**Git and GitHub for labs**

[Unit 1 - Introduction to Python](https://my.ironhack.com/cohorts/64f9d729a7ab01002ae1abce/lms/courses/course-v1:IRONHACK+DAFT4+202409_RMT_ES/modules/ironhack-course-chapter_1)

**Day 2**

[Daily Kick Off](https://my.ironhack.com/cohorts/64f9d729a7ab01002ae1abce/lms/courses/course-v1:IRONHACK+DAFT4+202409_RMT_ES/modules/ironhack-course-chapter_1/units/ironhack-course-chapter_1-sequential_2-vertical)

[Group Lab Review](https://my.ironhack.com/cohorts/64f9d729a7ab01002ae1abce/lms/courses/course-v1:IRONHACK+DAFT4+202409_RMT_ES/modules/ironhack-course-chapter_1/units/ironhack-course-chapter_1-sequential_2-vertical_1)

[1.02 Flow Control - Prework Review](https://my.ironhack.com/cohorts/64f9d729a7ab01002ae1abce/lms/courses/course-v1:IRONHACK+DAFT4+202409_RMT_ES/modules/ironhack-course-chapter_1/units/ironhack-course-chapter_1-sequential_2-vertical_2)

[1.03 Introduction to Functions - Prework Review](https://my.ironhack.com/cohorts/64f9d729a7ab01002ae1abce/lms/courses/course-v1:IRONHACK+DAFT4+202409_RMT_ES/modules/ironhack-course-chapter_1/units/ironhack-course-chapter_1-sequential_2-vertical_3)

**[Git and GitHub for labs](https://my.ironhack.com/cohorts/64f9d729a7ab01002ae1abce/lms/courses/course-v1:IRONHACK+DAFT4+202409_RMT_ES/modules/ironhack-course-chapter_1/units/ironhack-course-chapter_1-sequential_2-vertical_4)**

[Lab | Flow Control](https://my.ironhack.com/cohorts/64f9d729a7ab01002ae1abce/lms/courses/course-v1:IRONHACK+DAFT4+202409_RMT_ES/modules/ironhack-course-chapter_1/units/ironhack-course-chapter_1-sequential_2-vertical_5)

[Lab | Functions](https://my.ironhack.com/cohorts/64f9d729a7ab01002ae1abce/lms/courses/course-v1:IRONHACK+DAFT4+202409_RMT_ES/modules/ironhack-course-chapter_1/units/ironhack-course-chapter_1-sequential_2-vertical_6)

[Quest - day 2: Flow Control Fun and Coding Best Practices](https://my.ironhack.com/cohorts/64f9d729a7ab01002ae1abce/lms/courses/course-v1:IRONHACK+DAFT4+202409_RMT_ES/modules/ironhack-course-chapter_1/units/ironhack-course-chapter_1-sequential_2-vertical_7)

[Quest - Self Guided: Coding Best Practices](https://my.ironhack.com/cohorts/64f9d729a7ab01002ae1abce/lms/courses/course-v1:IRONHACK+DAFT4+202409_RMT_ES/modules/ironhack-course-chapter_1/units/ironhack-course-chapter_1-sequential_2-vertical_8)

[Git & Github for labs Demo](https://my.ironhack.com/cohorts/64f9d729a7ab01002ae1abce/lms/courses/course-v1:IRONHACK+DAFT4+202409_RMT_ES/modules/ironhack-course-chapter_1/units/ironhack-course-chapter_1-sequential_2-vertical_9)

[Lab | Flow Control Extended](https://my.ironhack.com/cohorts/64f9d729a7ab01002ae1abce/lms/courses/course-v1:IRONHACK+DAFT4+202409_RMT_ES/modules/ironhack-course-chapter_1/units/ironhack-course-chapter_1-sequential_2-vertical_10)

[Lab | Functions Extended](https://my.ironhack.com/cohorts/64f9d729a7ab01002ae1abce/lms/courses/course-v1:IRONHACK+DAFT4+202409_RMT_ES/modules/ironhack-course-chapter_1/units/ironhack-course-chapter_1-sequential_2-vertical_11)

