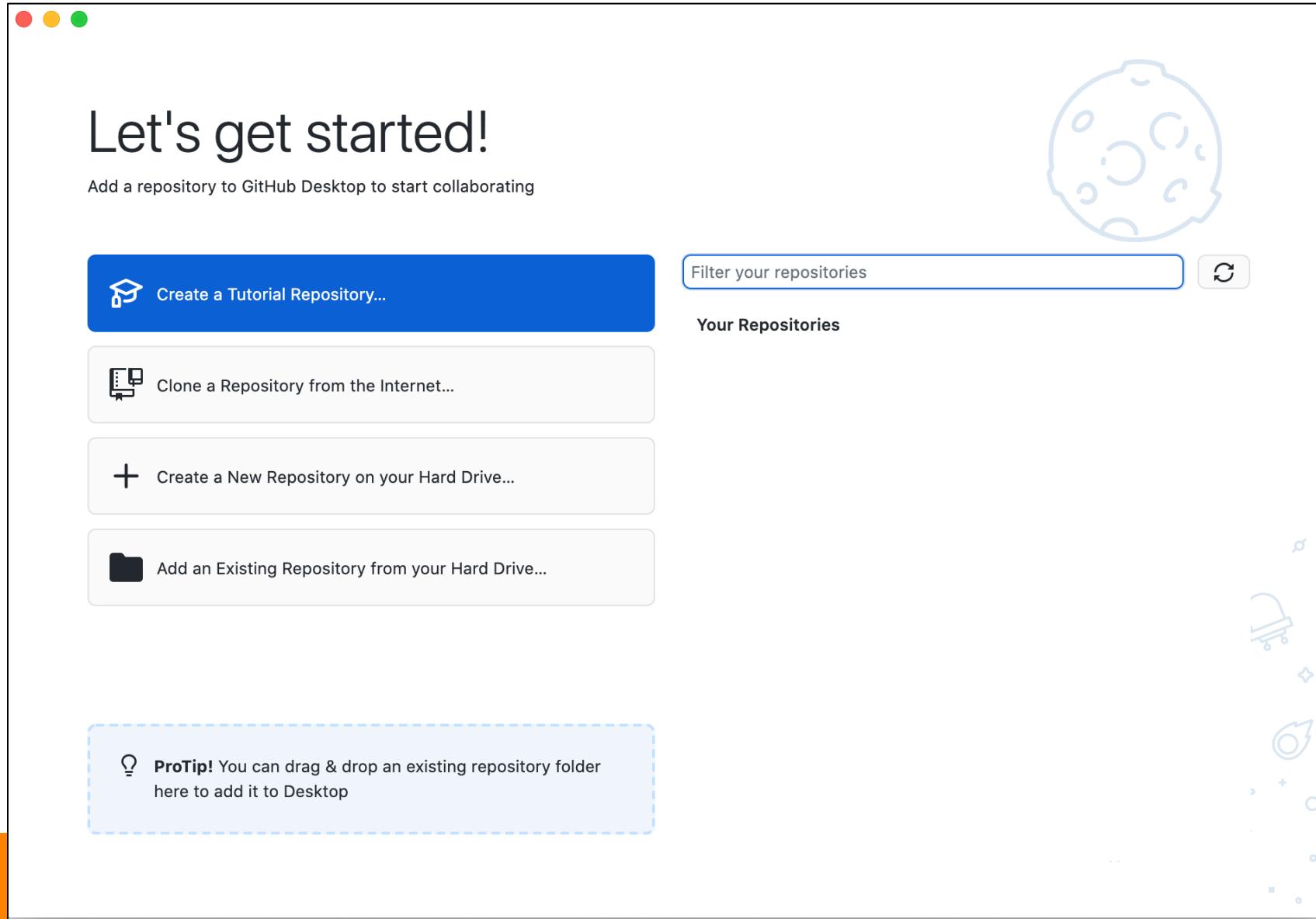
[Sign in to GitHub.com](#) 

[Sign in to GitHub Enterprise Server](#)

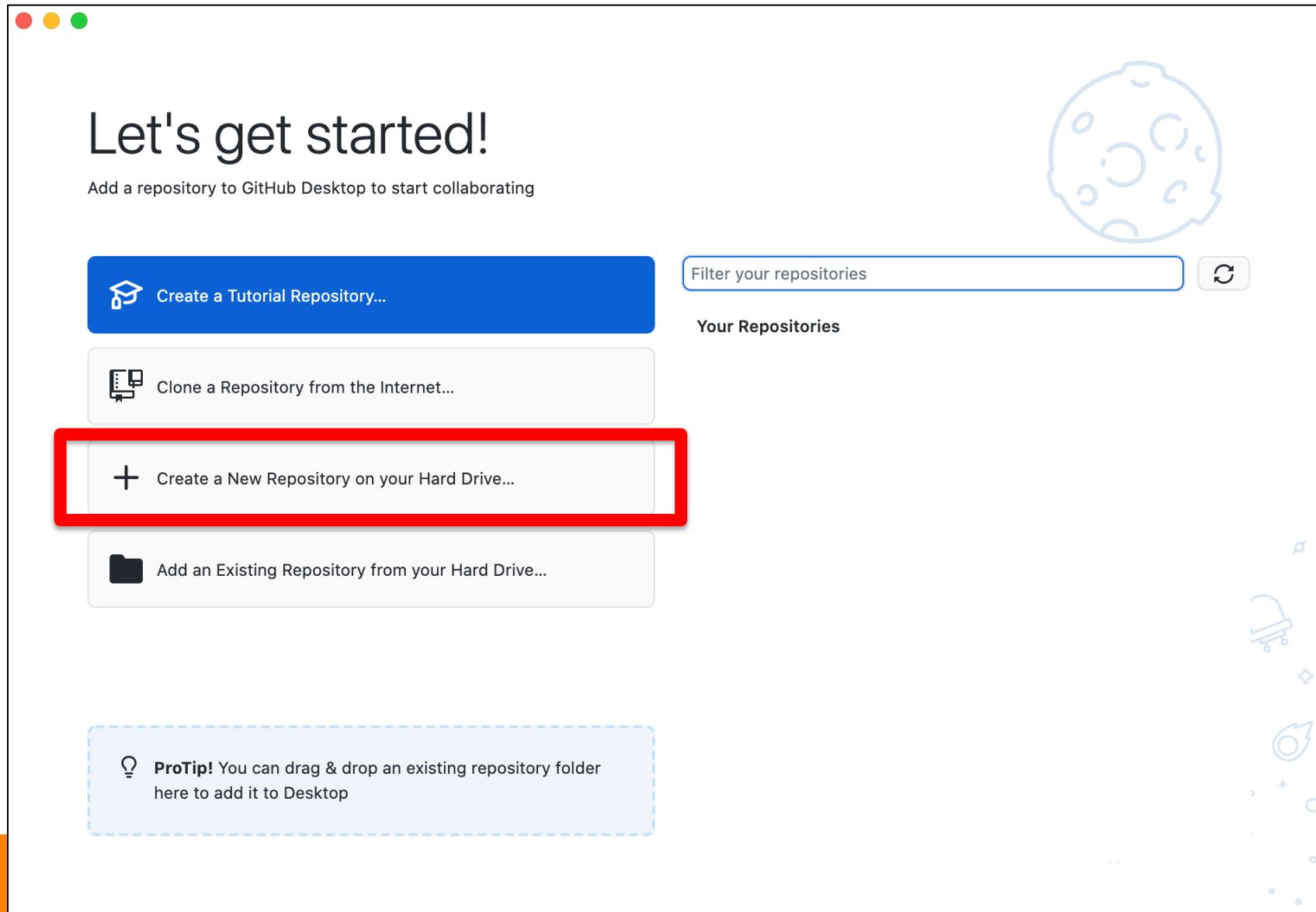
[Skip this step](#)



