



**GITHUB**

## WHAT IS GITHUB?

- It's an **online hosting service for git repositories**
- Github repos can be:
  - Public
  - Private



## GITHUB: REASONS TO HAVE AN ACCOUNT

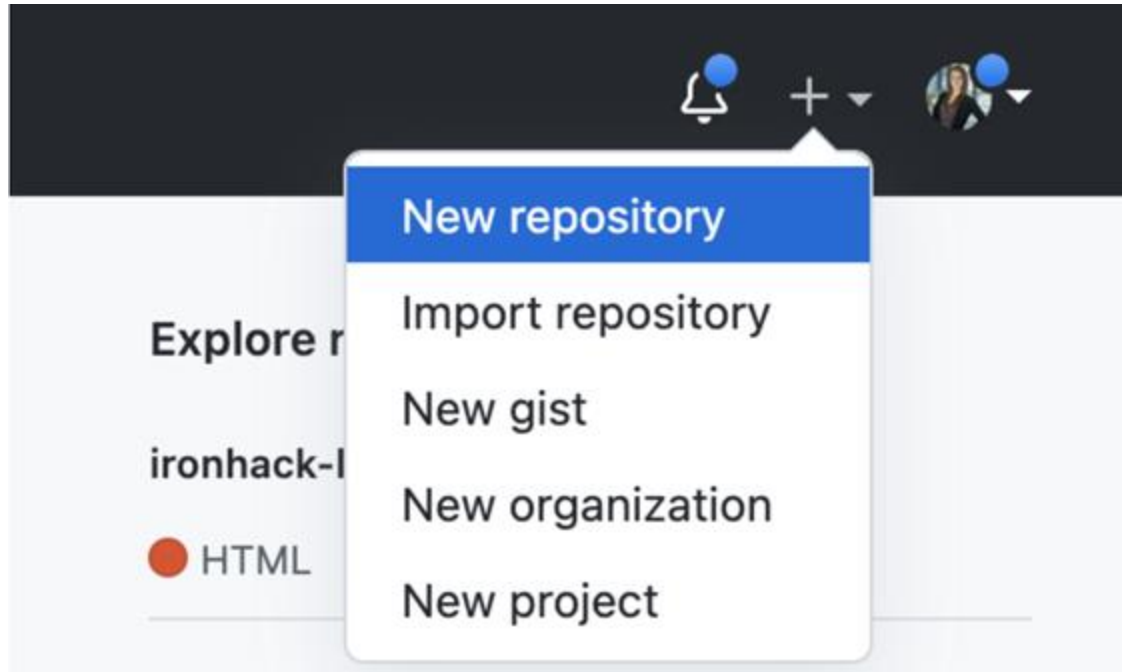
- Backup your git repositories
- Collaborate with other developers
- Create open source projects
- Expose your coding capabilities to hiring managers

## GITHUB: CREATING AN ACCOUNT

- In order to create a GitHub account, you must go to: <https://github.com/> and create an account.
- Now you can create as many “repos” on GitHub as you want. **However, they have nothing to do with your “local git repos” in your computer unless you “link” each other.**

# GITHUB: CREATING A REPOSITORY

- In order to create a GitHub repository, go to your GitHub account and, on the “+” sign click on “New repository”



# GITHUB: CREATING A REPOSITORY

- Afterwards, you will be presented with the screen shown on the right.
- Then, you will have to:
  - Give a name to the repository
  - Set the visibility
  - Leave everything else as default and click on “Create 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 \*



sandrabosk

Repository name \*

git-practice



Great repository names are short and memorable. Need inspiration? How about [furry-octo-rotary-phone?](#)

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.](#)