**Git and GitHub for labs**

**LESSON**

Generate a quiz about this unit

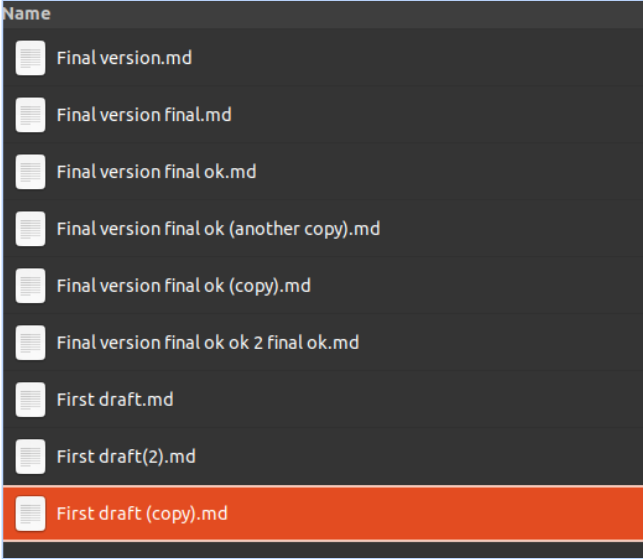
Summarize this unit

Have any doubt about this content? Ask!

**How to use Git and GitHub for the Labs/Assignments**

In this document, you will find all the steps you should follow when working on the labs.

**Why should you use version control**



* **Tracking Changes**: Maintain a detailed history of project changes, enabling easy comparison, retrieval, and understanding of project evolution over time.
* **Collaboration**: Enable seamless teamwork and efficient collaboration by allowing multiple individuals to work on the same project simultaneously, track changes, and merge modifications.
* **Error Recovery**: Serve as a safety net, enabling easy recovery from errors or mistakes by reverting to known stable states or identifying problematic changes.

**What is Git and GitHub**

**Git** is a version control system that helps you track changes in your project files over time. It allows you to save different versions of your work and easily switch between them.

**GitHub** is a website where you can store and share your Git repositories. It provides a user-friendly interface for managing your code and collaborating with others. It also allows the teaching team to see your work.

**The steps to succeed during labs**

1. **Fork**: the action of copying someone’s repo into your account
2. **Clone**: downloads a version of the project into your computer
3. **Work**
4. **Add**: Getting ready to save your changes by telling Git which files or modifications should be included in the next snapshot (commit). Not mandatory, but recommended.
5. **Commit**: creates a new snapshot of your project and saves the latest changes you added
6. **Push**: the action of uploading new changes to the project into the GitHub online repository
7. **Share**

If you work with peers, you will also use pull (the action of updating a previously downloaded repository), merge (combining changes from one branch into another, integrating the modifications into a single unified branch) and more. We will learn this later on the bootcamp.

**Detailed step by step instructions**

In the following guide, the option to use git with the terminal on your local machine and GitHub Desktop are both explained. You should follow steps 1-4, and then decide if you would like to use GitHub Desktop (instructions under the heading “Using GitHub Desktop” steps 5-13) or use the terminal (instructions under the heading “Using the terminal” steps 5-13).

The first time you work with Git, you need to **[install](https://git-scm.com/book/en/v2/Getting-Started-Installing-Git" \t "_blank)** it. Other source for instructions **[here](https://github.com/git-guides/install-git" \t "_blank)**.

1. Open the lab link from the student platform.
2. Log in to GitHub.
3. Fork the repository.

The majority of time, you will have to start the process with forking. Just a quick reminder on what is the process of forking.

Creating a “fork” is producing a personal copy of someone else’s project. Forks act as a sort of bridge between the original repository and your personal copy.

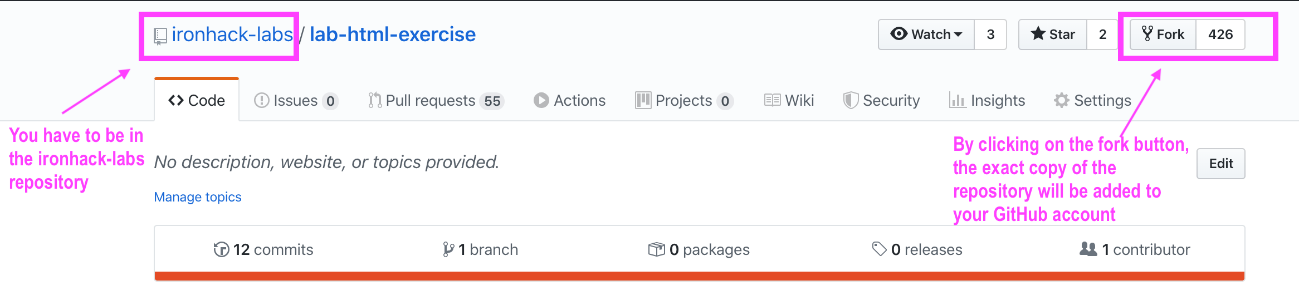
If you forked someone’s project, you will be able to push your code (this is called *making pull request*) so the owner of the original project can check it and if they find it useful, use it.