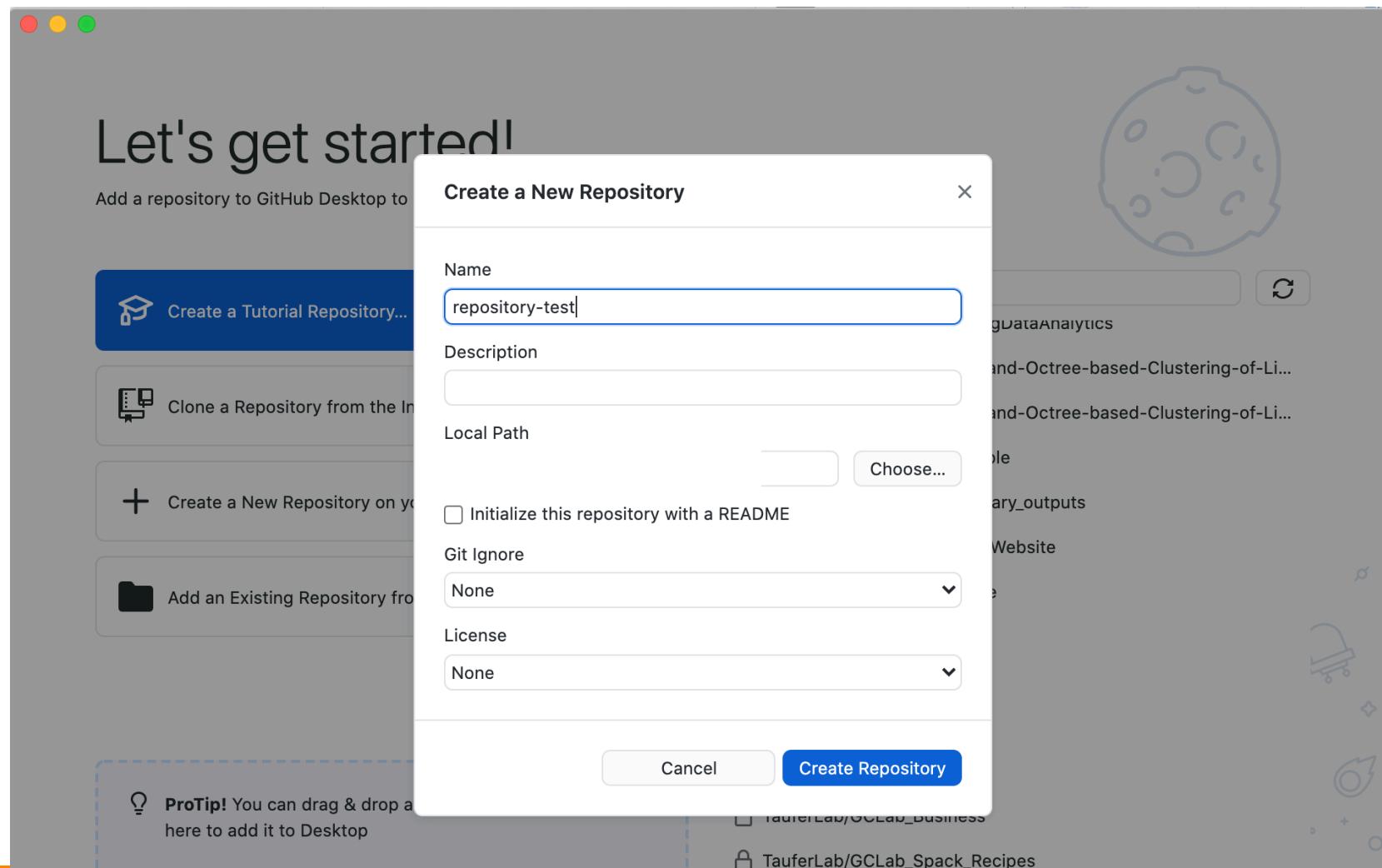
GitHub Desktop Application: Let's start



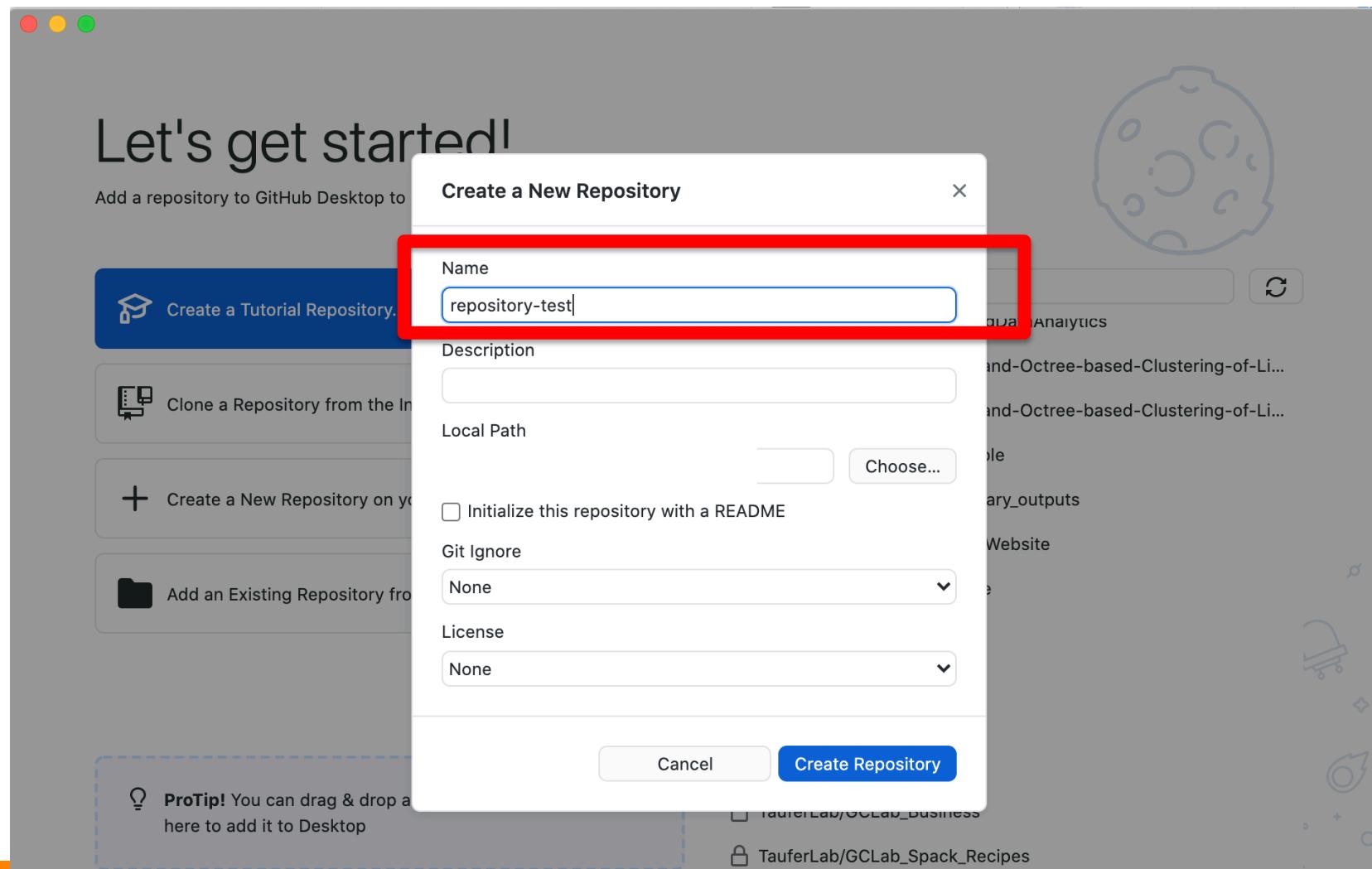
GitHub Desktop Application: Let's start



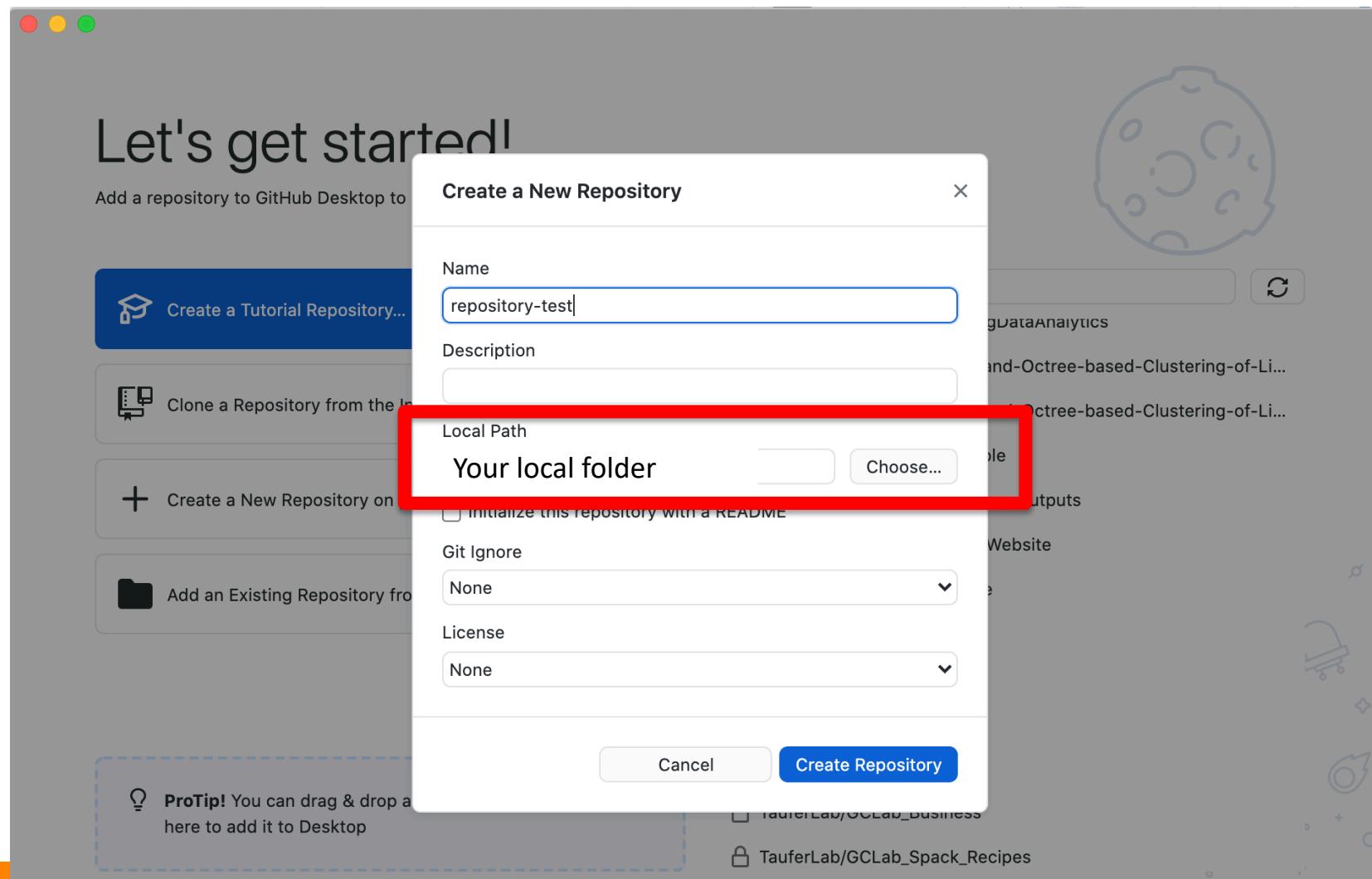
Create a Repository (I)



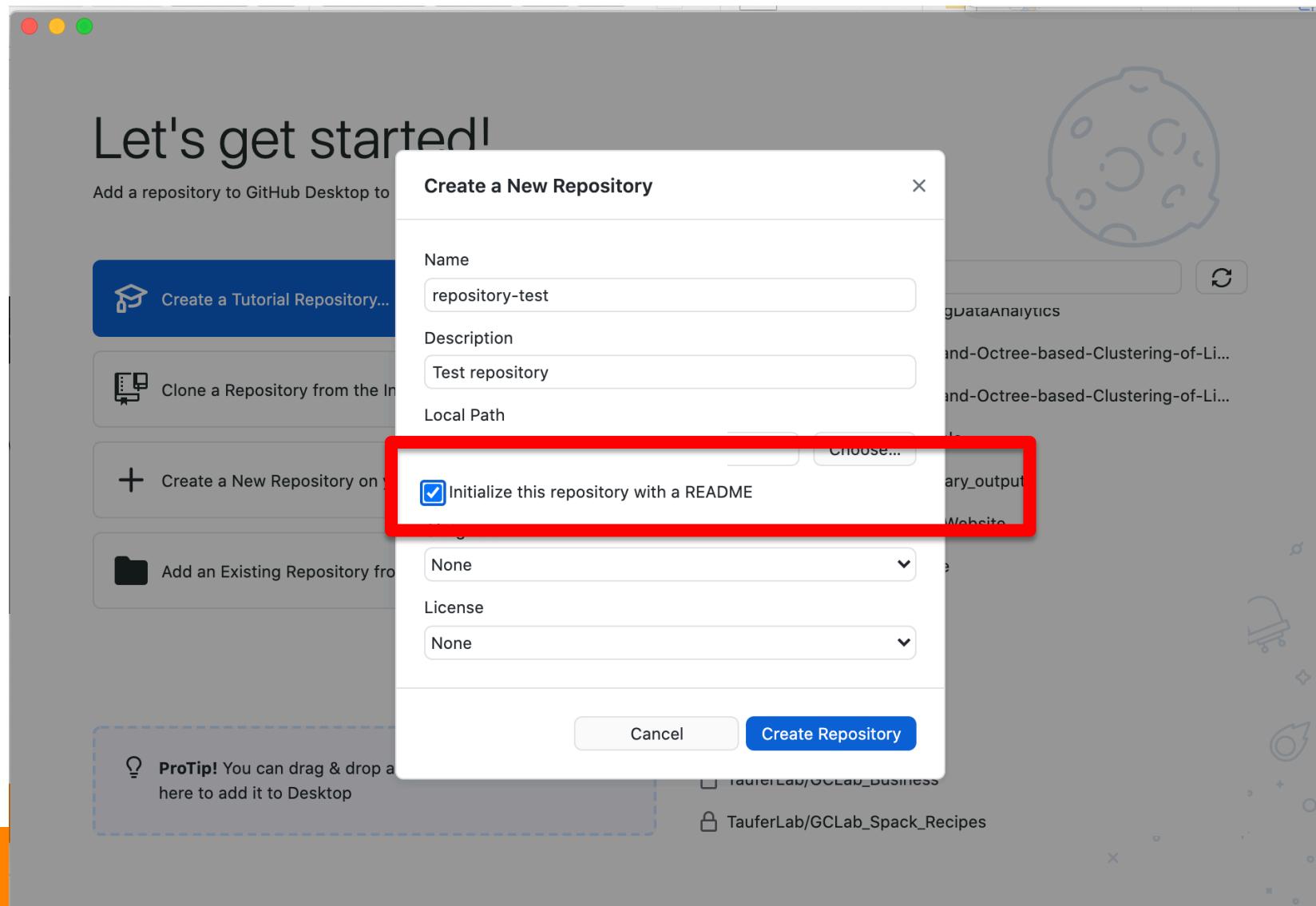
Create a Repository (II)



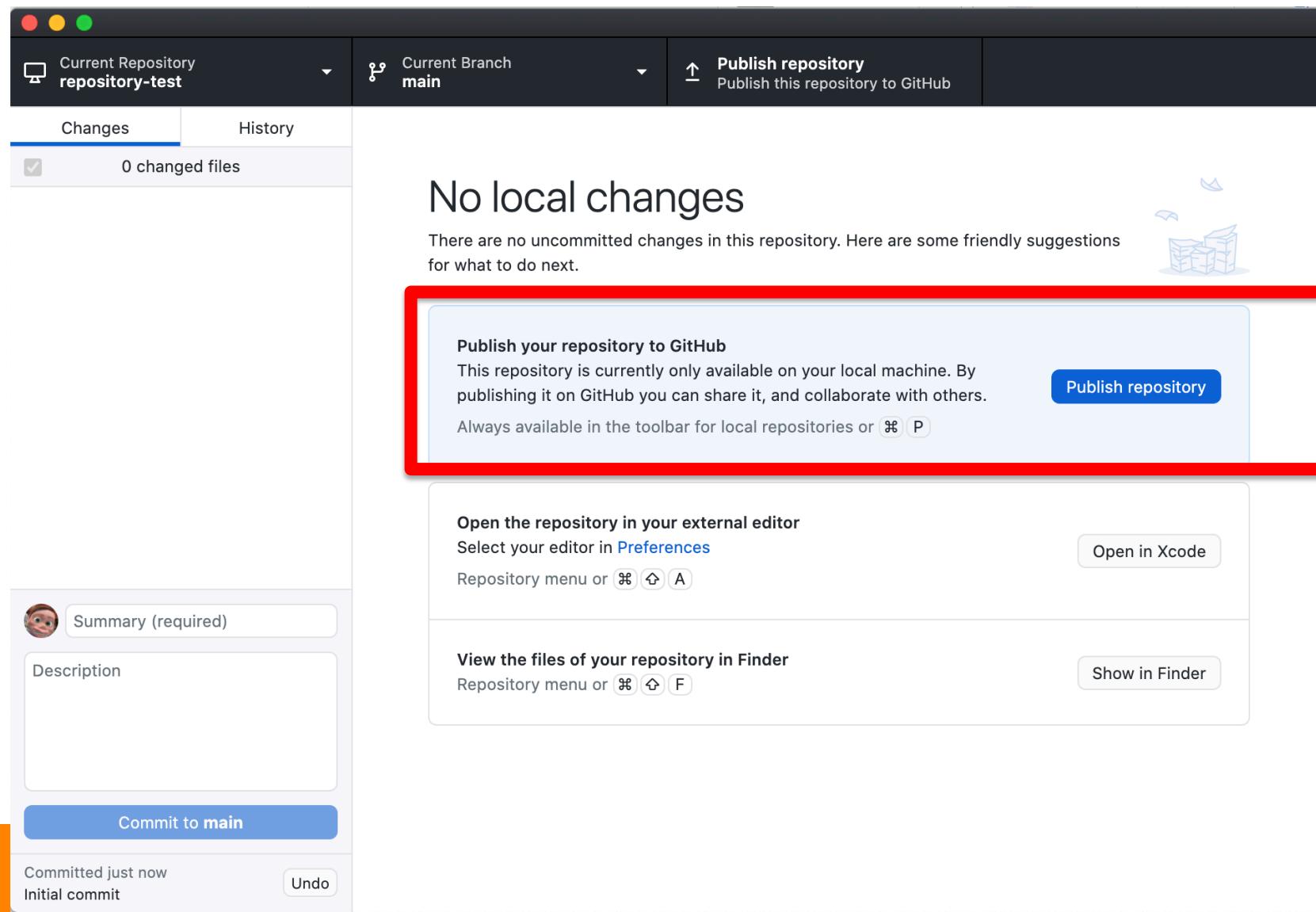
Create a Repository (III)



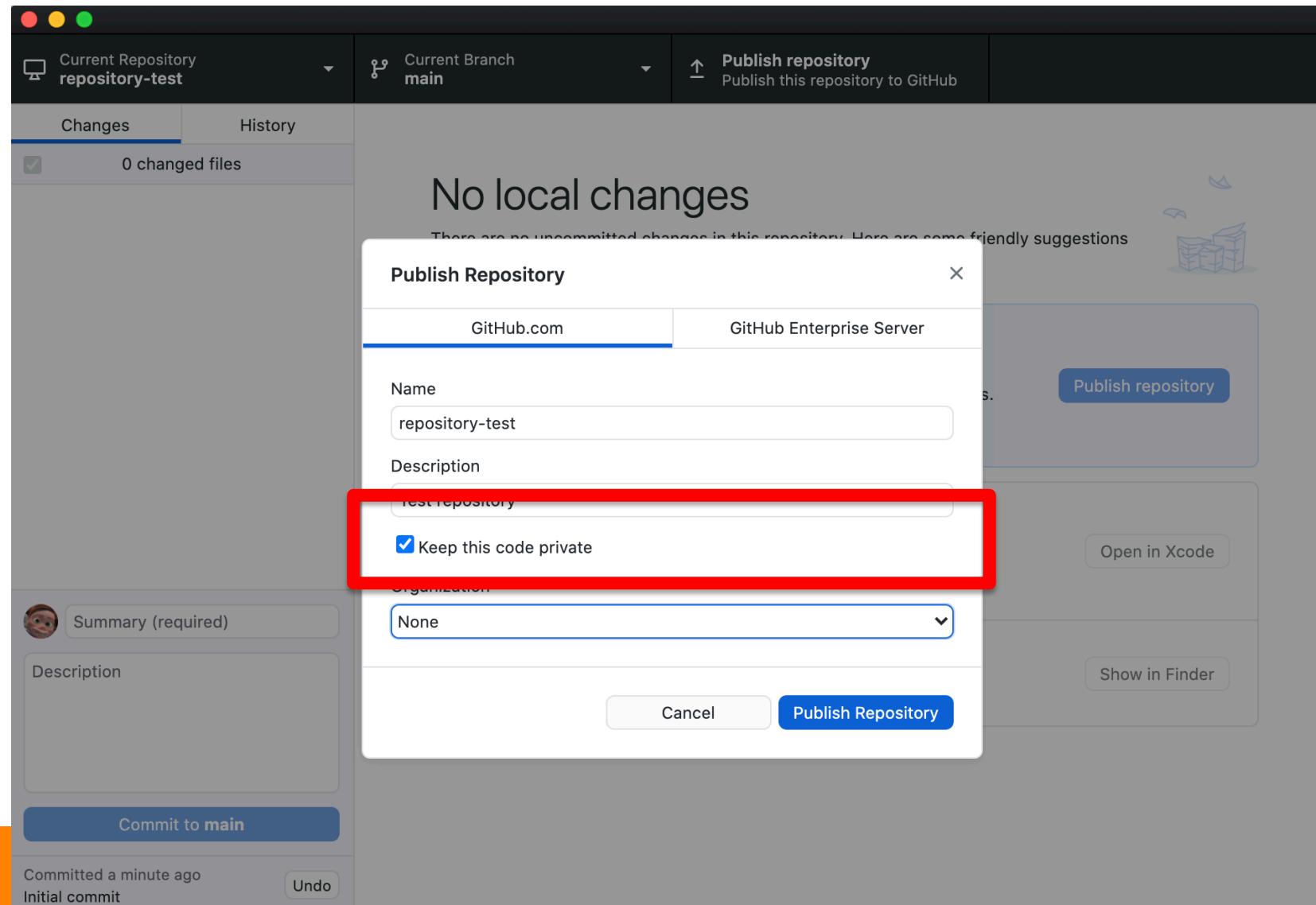
Create a Repository (IV)



Publish your new Repository (I)



Publish your new Repository (II)



Publish your new Repository (III)

The screenshot shows the Xcode interface with a repository named "repository-test" and a branch "main". The "Changes" tab is selected, showing "0 changed files". A central message says "No local changes" with a suggestion to "Open the repository page on GitHub in your browser". This option is highlighted with a red box.