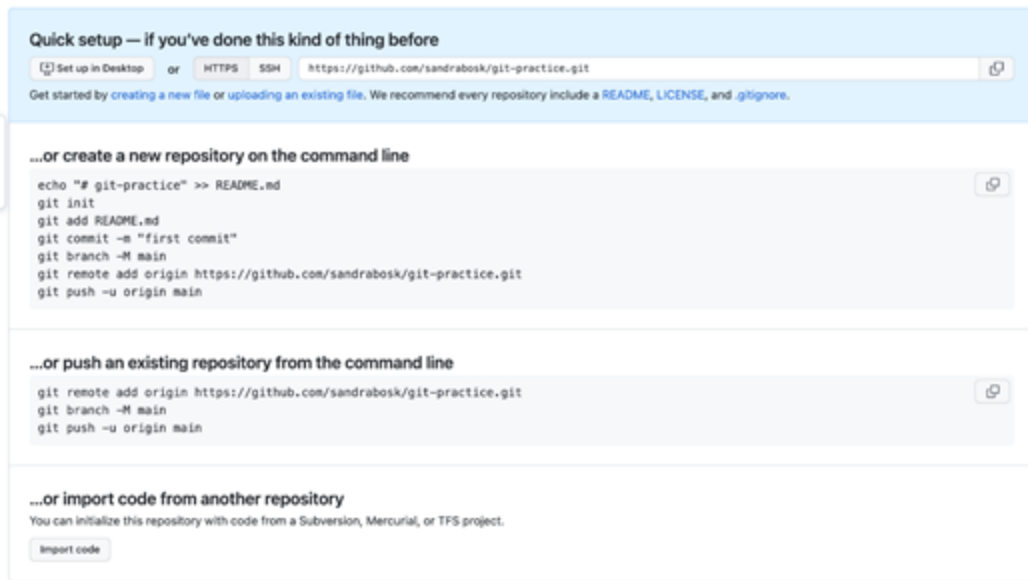
☐ Choose a license

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

Create repository

# GITHUB: CREATING A REPOSITORY

- Next, you will be presented with an screen like this one



 **ProTip!** Use the URL for this page when adding GitHub as a remote.

## LINKING A LOCAL REPO AND A GITHUB REPO

- **GitHub repos and local repos can't see each other**, unless we link them. Here is how to do it.

```
git remote -v          # to check the remote
git remote add origin <github_url>  # link is the link to the repo on GitHub
git remote -v          # to check if it is added properly
```



`git remote add origin <github_url>`



**ORIGIN**





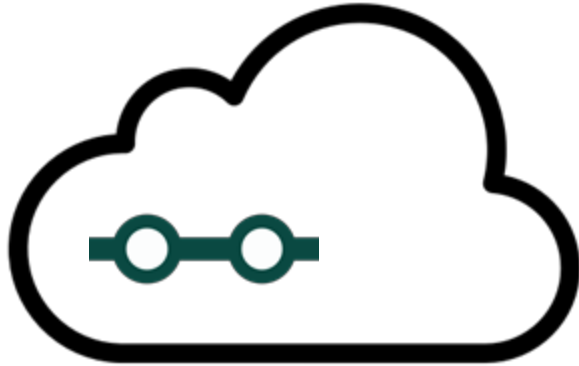
# UPLOADING COMMITS TO GITHUB REPOS: PUSH



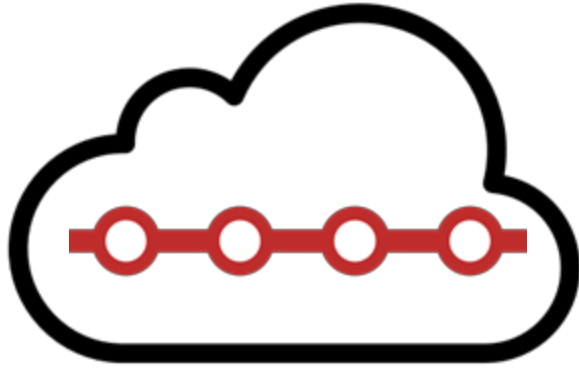
- push

## SOME CONSIDERATIONS ABOUT GITHUB AND GIT REPOS

- Even though a local git repo and a GitHub repository can be linked, the files in both repositories **are not automatically synchronized**.
- The synchronization has to be done manually using **push**



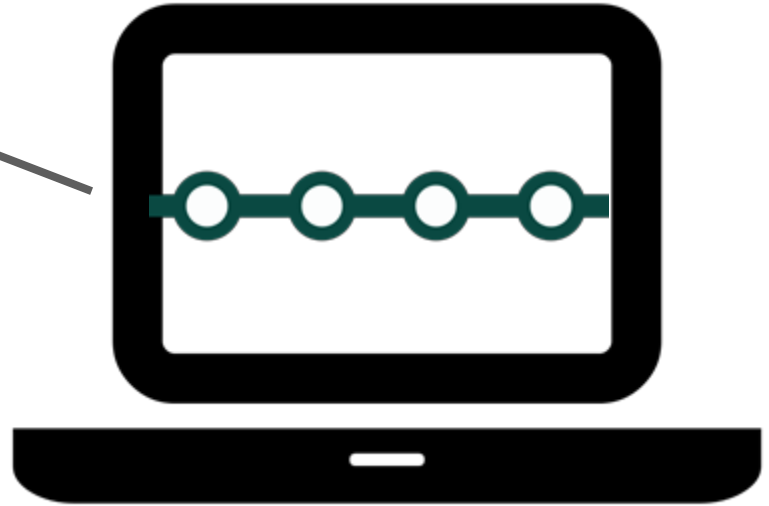
## UPLOADING LOCAL COMMITS TO GITHUB: PUSH



**git push origin master/main**

**We must be the owners of the GitHub repo to be allowed to make a “push”.**

We can push any local branch as we want as long as we are the owners of the GitHub repo.



## PUSHING COMMITS

Caveat!!!

The first time that you do a “push”, the computer will ask for your GitHub username and the password.

**However, when it says “password” it means the Personal Access Token (PAT) that you generated during the prework!!!!**

**(for security reasons you will not see the Personal Access Token or anything when you type it for security reasons).  
Copy and paste it on the terminal and press intro.**

# DOWNLOADING COMMITTS FROM OUR GITHUB REPO: PULL



- pull :

## GIT PULL COMMAND

- The command to download commits from a GitHub repository is: **pull**.
- This command downloads any additional remote changes to our remote repository and integrates the changes in our folder.
- **The changes ARE applied onto our local files.**