**During your Ironhack journey, you have to fork each lab to be able to submit it. This way, your TAs, and instructors will be able to check your work and give you feedback.**

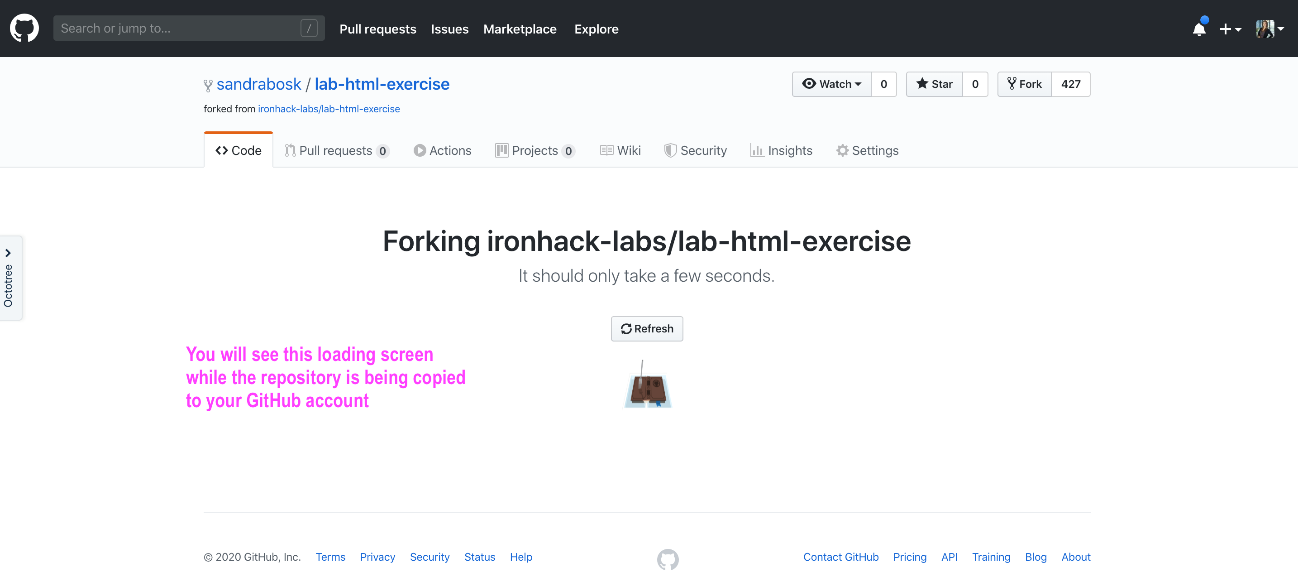
**How to fork?**

All the examples will be related to your Ironhack journey, and they will point to the **[ironhack-labs](https://github.com/ironhack-labs" \t "_blank)** GitHub organization.

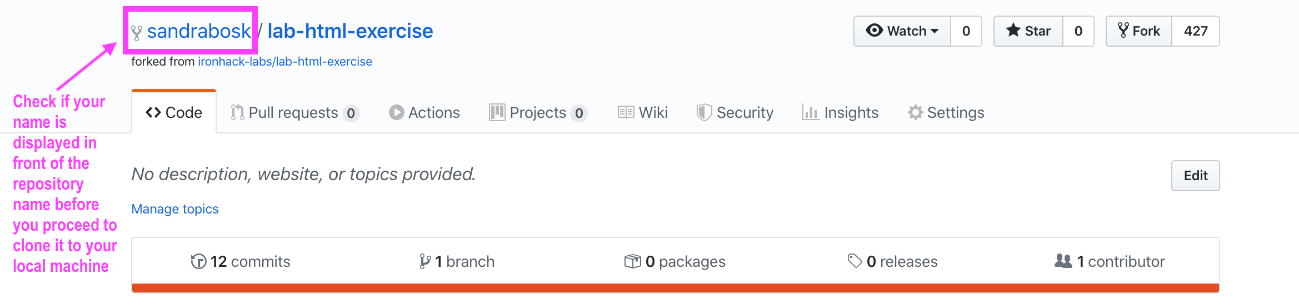
Navigate to the repository you want to fork. Let’s take an example of the prework [**HTML exercise**](https://github.com/ironhack-labs/lab-html-exercise).



After clicking on the fork button, you will be presented with the loading screen that will most likely look like this:



After the forking is done successfully, you will be redirected to the **copied repository on your GitHub account**.