Current Repository: repository-test

Current Branch: main

Last fetched 7 minutes ago

Changes History

0 changed files

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Open the repository in your external editor
Select your editor in Preferences
Repository menu or ⌘ ⌘ A

Open in Xcode

View the files of your repository in Finder
Repository menu or ⌘ ⌘ F

Show in Finder

Open the repository page on GitHub in your browser
Repository menu or ⌘ ⌘ G

View on GitHub

Summary (required)

Description

Commit to main



Publish your new Repository (IV)

🔒 [taufer / repository-test](#) Private

[Unwatch](#) 1 [Star](#) 0 [Fork](#) 0

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#) [Settings](#)

[main](#) [1 branch](#) [0 tags](#)

[Go to file](#) [Add file](#) [Code](#)

 taufer Initial commit	2ef51ac 3 minutes ago	⌚ 1 commit
.gitattributes	Initial commit	3 minutes ago
README.md	Initial commit	3 minutes ago

[README.md](#)

repository-test

Test repository

About 

Test repository

[Readme](#)

Releases

No releases published

[Create a new release](#)

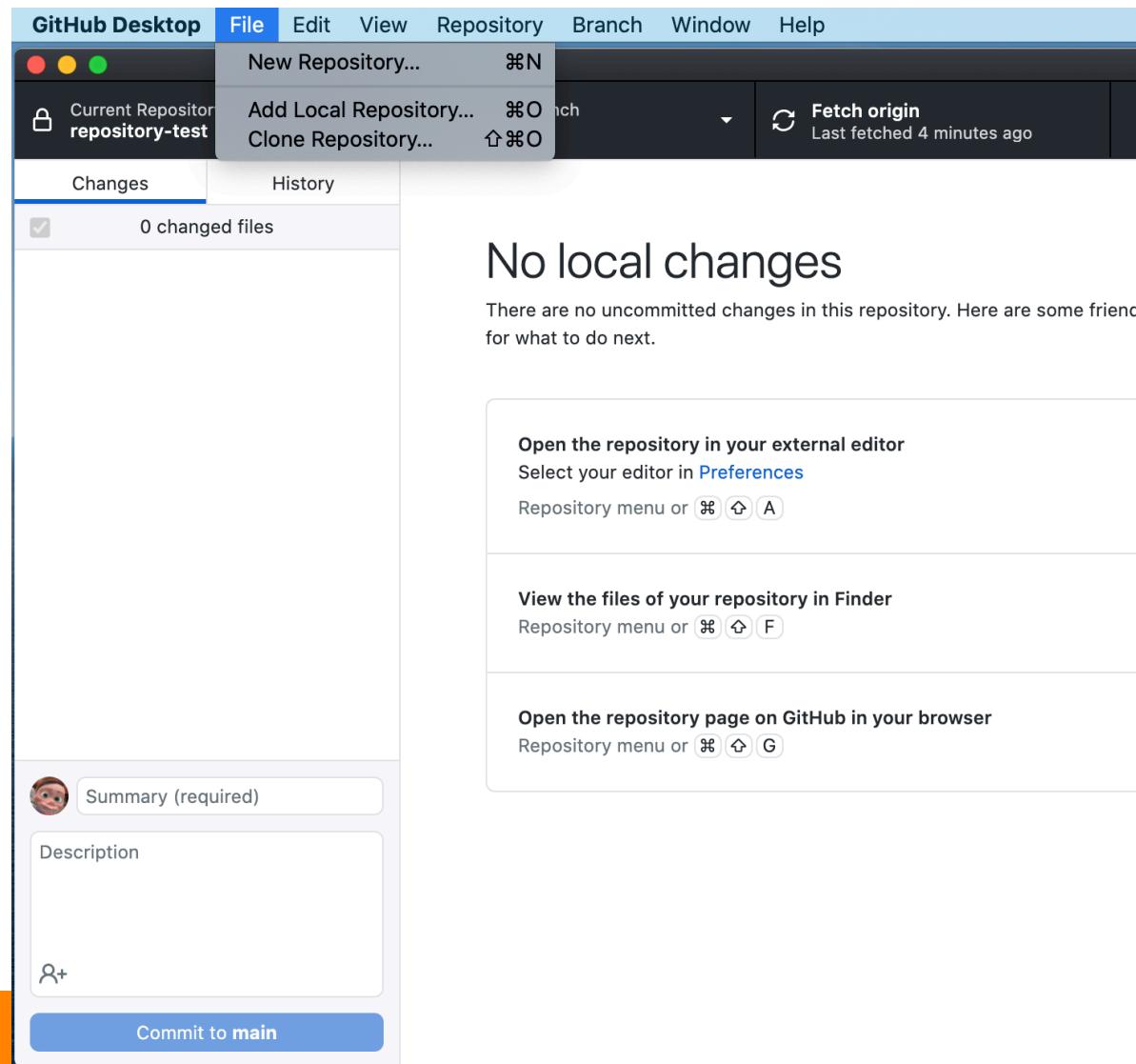
Packages

No packages published

[Publish your first package](#)



Clone the Course Repository (I)



No local changes

There are no uncommitted changes in this repository. Here are some friendly
for what to do next.

Open the repository in your external editor

Select your editor in [Preferences](#)

Repository menu or ⌘ ⌘ A

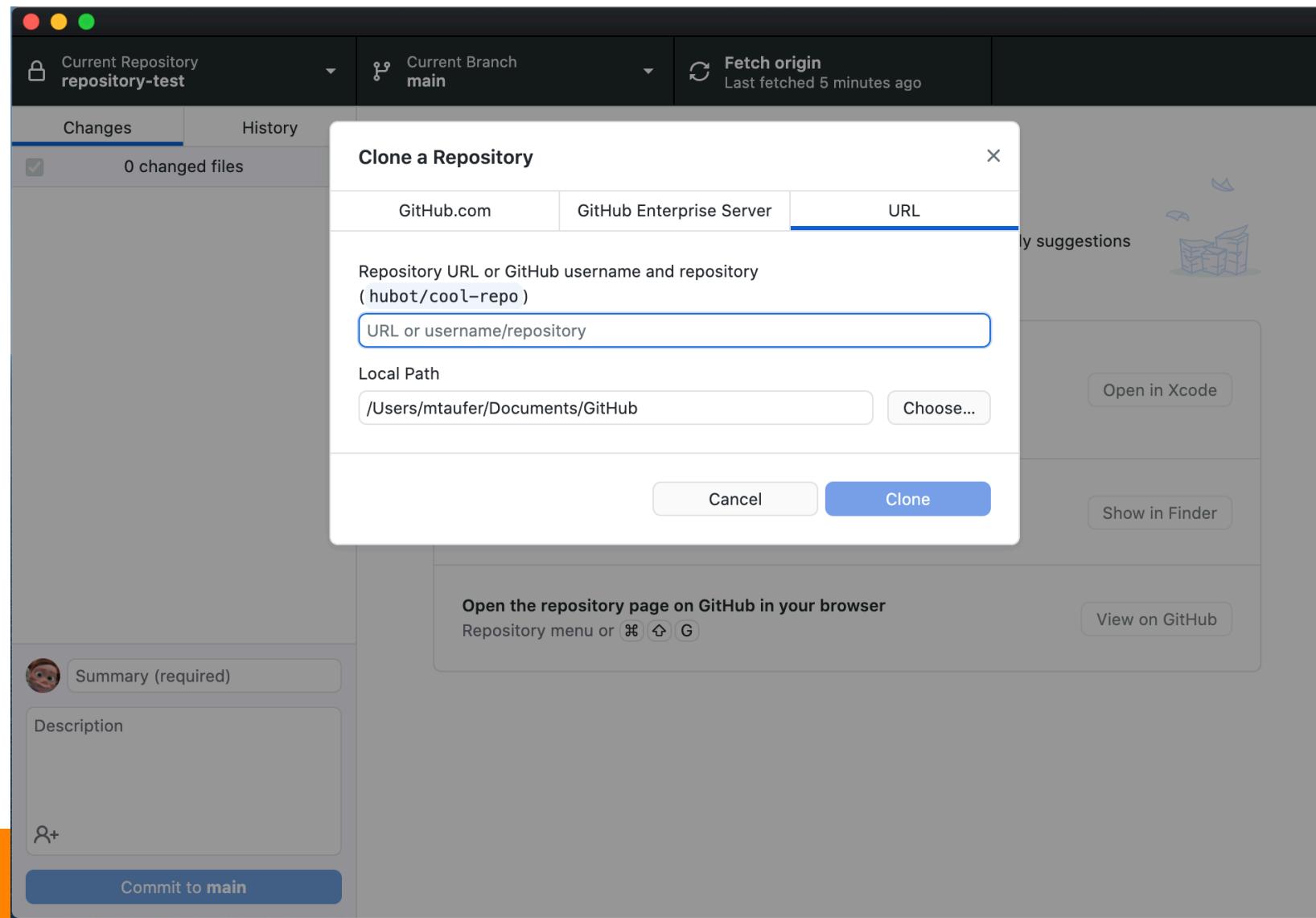
View the files of your repository in Finder

Repository menu or ⌘ ⌘ F

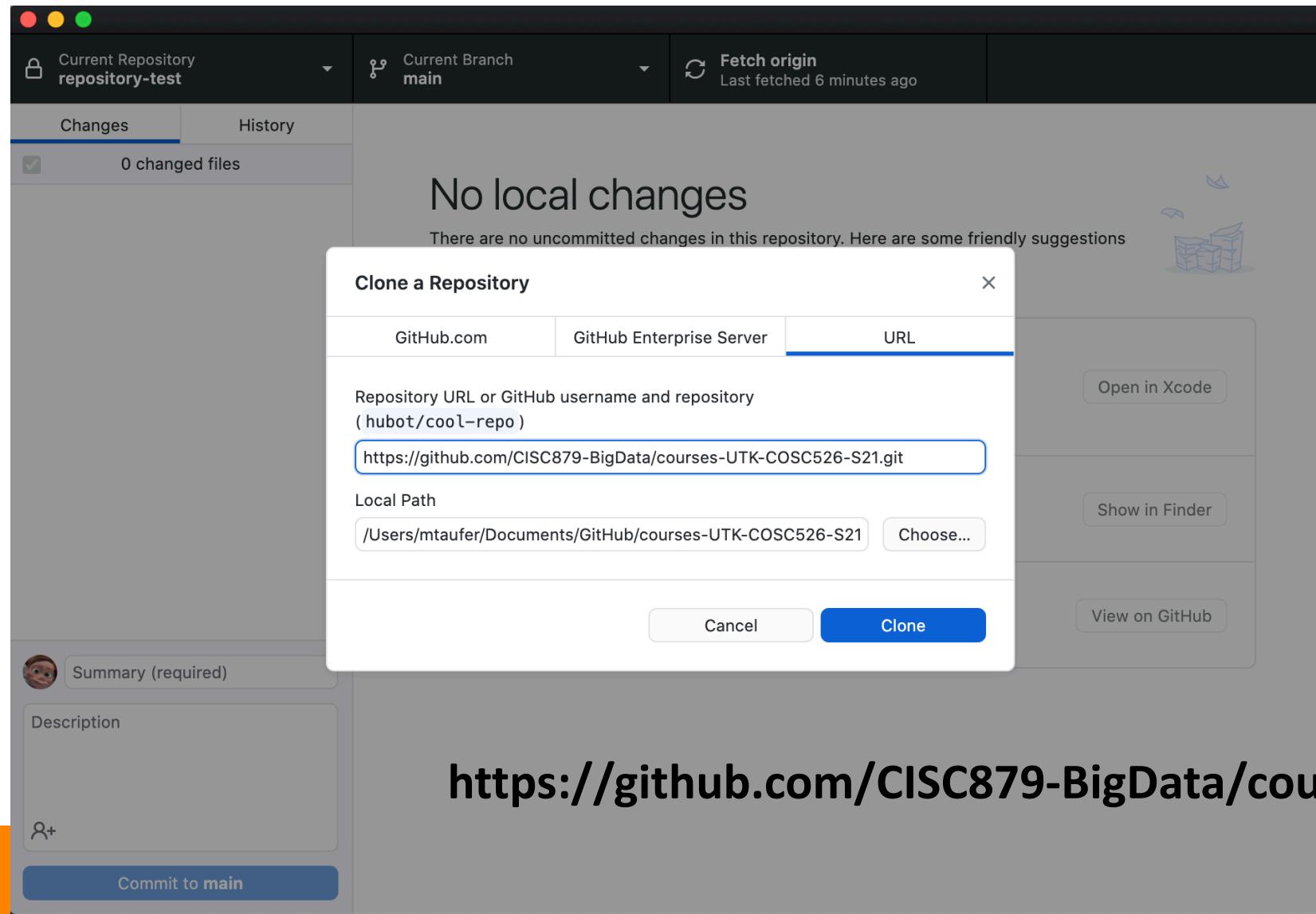
Open the repository page on GitHub in your browser

Repository menu or ⌘ ⌘ G

Clone the Course Repository (II)

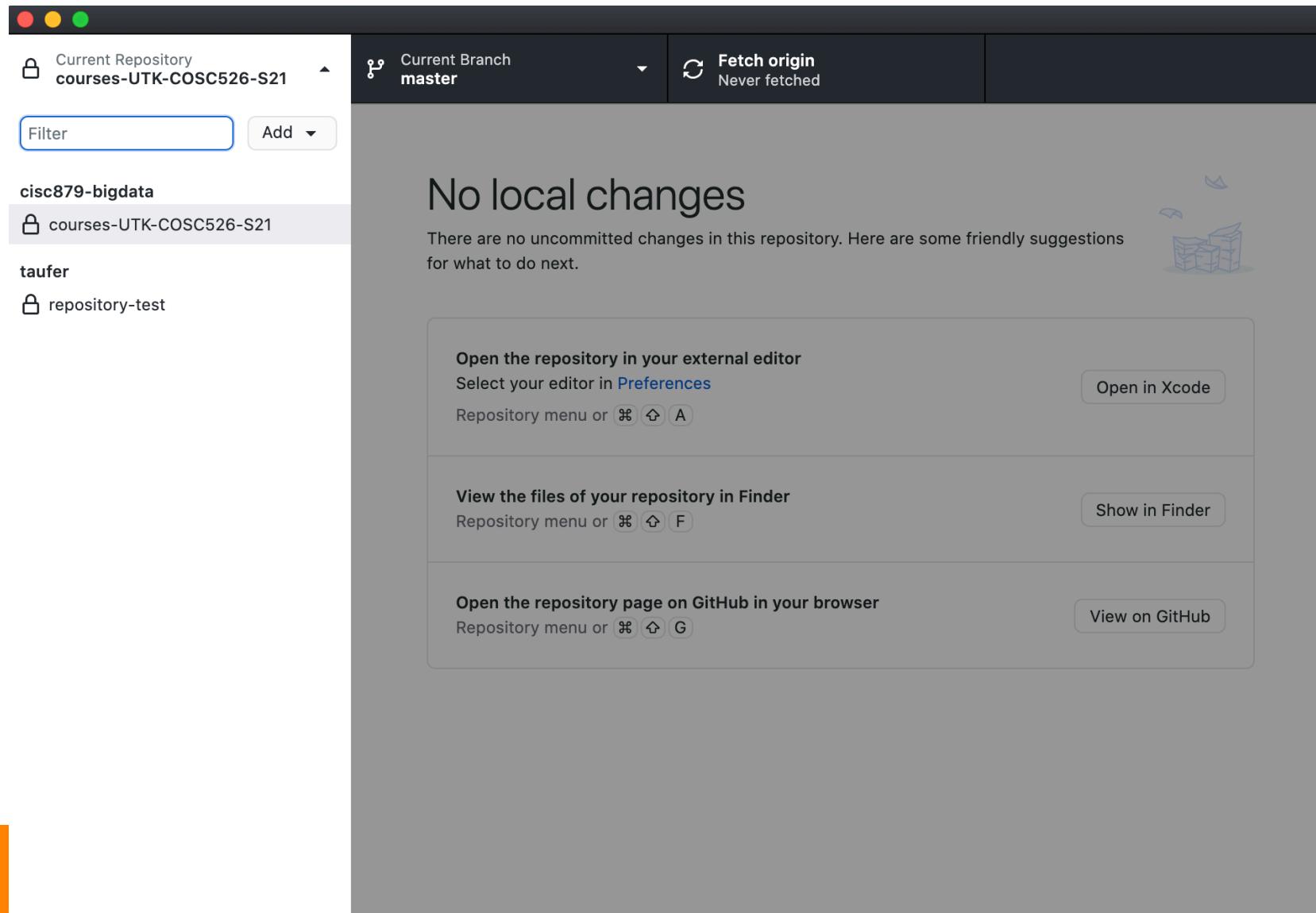


Clone the Course Repository (III)



<https://github.com/CISC879-BigData/courses-UTK-COSC526-S21.git>

Clone the Course Repository (IV)



Clone the Course Repository (V)

[CISC879-BigData / courses-UTK-COSC526-S21](#) Private

[Unwatch](#) 2 [Star](#) 0 [Fork](#) 0

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

[master](#) [1 branch](#) [0 tags](#) [Go to file](#) [Add file](#) [Code](#)

 **nphtan** Initial commit 195ab77 23 days ago 1 commit

[README.md](#) Initial commit 23 days ago

README.md [Edit](#)

courses-UTK-COSC526-S21

About 

No description, website, or topics provided.

