

{ POWER.CODERS }

Intro to GitHub

CONTENTS

- Git and GitHub: What's the difference?
- Create a Github account
- Understanding Repositories
- Discussions & Code Review on Github
- Tips and Resources

Git and GitHub: What's the difference?

GIT AND GITHUB: WHAT'S THE DIFFERENCE?

Git is a version control system software that you can run on your own computer, or on a server.

GitHub is a cloud-based service for storing, sharing, and collaborating on projects managed with Git.

TEAM COLLABORATION

GitHub is the server, where your team projects are hosted.

Git allows you to communicate with Github and push/pull your files.

ALTERNATIVES

We will use GitHub, but be aware that there are similar services, like GitLab or BitBucket.

[Read more on that](#)

Create a Github account

CREATE A GITHUB ACCOUNT

Already have a Github account? Go to the next slide.

1. Go to <https://www.github.com/join>
2. Find yourself a username. It does not have to be your name.
Mine - as an example - is **sardaykin**.
3. Use your powercoders email address.
4. Choose the free, public plan.
5. Verify your email address.

SET UP YOUR PROFILE

If you already have a Github account, make sure your **powercoders.org** email address is your **new primary address**.

Fill out your biography. Make sure to mention that you are a student or participant of the PowerCoders program.

Your GitHub Profile is like a social media profile for your code. All interested companies will look at your profile, projects and code quality.

GITHUB STUDENT DEVELOPER PACK

We will prepare for that

Github offers free additional benefits for schools and education programs, including partner offers.

Apply now

- Make sure you have your bio filled out before applying
- Make sure you have your account authenticated and 2FA enabled

SSH

- SSH is a protocol for secure communication between computers
- We'll use SSH to communicate between our computers and github servers
- To establish a secure connection we need to generate a public-private key pair.
- ([Learn more about SSH here](#))

CREATING AN SSH KEY FOR YOUR COMPUTER

- On the command line: `ssh-keygen`
- When asked for the path, just hit enter to save at the default location.
- When asked for the passphrase, leave empty.
- This generates two files, one ending in `.pub`. This is the *public key* that you can share with others.

REGISTERING YOUR COMPUTER'S PUBLIC KEY ON GITHUB

1. Run `cat ~/.ssh/id_rsa.pub` to display the public key.

Example:

```
seb@seb-mbp:~% cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDOM4TM74JdxVHSAg0A4bWaqhxTVM80HvKw6n3iF1MNT8LX0W4osjTZ6TLhuTaRG
lC8iDXG5BNdDh4uvkts41v/QBQZZA89MZPpuM4yRs8pu/QA7sg8vw+MsMHyJvElSMG0R/kErwhRAmuMPtcVnLhRjo5n06XHsDK0MM
cDCypd9UgTERJifS+/CmcXh24LsQPVMU2qjq@weW2EToYyef/LFUtoVTS4F2I5Qi68p3lW+G0/W3r2ETjNGkietINej4/HE3h44Gq
XeCS2yivvkAxYoUhy1NKqzpQMhfiIWPpnyKai3sVoLEMMmpeZ7hmHdn75LFZxNYyVlqd5T7z4IcTt seb@Windows-Phone
seb@seb-mbp:~%
```

2. Go to your [GitHub account settings](#) > SSH keys > add new
3. Name your key to remember on which computer it is, and paste the key in the appropriate form field.
4. Voilà, you'll be able to clone/push to GitHub from this computer for authorized repositories!

Understanding Repositories

CREATE A REPOSITORY AT GITHUB

- > <https://help.github.com/articles/create-a-repo/>
- > Create a public repository, called `poco-yourname`
- > Description: "Powercoders Switzerland: YOUR NAME"
- > Do **not** initialize with a `README` or `LICENSE`

CONFIGURE LOCAL REPO TO SYNC

- > Other copies of your repository are called **remotes**
- > Find your GitHub repository URL, it looks like

```
git@github.com:username/poco-yourname.git
```

- > Open terminal in VSC and enter

```
$ git remote add origin  
git@github.com/user/poco-yourname.git
```

- > \$ `git remote -v` to view configured remotes

PUSH COMMITS TO GITHUB

```
> git push -u origin main - first time
```

```
> git push - after that
```

The very first time we push a commit we have to tell Git exactly how these two repositories (local and GitHub) are linked, including how they branch.