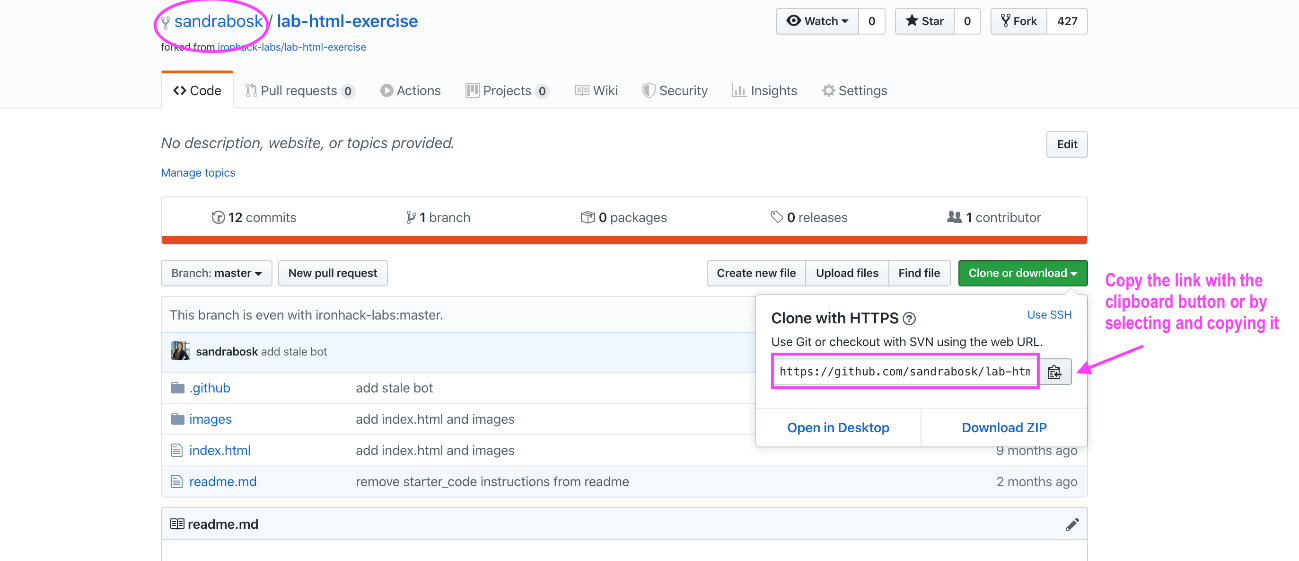
We are done here with the forking process.

1. Clone the repository.

Now, on your GitHub account, you have access to the copied repository. This will be repository you will copy (clone) to your local machine, and that will be your working area.

**How to clone repository?**

Click on the green ‘Code’ button and copy the link by selecting and copying or by using the clipboard button.



Make sure that you are in your GitHub account (on the top of the page you should see your username/name\_of\_forked\_lab).

**Using GitHub Desktop**

1. [**Install**](https://docs.github.com/en/desktop/installing-and-configuring-github-desktop/installing-and-authenticating-to-github-desktop/installing-github-desktop) and open GitHub Desktop.
2. Finish cloning the lab. Look for Clone repository in GitHub Desktop, and paste the link afterwards. For example, in Mac, you might see something like the following:
3. Using cloned repository locally. Navigate into the cloned folder, edit files or create new files as specified in the lab.
4. Uploading your changes to your GitHub account (online)
   1. Selecting the files you want to commit in GitHub Desktop.
   2. Write a commit message and click “Commit”
   3. Click on “Push”
5. Copy the link to the directory of the lab in your GitHub account and paste it into student platform.
6. Have some rest and get ready for tomorrow!

*You can repeat step 7 as many times as you want, and your changes will be updated on GitHub. This way, we will be able to see the latest version of your lab.*