[Readme](#)

Releases

No releases published [Create a new release](#)

Packages

No packages published [Publish your first package](#)



Committing Changes to a Repository (I)

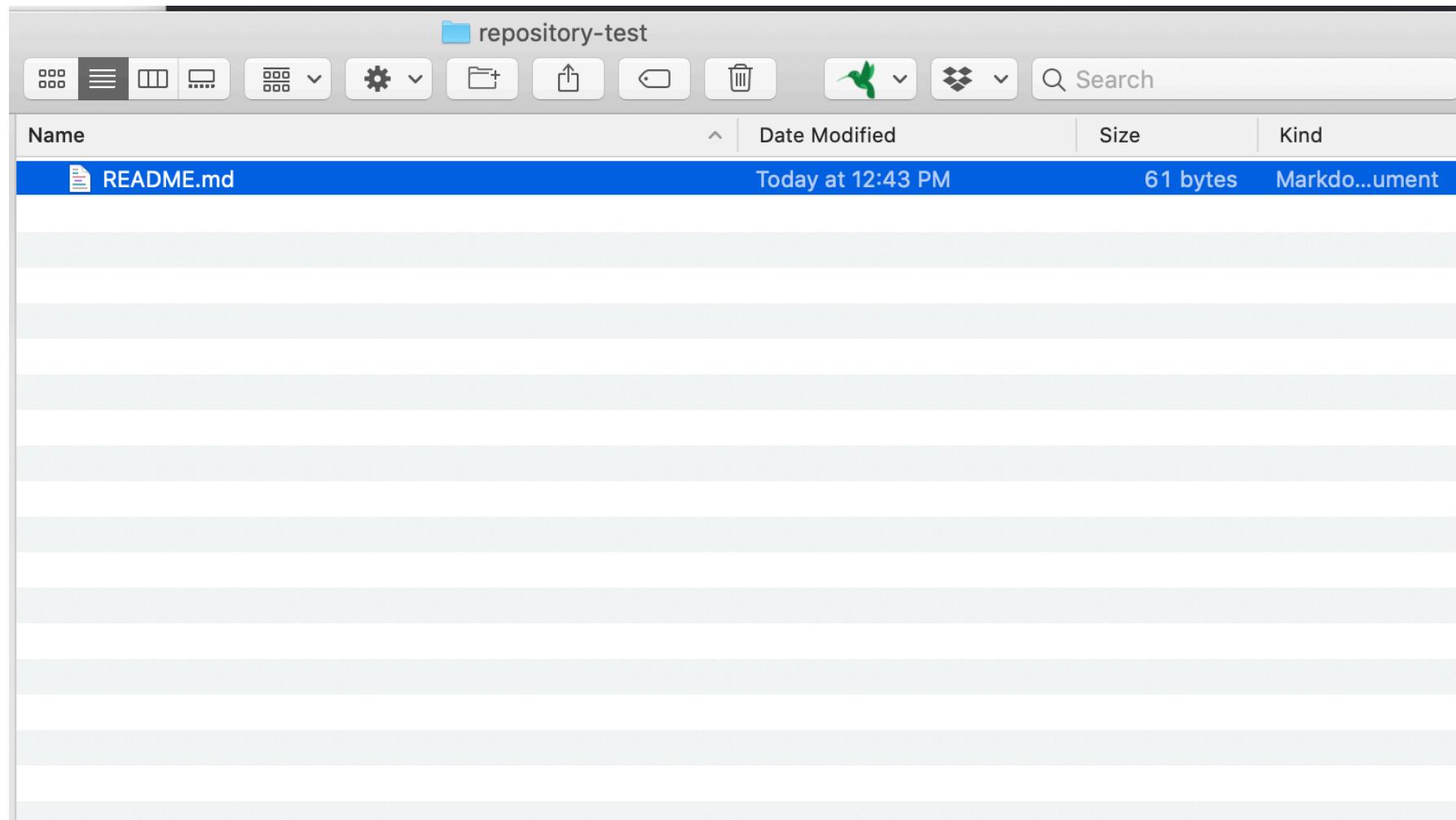
The screenshot shows the GitHub Desktop application interface. At the top, there's a dark header bar with three colored window control buttons (red, yellow, green) on the left. To the right of these are three dropdown menus: 'Current Repository' set to 'repository-test', 'Current Branch' set to 'main', and 'Fetch origin' with a note 'Last fetched 7 minutes ago'. Below the header is a navigation bar with tabs: 'Changes' (which is underlined in blue, indicating it's selected), 'History', and 'Commits'. Under the 'Changes' tab, there's a checkbox followed by the text '0 changed files'. The main content area has a large heading 'No local changes' and a subtext: 'There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.' To the right of this text is a small icon of a stack of papers with arrows above them. Below the text are three sections, each with a title, a 'Repository menu or ⌘ ⌘ A' keybinding, and a 'View on GitHub' button:

- Open the repository in your external editor**
Select your editor in [Preferences](#)
Repository menu or ⌘ ⌘ A [Open in Xcode](#)
- View the files of your repository in Finder**
Repository menu or ⌘ ⌘ F [Show in Finder](#)
- Open the repository page on GitHub in your browser**
Repository menu or ⌘ ⌘ G [View on GitHub](#)

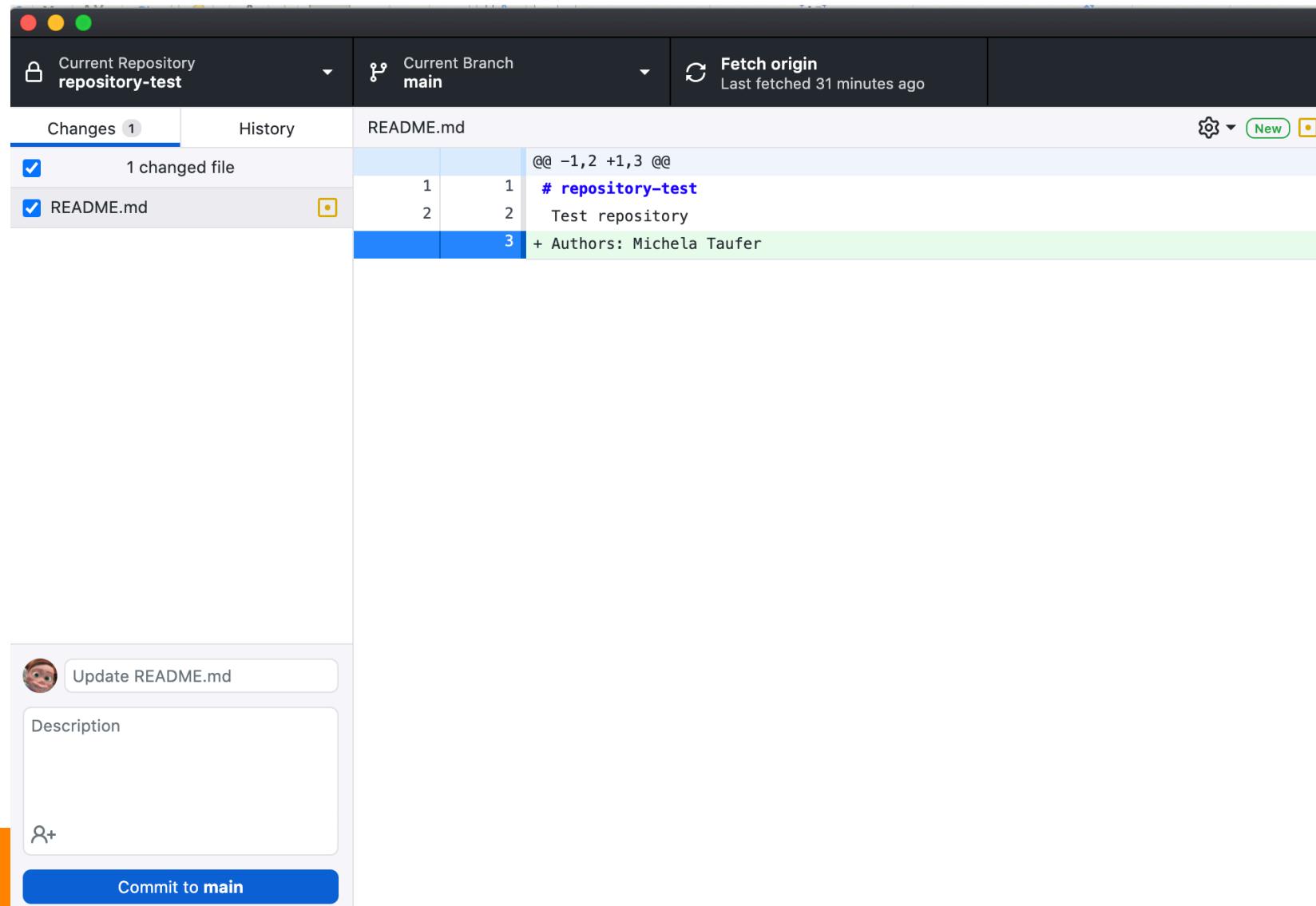
At the bottom left, there's a sidebar with a user profile picture and the text 'Summary (required)'. Below this are two input fields: 'Description' and 'Assignees' (indicated by a person icon). At the very bottom is a large blue button labeled 'Commit to main'.



Committing Changes to a Repository (II)



Committing Changes to a Repository (III)



Committing Changes to a Repository (IV)

The screenshot shows a GitHub desktop application window. At the top, there are three dropdown menus: 'Current Repository' set to 'repository-test', 'Current Branch' set to 'main', and 'Fetch origin' with a note 'Last fetched just now'. Below these are two tabs: 'Changes' (selected) and 'History'. Under 'Changes', it says '1 changed file' and lists 'README.md'. The commit message in the main pane is:

```
@@ -1,2 +1,3 @@
# repository-test
Test repository
+ Authors: Michela Taufer
```

In the bottom-left corner, there's a modal for 'Update README.md' with fields for 'Add author name' and a 'Commit to main' button.

Committing Changes to a Repository (V)

Current Repository: repository-test

Current Branch: main

Push origin
Last fetched just now

Changes History

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Push commits to the origin remote
You have 1 local commit waiting to be pushed to GitHub.
Always available in the toolbar when there are local commits waiting to be pushed or ⌘ P

Open the repository in your external editor
Select your editor in Preferences
Repository menu or ⌘ ⌘ A

View the files of your repository in Finder
Repository menu or ⌘ ⌘ F

Open the repository page on GitHub in your browser
Repository menu or ⌘ ⌘ G

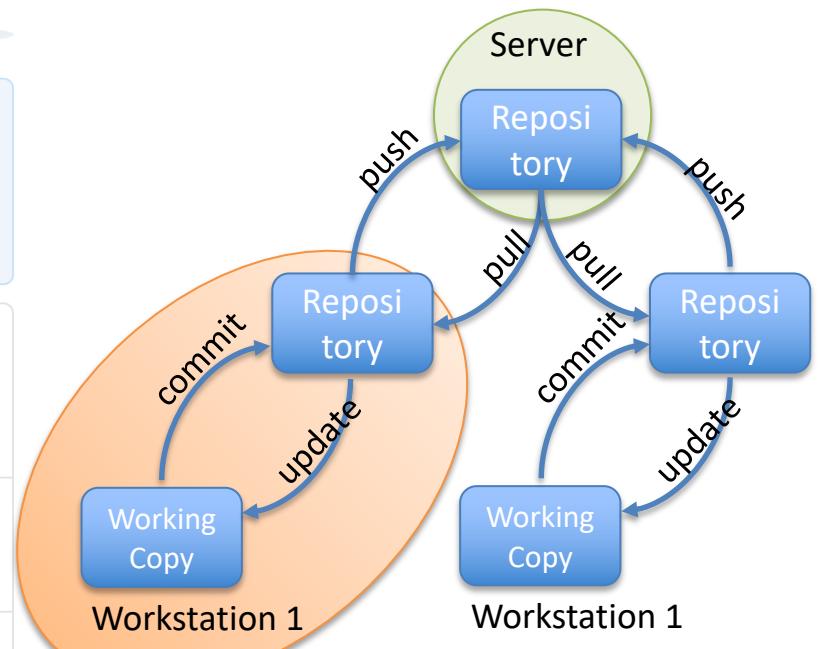
Summary (required)

Description

Commit to main

Committed just now
Update README.md

Undo



Committing Changes to a Repository (VI)

Current Repository
repository-test

Current Branch
main

Fetch origin
Last fetched just now

Changes History

0 changed files

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

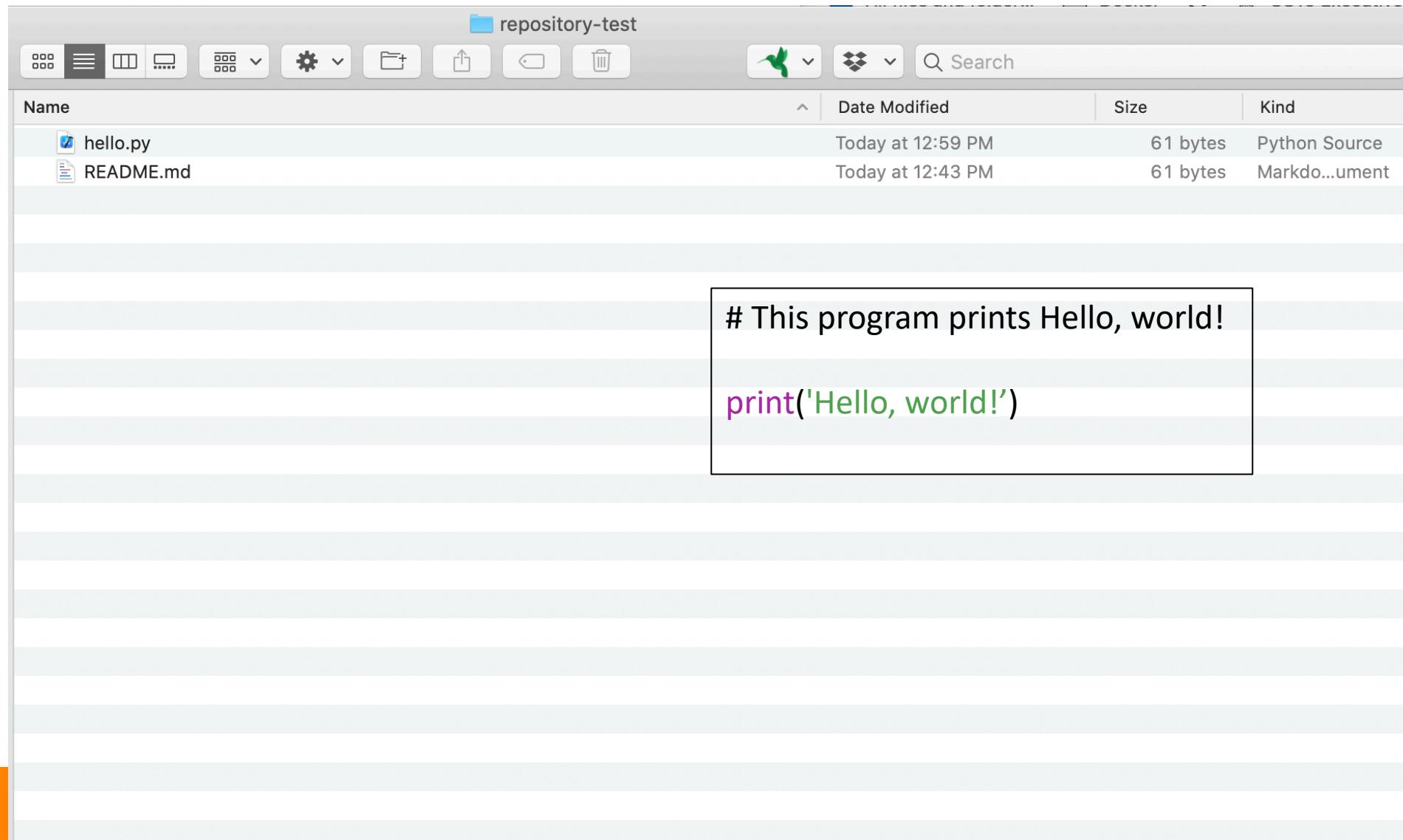
- Open the repository in your external editor
Select your editor in [Preferences](#)
Repository menu or ⌘ ⌘ A [Open in Xcode](#)
- View the files of your repository in Finder
Repository menu or ⌘ ⌘ F [Show in Finder](#)
- Open the repository page on GitHub in your browser
Repository menu or ⌘ ⌘ G [View on GitHub](#)

Summary (required)

Description

+ Commit to main

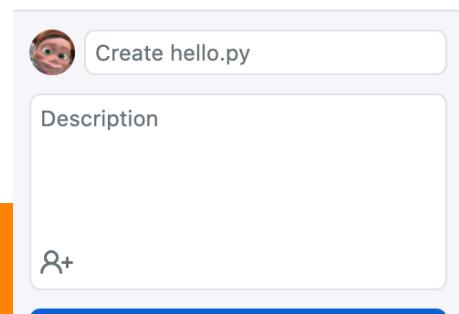
Add and Create new Files in a Repository (I)



Add and Create new Files in a Repository (II)

The screenshot shows a GitHub repository interface. At the top, it displays the current repository as 'repository-test' and the current branch as 'main'. A 'Fetch origin' status message indicates it was last fetched 17 minutes ago. Below this, there are two tabs: 'Changes' (which is selected) and 'History'. Under 'Changes', it shows '1 changed file' named 'hello.py'. The main area displays the diff for 'hello.py':

```
@@ -0,0 +1,4 @@
1+
2+## This program prints Hello, world!
3+
4+print('Hello, world!')
```



Add and Create new Files in a Repository (I)

The screenshot shows a GitHub desktop application interface. At the top, there are three dropdown menus: 'Current Repository' set to 'repository-test', 'Current Branch' set to 'main', and 'Fetch origin' with a note 'Last fetched 17 minutes ago'. Below these are two tabs: 'Changes' (selected) and 'History'. Under 'Changes', there is a list of '1 changed file' named 'hello.py'. The main pane displays the contents of 'hello.py' with the following code:

```
@@ -0,0 +1,4 @@
+
2 +# This program prints Hello, world!
3 +
4 +print('Hello, world!')
```

At the bottom left, a modal dialog is open for creating a new file. It has a placeholder 'Create hello.py' with a user icon. Below it is a text input field containing 'Add hello world test|'. At the bottom of the dialog is a blue button labeled 'Commit to main'.