The `-u` flag is used to set origin as the **upstream** remote in your git config. You only have to do that the first time and then can use `push` and `pull` without arguments.

You will probably be prompted for credentials or an authentication token.

NEW WORKFLOW

- > `git add [...]` to add changes
- > `git commit [...]` to commit the changes locally
- > `git push` to send the changes to GitHub

We had `add` and `commit` before, new is the command `push`

WORKING IN A TEAM

- > Make sure you have the latest changes to avoid merge conflicts
- > `git pull origin main` before you start working on a repo that day

GITHUB FROM NOW ON

- > Commit all your work to your Git repository
- > `git push` regularly
- > Expect comments! Programmers **communicate**

Dicussions & Code Review on Github

COMMENTS ON GITHUB

Anyone can see the comments and comment on public repositories.

Check it out yourself and go to the **conversation** tab.

Comment on your buddy's repository. Request a change for example.

PULL REQUESTS

Github has a mechanism that allows you to contribute to someone else's repo, called "pull requests".

You do your change on a personal branch, then submit a pull request to the repo owner.

The owner (or reviewer) can then accept or dismiss the pull request, or ask for changes.

When working in a small team you do not *have* to use pull requests, it's mostly used for outside contributions or big teams.

ACCEPTING FEEDBACK

It's totally normal for a pull request to require changes.

Don't be afraid to ask questions for clarification.

Most maintainers are friendly and helpful. If you encounter one who isn't, consider finding a different project worthy of your contributions.

WE'VE GOT ISSUES

GitHub **issues** are where collaborators track bugs, propose features, and discuss changes. They are for conversation, and do not actually change code.

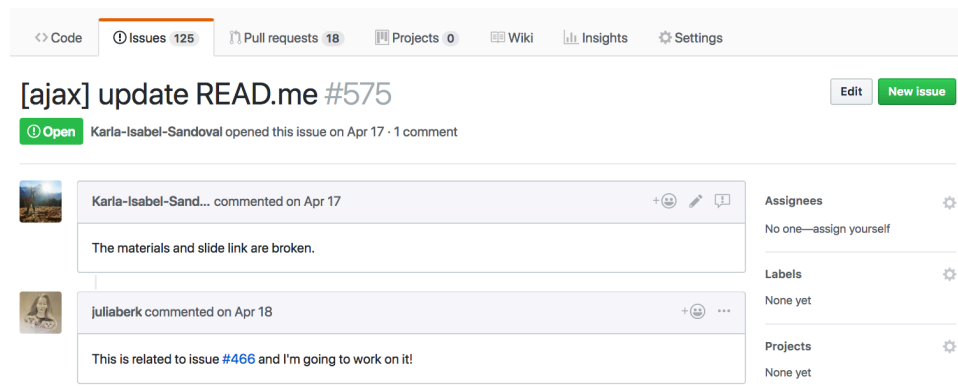
Anyone can open or comment on an issue.

Try it yourself. Propose a feature to other participants.

CLAIMING ISSUES

On Open Source projects anyone can claim issue to solve them for the community.

If you find an issue you'd like to work on, avoid duplicating work by commenting to say you'll volunteer.



The screenshot shows a GitHub issue page for the repository '[ajax] update READ.me #575'. The issue is currently open and was created by Karla-Isabel-Sandoval on April 17. The issue title is '[ajax] update READ.me #575'. The issue description is 'The materials and slide link are broken.' There are two comments: one from Karla-Isabel-Sandoval on April 17 and one from julliberk on April 18. The comment from julliberk says 'This is related to issue #466 and I'm going to work on it!'. The issue has no assignees, labels, or projects.

Feel free to ask questions if you need clarification before starting!

Tips and Resources

TIPS AND RESOURCES

- [GitHub Guides](#) (by GitHub)
- [Youtube Playlist for GitHub beginners](#)
- [Kent C. Dodds's GitHub video series](#)
- [Shubeksha Jalan's insightful story](#) of her journey into open source
- [A list of awesome beginners-friendly projects](#). Check it out.