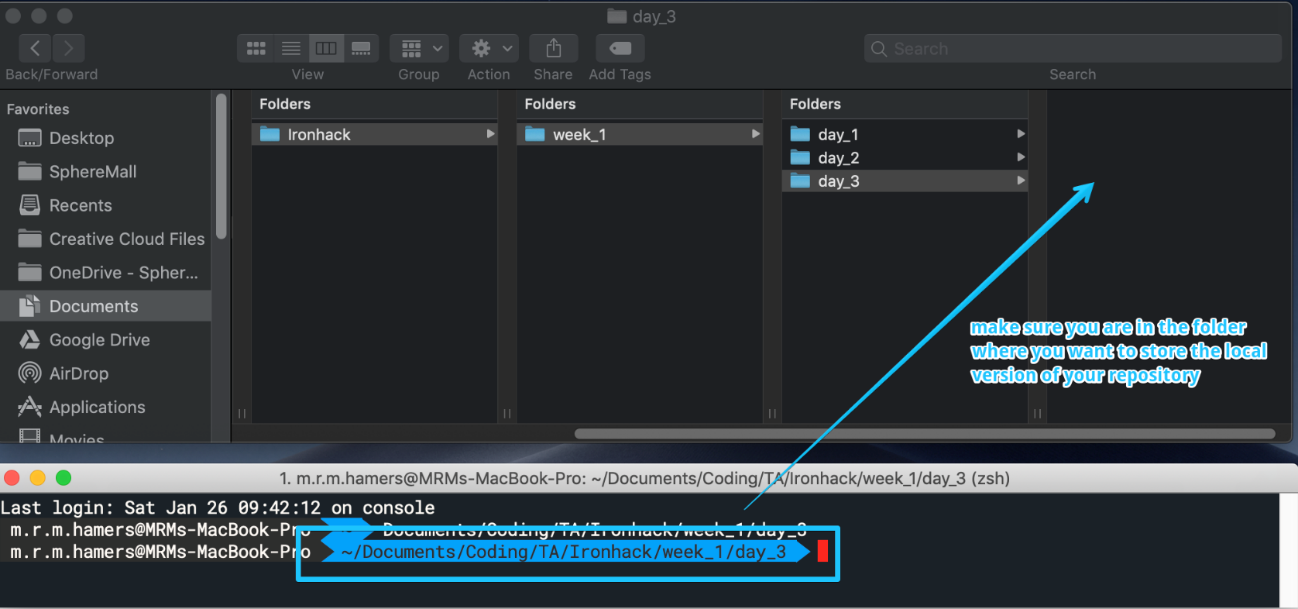
**Using the terminal**

1. Finish Cloning the lab
   * Open the folder where you want to store your lab, and open the terminal there, or open your terminal and navigate to the location where you want to store the local version of your repository.

You can navigate in the terminal to the folder where you want to store your lab, using as many times as necessary until you get to the folder you want

1. Copy

|  |  |
| --- | --- |
|  | cd name-of-next folder |

1. **Explain this code**
2. 
   * When you navigated to the correct folder in your terminal run the following command:

Copy

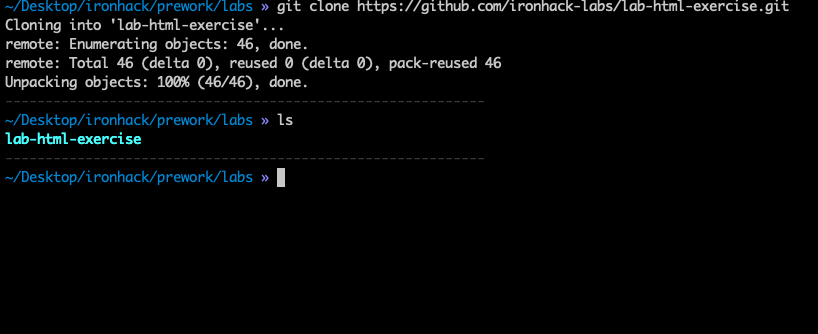
|  |  |
| --- | --- |
|  | $ git clone <paste url from the clipboard> |

**Explain this code**

Check the image below if you have any doubts:



* + Once the cloning process is done, you see the following information in your terminal. And you can check that repository was successfully cloned from GitHub to your local machine by running ls command in your terminal:



The cloning process is finished with this step. Now you can use the code to make changes before we proceed to the next step, which is pushing the changes to the forked repository on your GitHub.

1. Using cloned repository locally. Navigate into the cloned folder, edit files or create new files as specified in the lab.
2. Uploading your changes to your GitHub account (online)
   * Add the modified files with “git add .”, or instead of . write the names of the files you want to add separated by commas.

Copy

|  |  |
| --- | --- |
|  | git add . |

**Explain this code**

* + Then use “git commit -m ‘your message here’”

Copy

|  |  |
| --- | --- |
|  | git commit -m "fixed scatterplot axis" |

**Explain this code**

* + Use “git push”

Copy

|  |  |
| --- | --- |
|  | git push |
|  |  |

**Explain this code**

1. Copy the link to the directory of the lab in your GitHub account and paste it into student platform.
2. Have some rest and get ready for tomorrow!

*You can repeat step 7 as many times as you want, and your changes will be updated on GitHub. This way, we will be able to see the latest version of your lab.*

**Extra resources**

* [**GitHub Desktop Official Documentation**](https://docs.github.com/en/desktop/installing-and-configuring-github-desktop/overview/getting-started-with-github-desktop)
* [**GitHub CLI - Command Line**](https://docs.github.com/en/github-cli)