Add and Create new Files in a Repository (I)

Current Repository
repository-test

Current Branch
main

Push origin
Last fetched 18 minutes ... 1 ↑

Changes History

0 changed files

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

- Push commits to the origin remote
You have 1 local commit waiting to be pushed to GitHub.
Always available in the toolbar when there are local commits waiting to be pushed or ⌘ P
- Open the repository in your external editor
Select your editor in Preferences
Repository menu or ⌘ ⌘ A
- View the files of your repository in Finder
Repository menu or ⌘ ⌘ F
- Open the repository page on GitHub in your browser
Repository menu or ⌘ ⌘ G

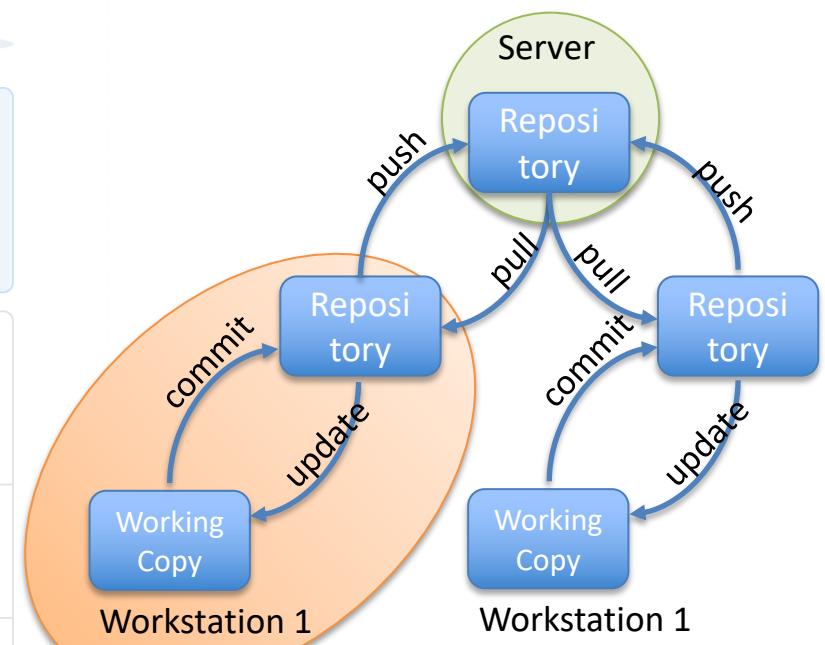
Summary (required)

Description

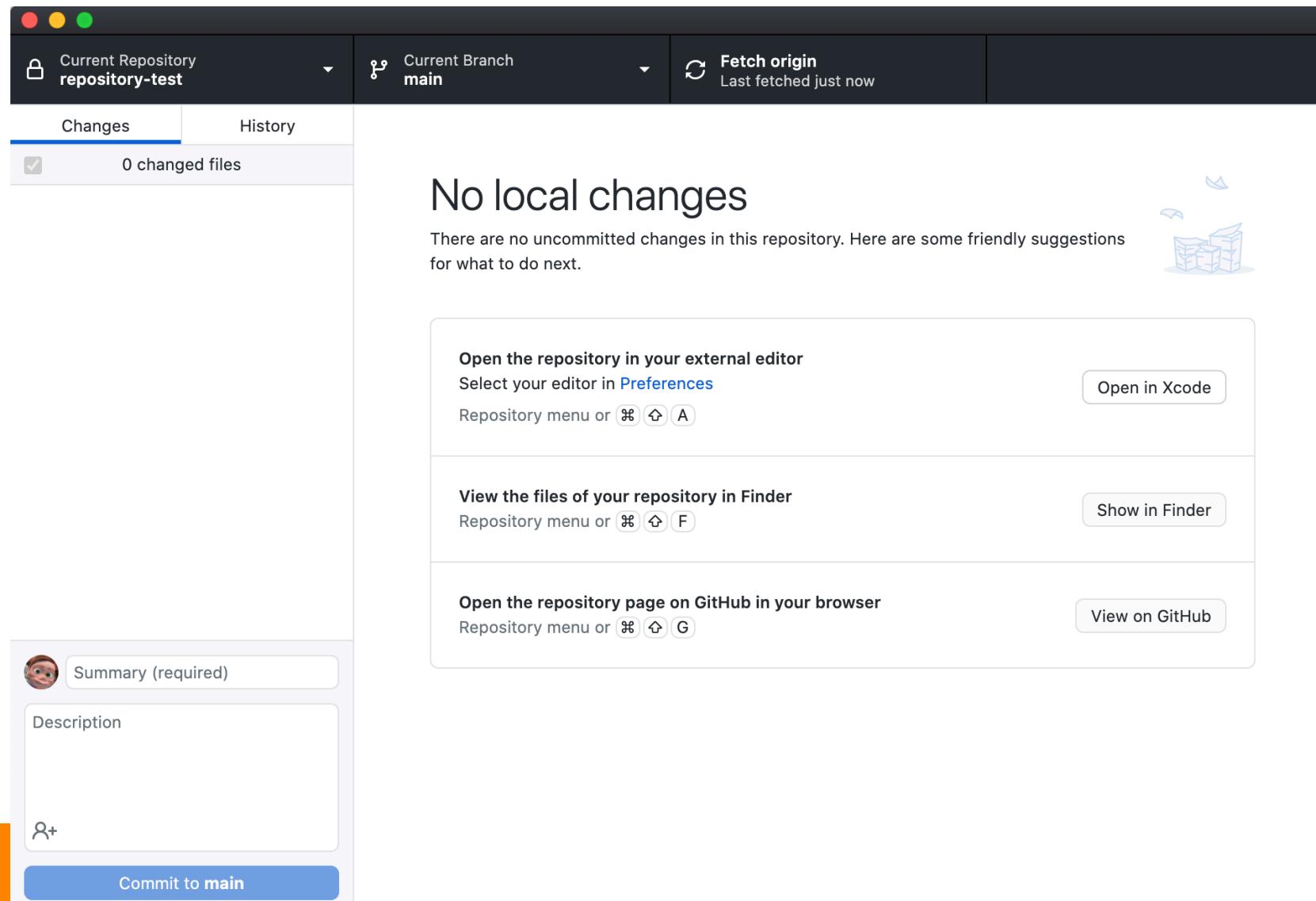
Commit to main

Committed just now
Create hello.py

Undo



Add and Create new Files in a Repository (I)



Pull from Repository (I)

- Not ready yet, bear with us



Command-line interface (CLI)



CLI: Your first time with git and GitHub (I)

CLI for git on Windows: <https://gitforwindows.org>

After installing git and GitHub desktop, if you are using CLI:

- Set up git with your user name and email

```
$ git config --global user.name "Your name here"  
$ git config --global user.email "your_email@example.com"
```

- Set up ssh on your computer
 - Look to see if you have files `~/.ssh/id_rsa` and `~/.ssh/id_rsa.pub`.
 - If not, create such public/private keys:

```
$ ssh-keygen -t rsa -C "your_email@example.com"
```



CLI: Your first time with git and GitHub (II)

- Copy your public key (the contents of the newly created `id_rsa.pub` file) into your clipboard – e.g., on Mac

```
$ pbcopy < ~/.ssh/id_rsa.pub
```

- Paste your ssh public key into your github account settings
 - Go to your github [Account Settings](#)
 - Click “[SSH Keys](#)” on the left.
 - Click “Add SSH Key” on the right.
 - Add a label (like “My laptop”) and paste the public key into the big text box.



CLI: Your first time with git and GitHub (III)

- In a terminal/shell, type the following to test it:

```
$ ssh -T git@github.com
```

- If it says something like the following, it worked:

```
Hi username! You've successfully authenticated, but Github does  
not provide shell access.
```



Use git and GitHub

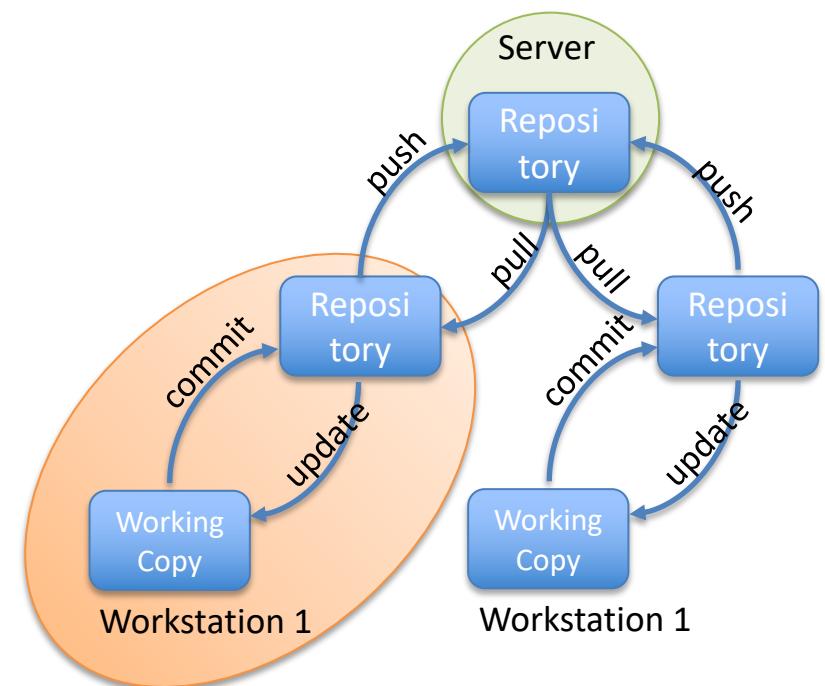
- The routine use of git involves just a few commands:
 - init
 - add and commit
 - push and pull
 - status
 - diff



A new repo from scratch

- Create a directory to contain the project
 \$ mkdir YourRepos
- Go into the new directory
- Type *git init*
 \$ *git init*
- Create a new file – ReadMe.md is a good start
 \$ echo "# YourRepos" >> ReadMe.md
- Type *git add* to add the file
 \$ *git add* README.md
- Type *git commit* to commit the file
 \$ *git commit -m* "first commit"

At this point, your repos is ONLY local



Keep the repo clean

- Create a **.gitignore** file to indicate all of the files you don't want to track in a (sub)directory

```
$ git add .gitignore
```

- Create a **.gitignore_global** file to indicate all of the files you don't want to track for the entire directory. You have to tell git about the global .gitignore

```
$ git add .gitignore_global
```

```
$ git config --global core.excludesfile ~/.gitignore_global
```



.gitignore_global and .gitignore

*~

.*~

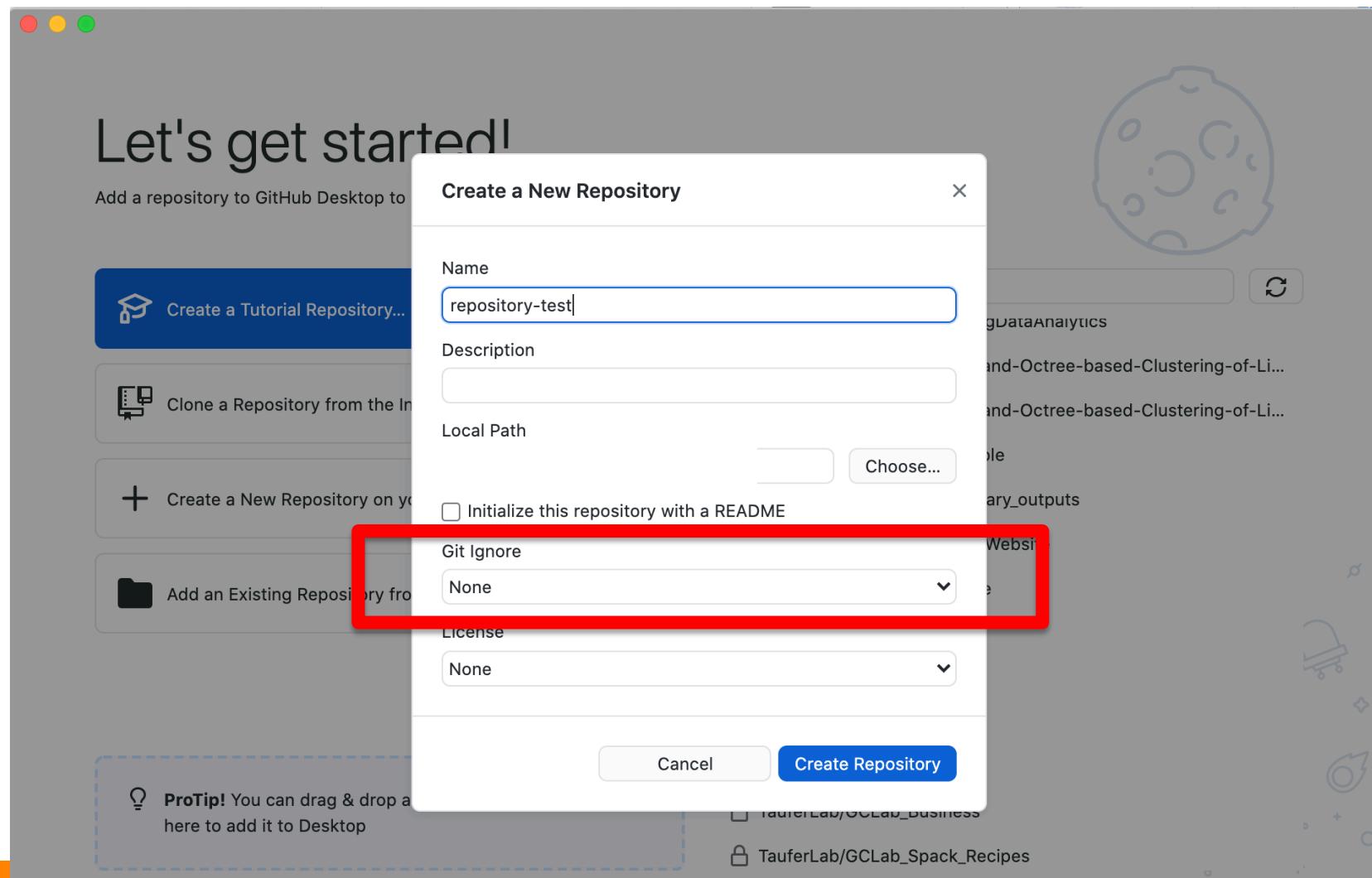
.DS_Store

.Rhistory

.RData

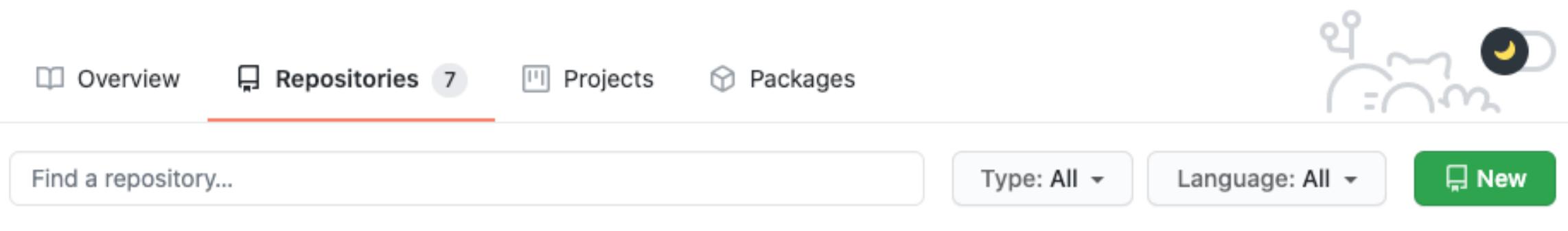


Create a Repository (I) repeat



Connect new repo to GitHub (I)

- Go to [github](#)
- Log in to your account
- Click the “New” repository button in the top-right
- Click the “Create repository” button



Create a new repository

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

Owner *



taufer A small blue dropdown arrow indicating a dropdown menu.

Repository name *

/

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

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](#).

Choose a license

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

[Create repository](#)

Connect new repo to GitHub (II)

- Clone a new or old repository

```
git clone https://github.com/YourAccount/YourRepos.git
```

The screenshot shows a GitHub repository page for 'repository-test'. The top navigation bar indicates 1 branch and 0 tags. The repository details show a commit from 'taufers' creating 'hello.py'. Below the commits is a file named 'README.md'. The main content area displays the text 'repository-test' and 'Test repository Authors: Michela Taufer'. On the right side, there's an 'About' section with links for 'Test repository', 'Readme', 'Releases', and 'Packages'. A prominent feature is a 'Code' dropdown menu that is open, showing options for cloning the repository via HTTPS, SSH, or GitHub CLI. The HTTPS URL, <https://github.com/taufers/repositories>, is highlighted with a blue box.

Add and commit (I)

- Add completely new files as well as to “add” modifications to files that already exist in the repository (e.g., README.md)

```
$ mkdir Lecture01
```

```
$ touch Lecture01/README.md
```

```
$ emacs README.md
```

```
$ git add Lecture01 Lecture01/README.md README.md
```

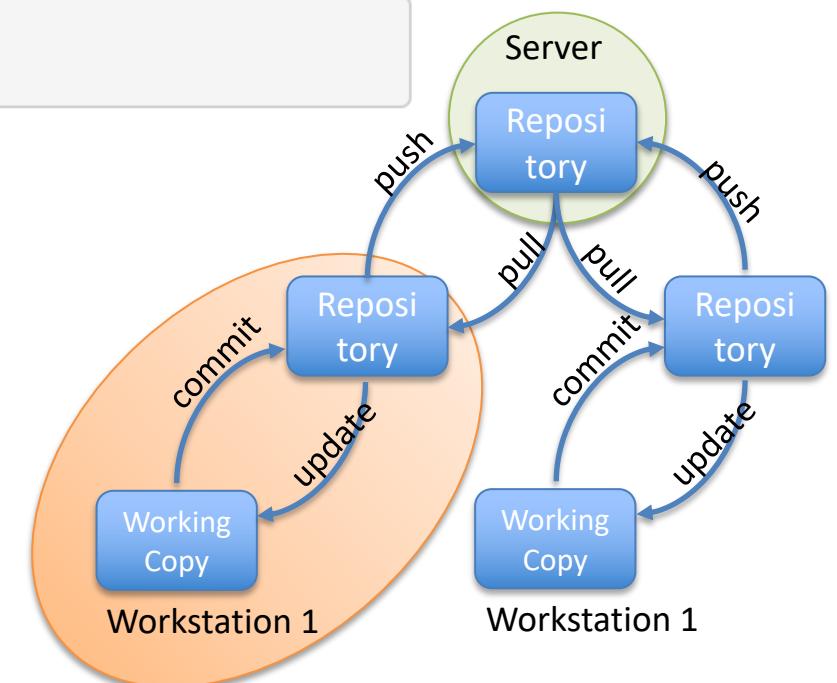


Add and commit (II)

- Use git commit to add the modifications to the repo

\$ git commit OR \$ git commit -m “Message: 40 / 60 characters”

```
$ git commit -m "Fix such and such"
```

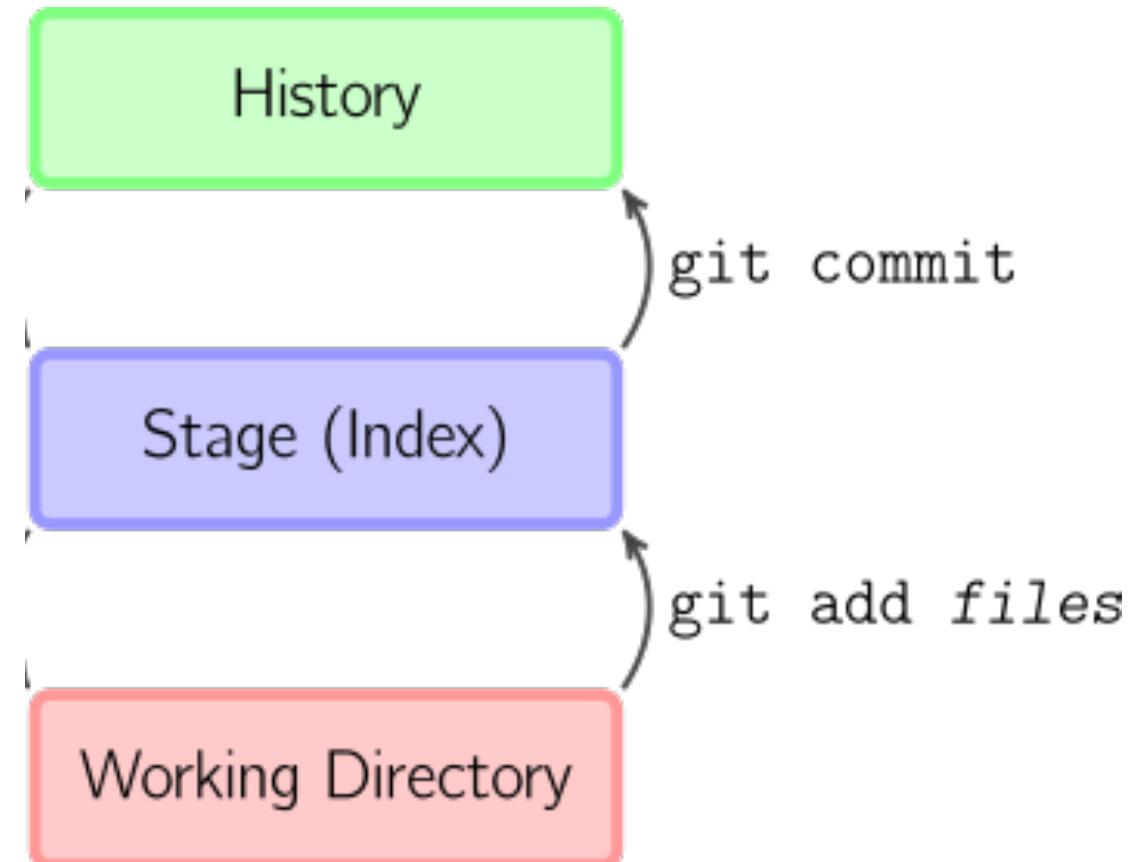
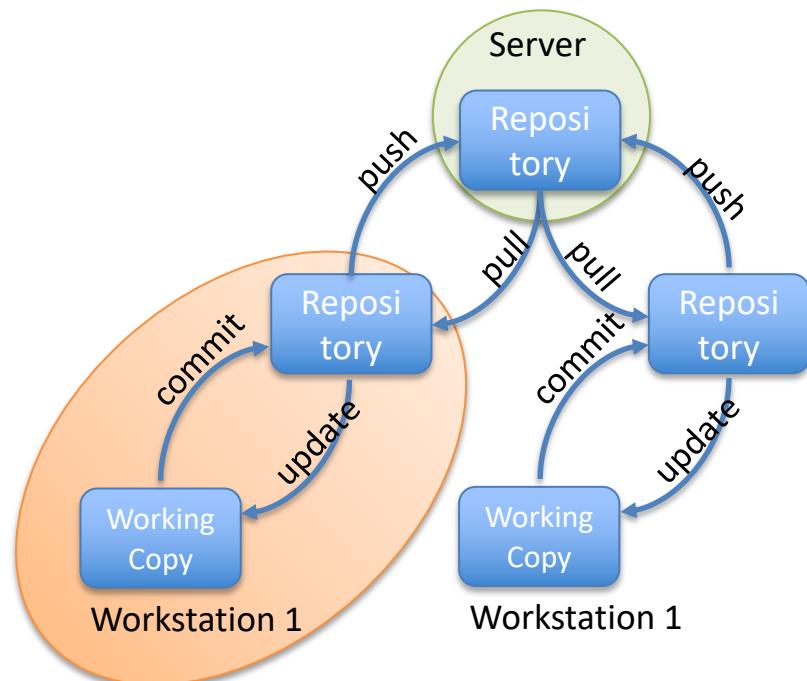


Add and commit (II)

```
$ git add <filename>
```

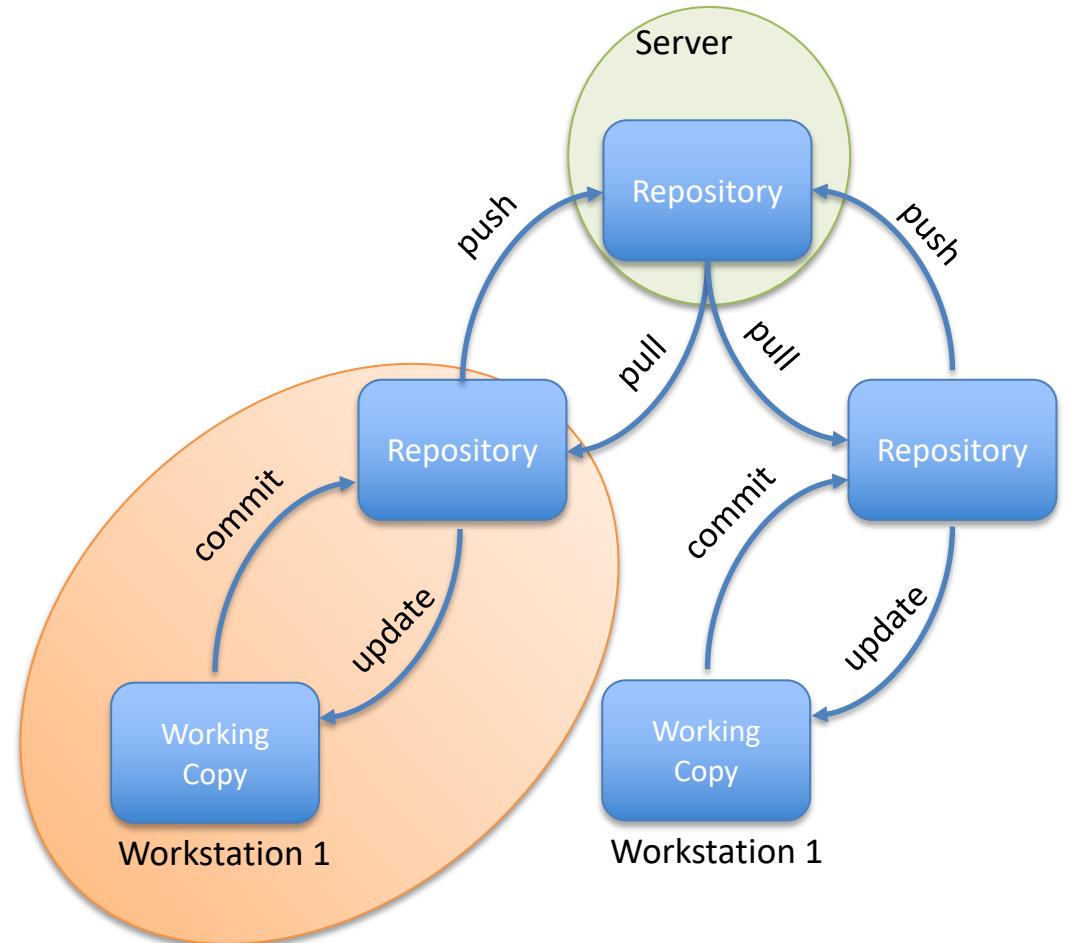
```
$ git add
```

```
$ git commit -m "Commit message"
```



Push to and pull from github

- After any *git add* and *git commit* commands, changes are in the **HEAD** of your **local working copy**
- To push committed changes to your remote repository - github, type:
 \$ git push
- To pull committed changes from your remote repository - github, type:
 \$ git pull



Differences github

- To extract differences and changes, type:

```
$ git diff
```

- To extract changes in one specific file, type

```
$ git diff README.md
```

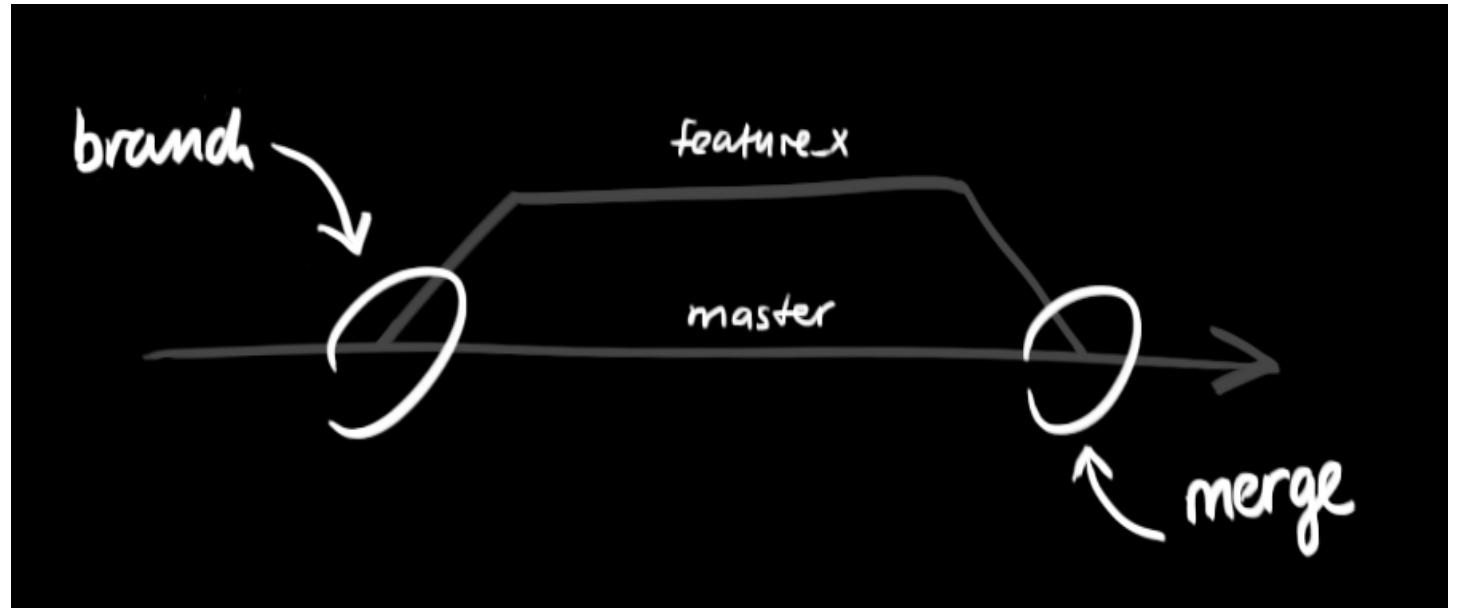


More about GitHub (Optional)



Branching and branches

- Branches are used to develop features isolated from each other
- The *master* branch is the "default" branch when you create a repository
- Use other branches for development and merge them back to the master branch upon completion



Branching and branches

- Create a new branch named "feature_x" and switch to it:
 \$ git checkout -b feature_x
- Switching back to master:
 \$ git checkout master
- Delete the feature branch:
 \$ git branch -d feature_x
- Pushing a branch for others to use:
 \$ git push origin <branch>



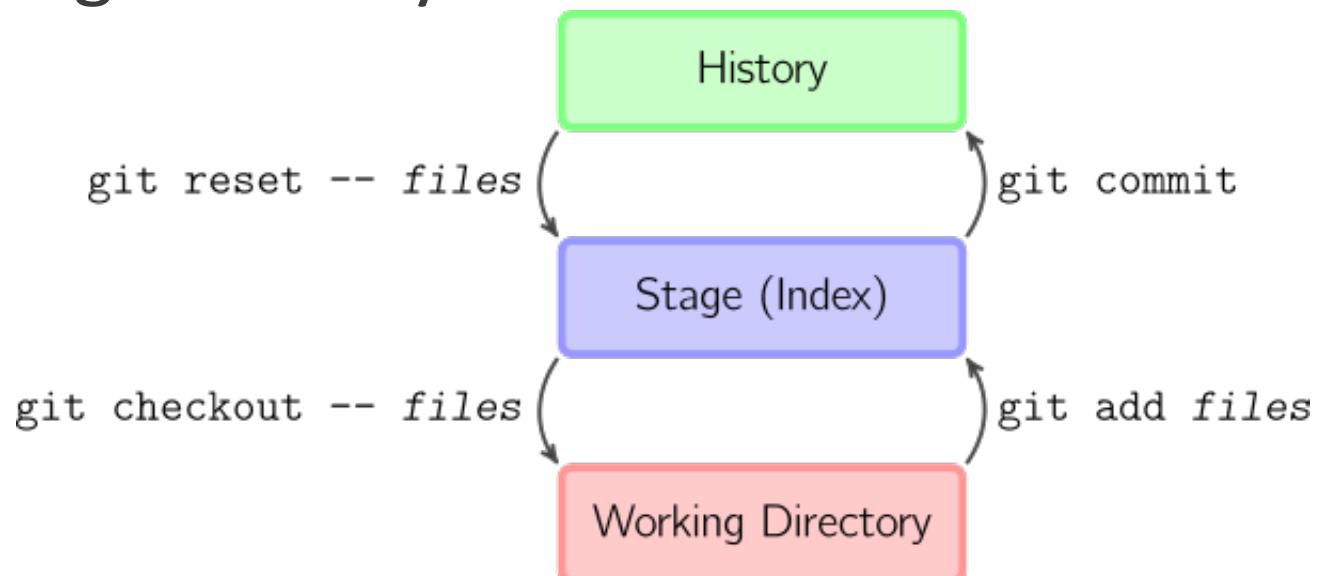
Syncing & Merging

- Updating your local repository to the newest commit:
 - `git pull`
- Merge another branch into your active branch (e.g. master):
 - `git merge <branch>`
- After resolving merge conflicts, you need to mark them as merged:
 - `git add <filename>`
- You can also preview them by using:
 - `git diff <source_branch> <target_branch>`



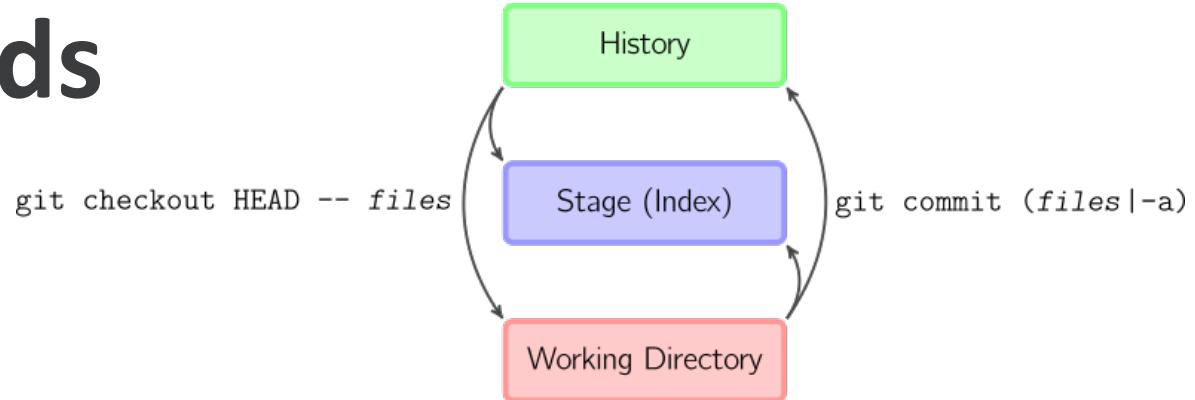
Undoing

- Unstage changes:
 - `git reset HEAD <file>...`
- Discard changes in working directory:
 - `git checkout -- <file> ...`



Combining Commands

- Add + Commit:
 - `git commit -a`
 - equivalent to running `git add` on all filenames that existed in the latest commit, and then running `git commit`
- Reset + Checkout:
 - `git checkout HEAD -- <file> ...`
 - copies *files* from the latest commit to both the stage and the working directory



Python, Anaconda, and Jupyter Notebook



Python and Anaconda

- **Python:** It is Python 3.7!
- **Anaconda:** Python distribution
 - Include many popular packages by default, including Jupyter notebook
 - Make installing additional packages easy
- Your task:
 - Install Anaconda and launch the Jupyter notebook

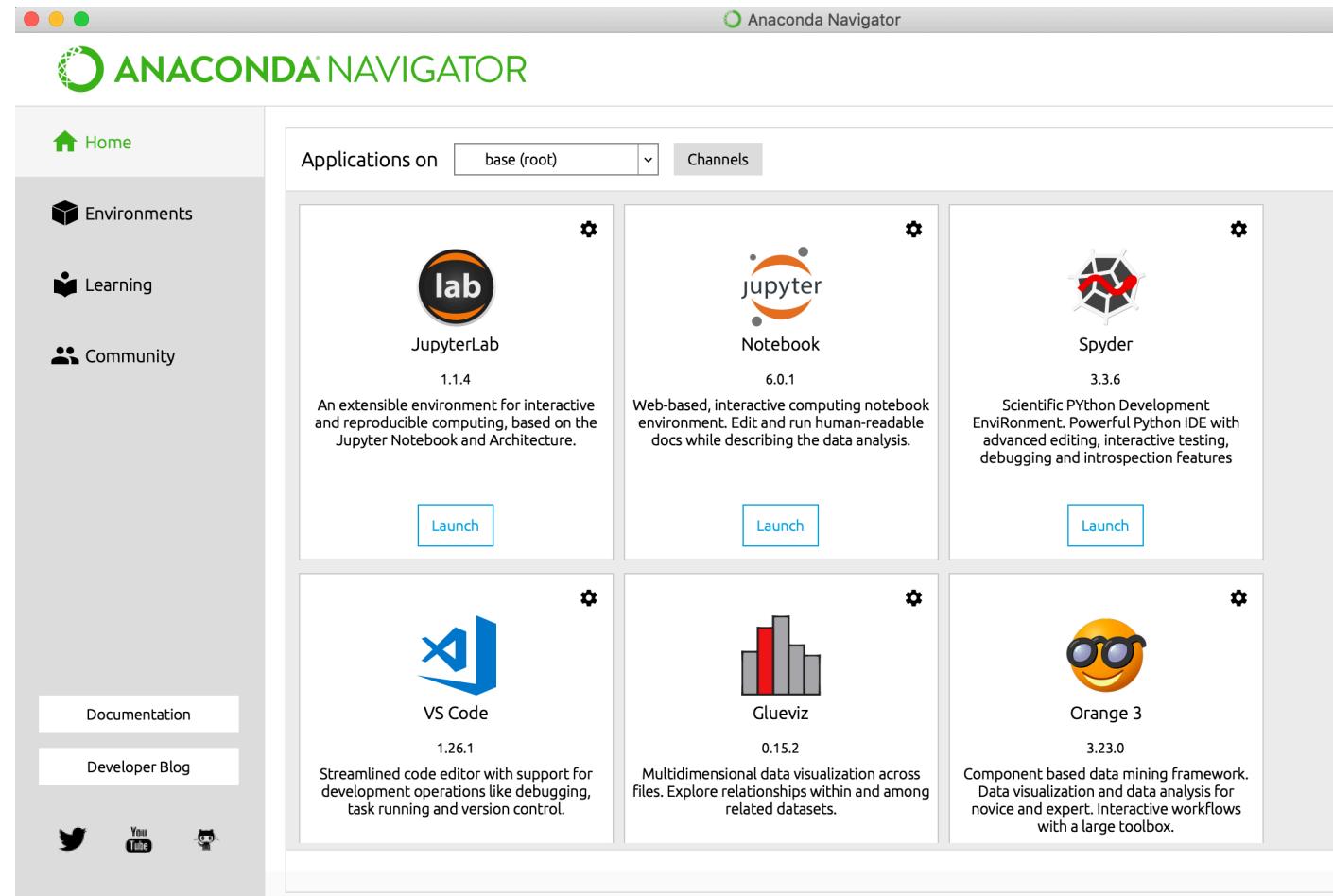


Install Anaconda

- To install Anaconda (Distribution, not Enterprise or Cloud), follow the installation instructions for your operating system:
- Windows
<https://docs.anaconda.com/anaconda/install/windows/>
- Mac
<https://docs.anaconda.com/anaconda/install/mac-os/>
- Linux
<https://docs.anaconda.com/anaconda/install/linux/>



Anaconda Navigator



Jupyter Notebook

- **Jupyter notebook:** Our notebook for data analytics
- Programming in a browser
 - Create code in a cell – code in edit mode
 - Run code in a cell – code in command mode
 - Write text before and after code cells – markdown
- Your task:
 - Launch Jupyter from Anaconda GUI (or command line)



Open your assignment directory

The screenshot shows a GitHub repository page for the repository **CISC879-BigData / 2018F-CS594-CS690**. The repository is private, has 5 issues, 0 pull requests, 0 projects, a wiki, insights, and settings. The branch is master. The repository contains files such as Assignment01.ipynb, StartHere.ipynb, StartHere.pdf, data.csv, and data.tsv. The latest commit was made by imnasnainaec 3 days ago.

File	Commit Message	Time Ago
Assignment01.ipynb	Initial commit	4 days ago
StartHere.ipynb	StartHere edits.	3 days ago
StartHere.pdf	StartHere edits.	3 days ago
data.csv	Initial commit	4 days ago
data.tsv	Initial commit	4 days ago

Open your assignment directory

jupyter

Quit Logout

Files Running Clusters

Select items to perform actions on them.

Upload New 

<input type="checkbox"/> 0		/ 00_git_repos / 2018F-CS594-CS690 / Assignment01	Name 	Last Modified	File size
<input type="checkbox"/>		..		seconds ago	
<input type="checkbox"/>		images		3 hours ago	
<input type="checkbox"/>		Assignment01.ipynb	Running	3 hours ago	11.1 kB
<input type="checkbox"/>		StartHere.ipynb	Running	an hour ago	6.87 kB
<input type="checkbox"/>		data.csv		3 hours ago	1.06 kB
<input type="checkbox"/>		data.tsv		3 hours ago	1.05 kB
<input type="checkbox"/>		StartHere.pdf		3 hours ago	289 kB

Create your own notebook

jupyter

Files Running Clusters

Select items to perform actions on them.

0 / 00_git_repos / 2018F-CS594-CS690 / Assignment01

Name ↴

Upload New ↴

Notebook: Python 3

Other: Text File

Folder

Terminal

<input type="checkbox"/>	Assignment01.ipynb	Run
<input type="checkbox"/>	StartHere.ipynb	Run
<input type="checkbox"/>	data.csv	3 hours ago 1.06 kB
<input type="checkbox"/>	data.tsv	3 hours ago 1.05 kB
<input type="checkbox"/>	StartHere.pdf	3 hours ago 289 kB

..

images

Assignment01.ipynb

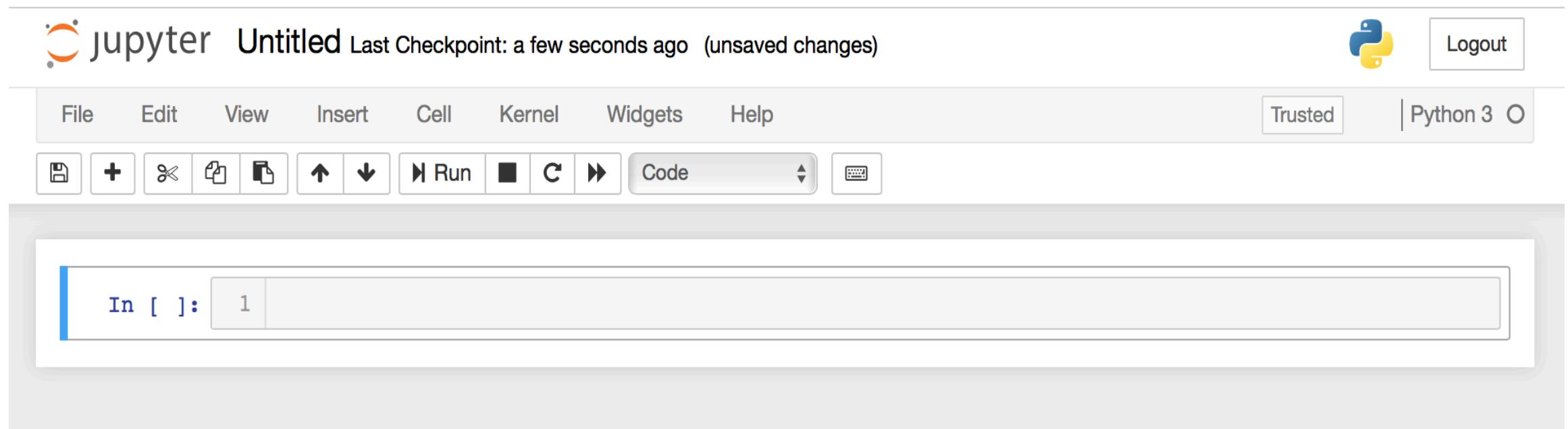
StartHere.ipynb

data.csv

data.tsv

StartHere.pdf

Create your own notebook

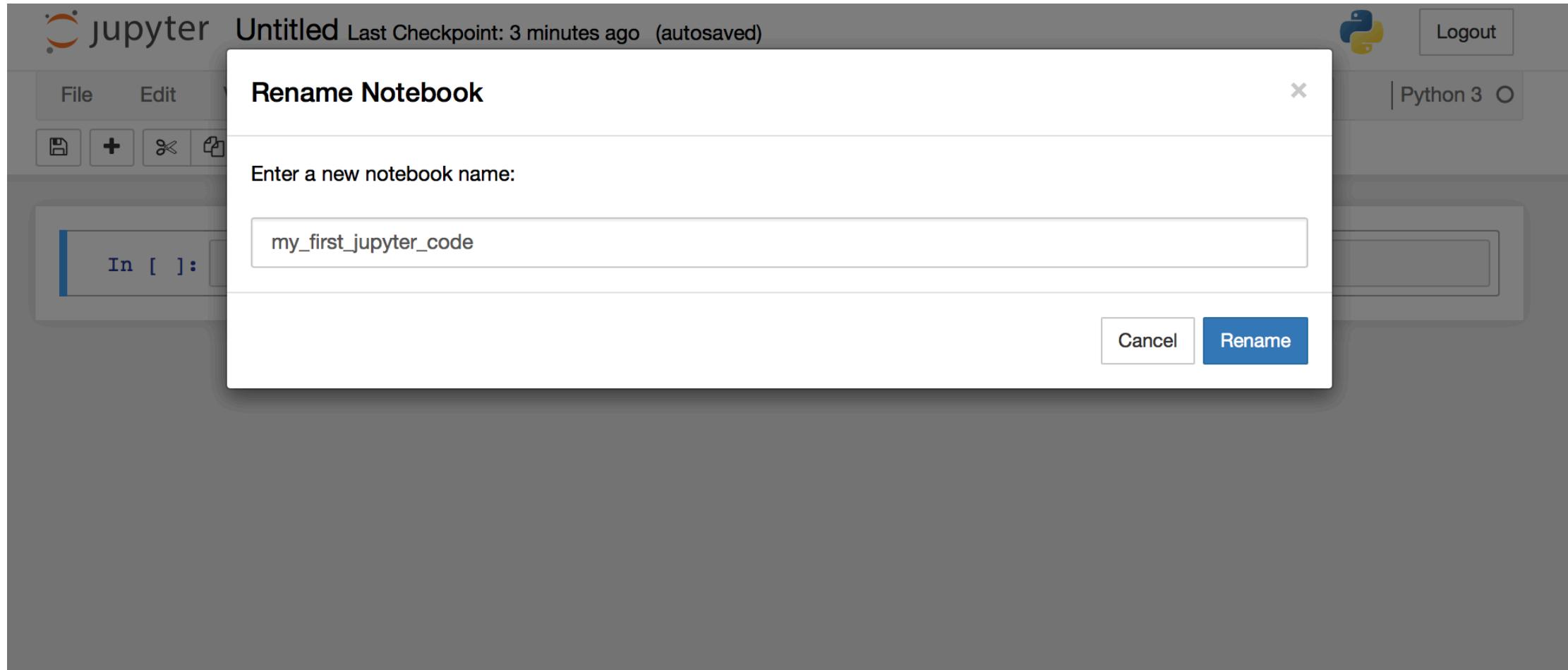


Create your own notebook

The screenshot shows a Jupyter Notebook interface. At the top, there is a header with the Jupyter logo, the title "Untitled", and status information "Last Checkpoint: a minute ago (unsaved changes)". To the right of the header are a Python logo icon and a "Logout" button. Below the header is a navigation bar with tabs: File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. To the right of the navigation bar are buttons for "Trusted" and "Python 3". A dropdown menu is open under the "File" tab, listing the following options: New Notebook, Open..., Make a Copy..., Rename..., Save and Checkpoint, Revert to Checkpoint, Print Preview, Download as, Trusted Notebook, and Close and Halt. The main workspace area is currently empty.

- New Notebook
- Open...
- Make a Copy...
- Rename...
- Save and Checkpoint
- Revert to Checkpoint
- Print Preview
- Download as
- Trusted Notebook
- Close and Halt

Rename the file



Create your own notebook

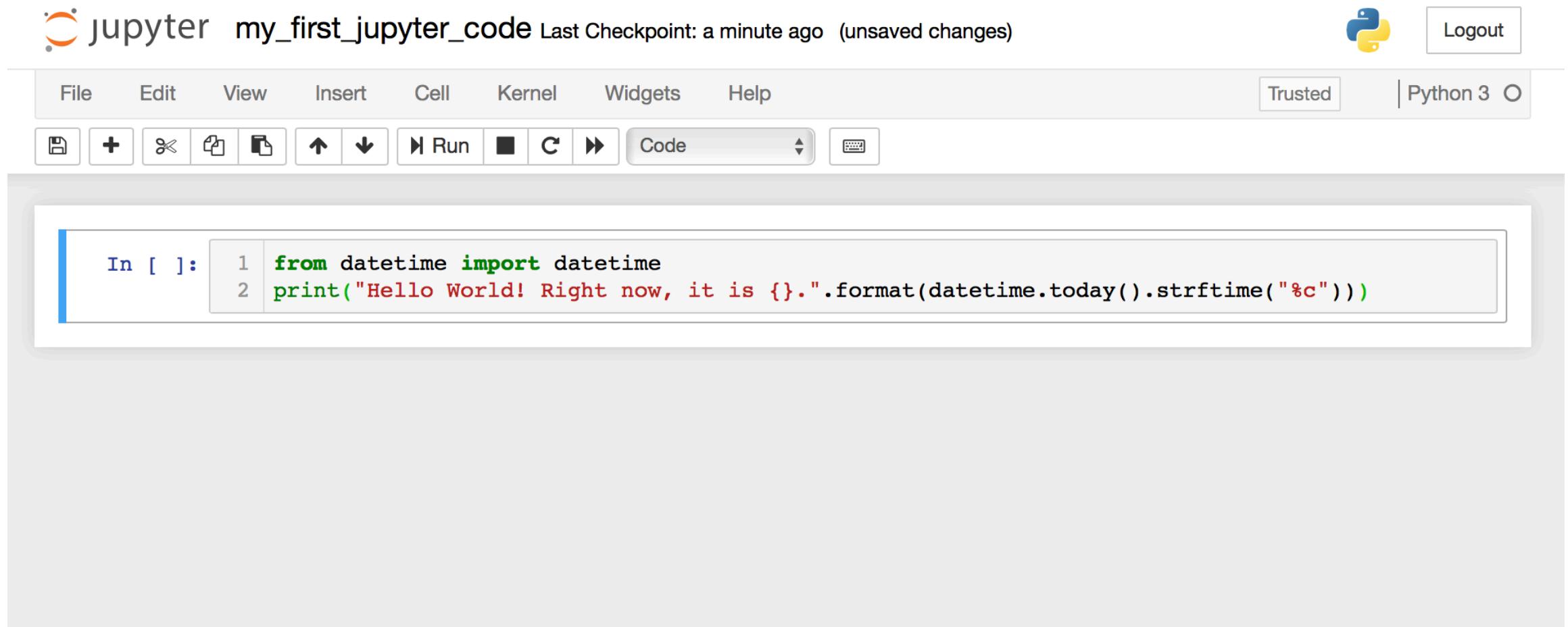
The screenshot shows a Jupyter Notebook interface. At the top left is the Jupyter logo. On the right are 'Quit' and 'Logout' buttons. Below the logo is a navigation bar with 'Files' (selected), 'Running', and 'Clusters' tabs. A message 'Select items to perform actions on them.' is displayed above a file list. The file list shows the contents of the directory `/ 00_git_repos / 2018F-CS594-CS690 / Assignment01`. The columns are 'Name', 'Last Modified', and 'File size'. The files listed are:

	Name	Last Modified	File size
<input type="checkbox"/>	0	seconds ago	
<input type="checkbox"/>	..	seconds ago	
<input type="checkbox"/>	images	3 hours ago	
<input type="checkbox"/>	Assignment01.ipynb	Running 3 hours ago	11.1 kB
<input type="checkbox"/>	my_first_jupyter_code.ipynb	Running seconds ago	555 B
<input type="checkbox"/>	StartHere.ipynb	Running an hour ago	6.87 kB
<input type="checkbox"/>	data.csv	3 hours ago	1.06 kB
<input type="checkbox"/>	data.tsv	3 hours ago	1.05 kB
<input type="checkbox"/>	StartHere.pdf	3 hours ago	289 kB

Buttons for 'Upload' and 'New' are located at the top right of the file list area.



Create a code cell



The screenshot shows a Jupyter Notebook interface. At the top, there's a header bar with the Jupyter logo, the title "jupyter my_first_jupyter_code", a timestamp "Last Checkpoint: a minute ago (unsaved changes)", a Python logo icon, and a "Logout" button. Below the header is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". To the right of the menu are "Trusted" and "Python 3" buttons. A toolbar below the menu contains icons for file operations like saving, creating, and deleting, along with buttons for "Run", "Cell", "Code", and "Cell Kernel". The main area is a code cell with a blue header bar labeled "In []:". Inside the cell, two lines of Python code are displayed:

```
In [ ]: 1 from datetime import datetime  
2 print("Hello World! Right now, it is {}".format(datetime.today().strftime("%c")))
```

Run your code code

The figure shows a Jupyter Notebook interface. At the top, there's a header with the logo, the notebook name "my_first_jupyter_code", the last checkpoint time ("Last Checkpoint: 2 minutes ago (unsaved changes)", and a Python logo icon. To the right of the Python icon is a "Logout" button. Below the header is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". To the right of the menu are buttons for "Trusted" and "Python 3". The main area has a toolbar with icons for file operations like open, save, and new, followed by buttons for "Run", "Cell", "Kernel", and "Code". A code cell is visible, containing the following Python code:

```
In [1]: 1 from datetime import datetime  
2 print("Hello World! Right now, it is {}".format(datetime.today().strftime("%c")))
```

The output of the code cell is displayed below it: "Hello World! Right now, it is Sat Aug 25 13:47:49 2018.". A new code cell is partially visible at the bottom, starting with "In []: 1".

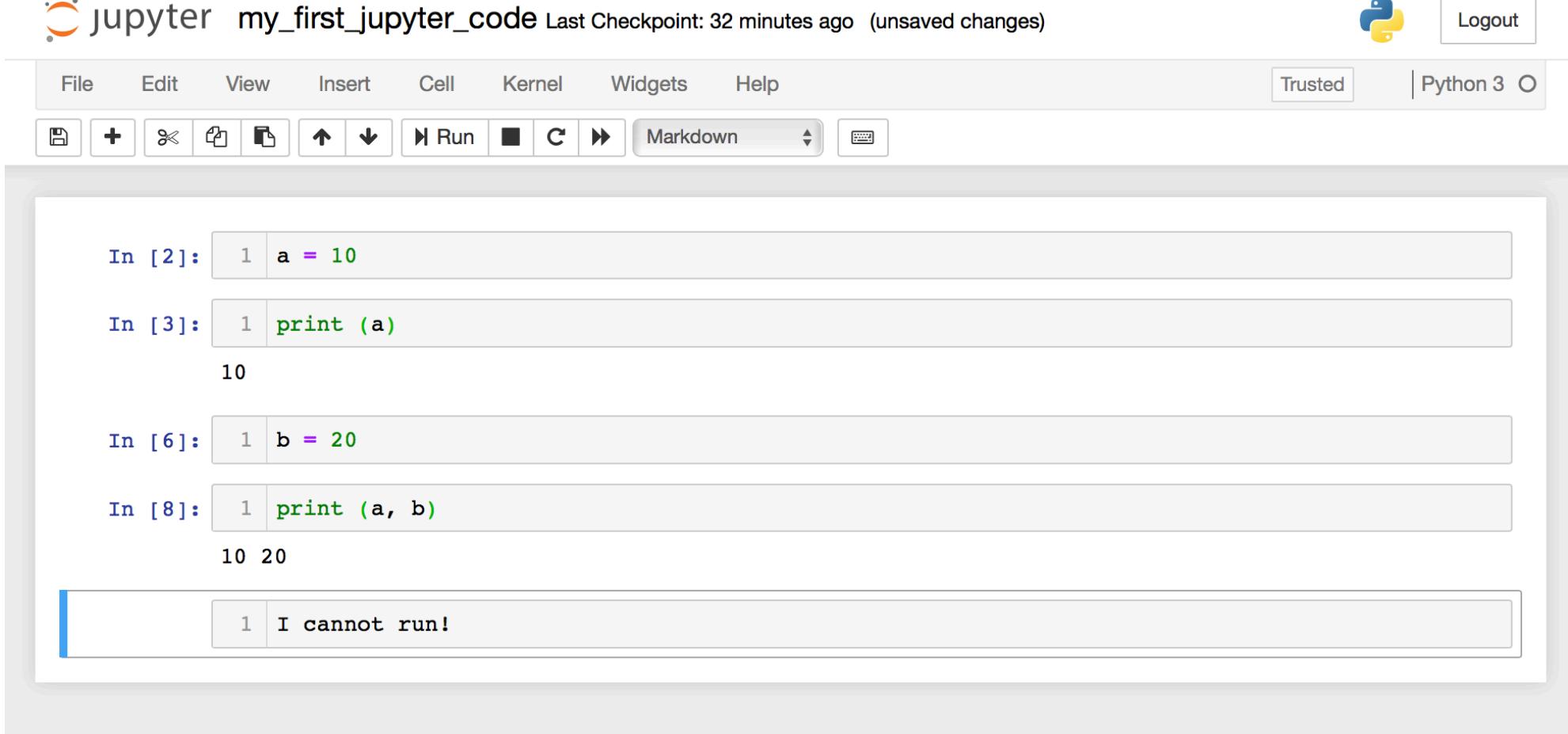


Propagations: from cell to cell

The screenshot shows a Jupyter Notebook interface with the following details:

- Header:** jupyter my_first_jupyter_code Last Checkpoint: 28 minutes ago (autosaved) Python 3
- Toolbar:** File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Notebook saved, Trusted.
- Cells:**
 - In [2]: 1 a = 10
 - In [3]: 1 print (a)
10
 - In [6]: 1 b = 20
 - In [7]: 1 print (a, b)
10 20

Add text to your notebook



The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** jupyter my_first_jupyter_code Last Checkpoint: 32 minutes ago (unsaved changes)
- User Icons:** Python logo and Logout button.
- Toolbar:** File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Trusted, Python 3.
- Tool Buttons:** File, New, Cell, Run, Kernel, Help, Markdown, Keyboard.
- Code Cells:**
 - In [2]: `1 a = 10`
 - In [3]: `1 print (a)`
10
 - In [6]: `1 b = 20`
 - In [8]: `1 print (a, b)`
10 20
 - In []: `1 I cannot run!`

Add text to your notebook

The screenshot shows a Jupyter Notebook interface with the following details:

- Title Bar:** jupyter my_first_jupyter_code Last Checkpoint: 33 minutes ago (unsaved changes)
- Toolbar:** File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Trusted, Python 3
- Code Cells:**
 - In [2]: `1 a = 10`
 - In [3]: `1 print (a)`
10
 - In [6]: `1 b = 20`
 - In [8]: `1 print (a, b)`
10 20
- Text Cell:** I cannot run!
- Input Cell:** In []: 1

Course Repository



Clone our Class Repository

- Using either the GitHub Desktop app or the git CLI, clone the class repo:

<https://github.com/CISC879-BigData/2021S-COSC526>



Clone the course repository

<https://github.com/CISC879-BigData/courses-UTK-COSC526-S21>

The screenshot shows the GitHub repository page for 'CISC879-BigData / courses-UTK-COSC526-S20'. The repository is private, has 5 issues, 0 pull requests, 0 actions, 0 projects, 0 wiki pages, 0 security vulnerabilities, 0 insights, and 1 contributor. It contains 3 commits, 1 branch, 0 packages, 0 releases, and 1 contributor. The latest commit was made 26 seconds ago by user 'taufert' and added an 'Assignment01' folder. Other files listed include 'Assignment01', 'Lecture01', '.gitignore_global', 'ReadMe.md', and 'syllabus-UTK-COSC526-S20.docx'. The 'ReadMe.md' file is open, showing the text 'Course COSC 526, Spring 2020'.

COURSE REPOS SPRING 2020 SEMESTER - COSC 526 COURSE - MATERIAL DISTRIBUTED TO STUDENTS ATTENDING THE COURSE

Manage topics

3 commits 1 branch 0 packages 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

taufert Add Assignment01 folder Latest commit 967555c 26 seconds ago

Assignment01 Add Assignment01 folder 27 seconds ago

Lecture01 Add Lecture01 folder 2 minutes ago

.gitignore_global Add files 1 hour ago

ReadMe.md Add files 1 hour ago

syllabus-UTK-COSC526-S20.docx Add files 1 hour ago

ReadMe.md

Course COSC 526, Spring 2020



Clone the course repository

The screenshot shows a GitHub repository page for the private repository `CISC879-BigData / courses-UTK-COSC526-S20`. The repository has 5 watchers, 0 stars, and 0 forks. The main navigation bar includes links for Code, Issues (0), Pull requests (0), Actions, Projects (0), Wiki, Security, Insights, and Settings.

The repository description is: "Course repos Spring 2020 semester - COSC 526 course - material distributed to students attending the course".

Key statistics shown: 3 commits, 1 branch, 0 packages, 0 releases, and 1 contributor.

The "Clone or download" button is highlighted in green. A tooltip for "Clone with HTTPS" provides the URL `https://github.com/CISC879-BigData/courses-UTK-COSC526-S20`.

The repository contents listed include:

- taufer** Add Assignment01 folder
- Assignment01** Add Assignment01 folder
- Lecture01** Add Lecture01 folder
- .gitignore_global** Add files
- ReadMe.md** Add files
- syllabus-UTK-COSC526-S20.docx** Add files

A preview of the `ReadMe.md` file content is shown:

```
Course COSC 526, Spring 2020
```



Open from CLI

```
$ git clone https://github.com/CISC879-BigData/courses-UTK-COSC526-S21.git
```



Open from GitHub Desktop

