

Git, GitHub and Markdown

<https://github.com/heig-vd-dai-course>

[Web](#) • [PDF](#)

L. Delafontaine and H. Louis, with the help of GitHub Copilot.

This work is licensed under the [CC BY-SA 4.0](#) license.

Objectives

- Understand Git, GitHub, and Markdown
- Develop crucial collaboration skills
- Emphasize the significance of documentation for code usability and project visibility



Git

More details for this section in the [course material](#). You can find other resources and alternatives as well.

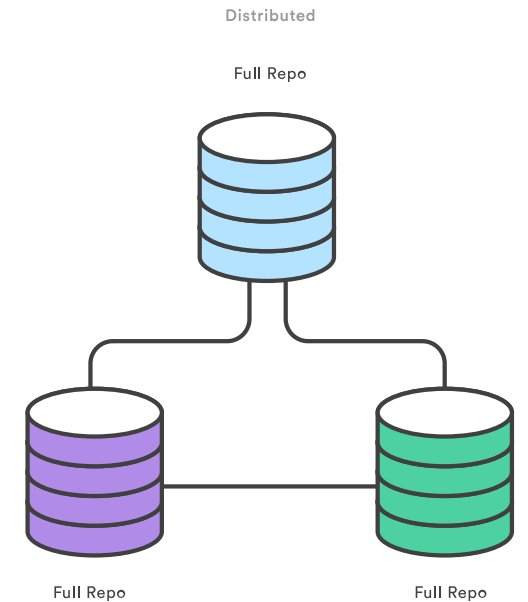
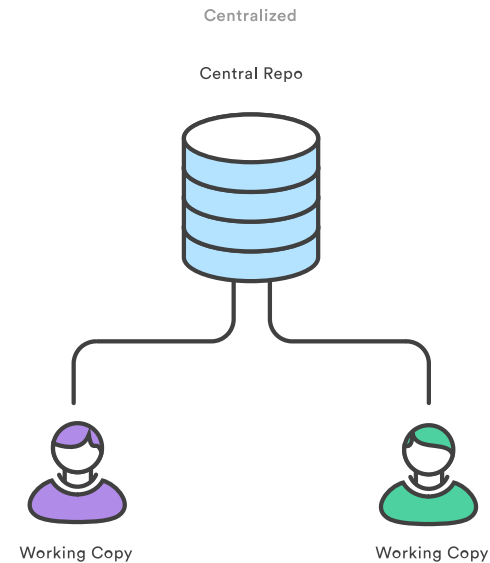
Git

- A version control system (VCS)
- Created in 2005 by Linus Torvalds
- Efficiently tracks changes in files
- Used to manage source code and documentation within a project



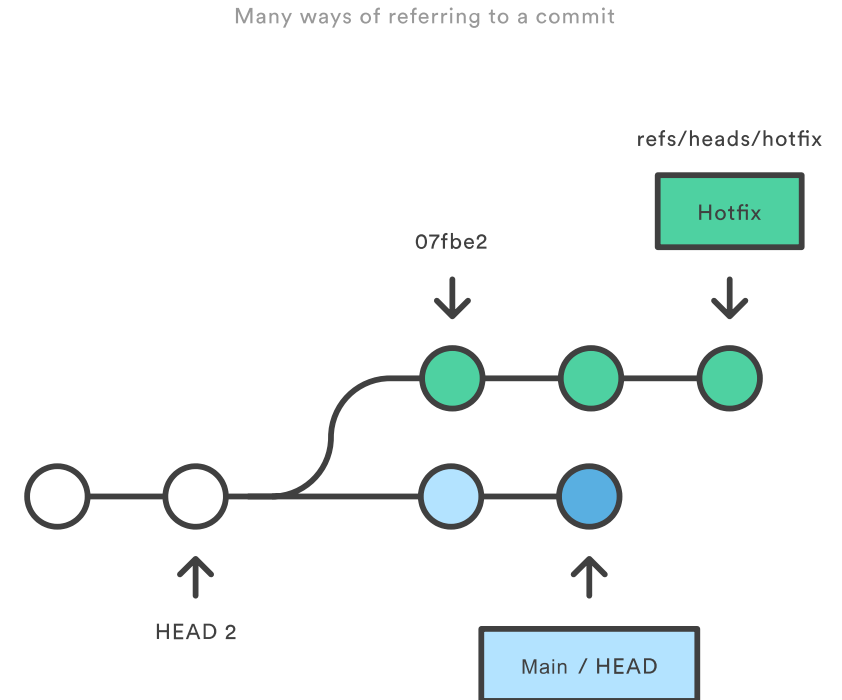
Architecture

- Git operates with a client-server structure:
 - Remote repositories (server)
 - Local repositories (local copies known as *clones*)
- Fully distributed VCS for offline work: each clone contains the full history



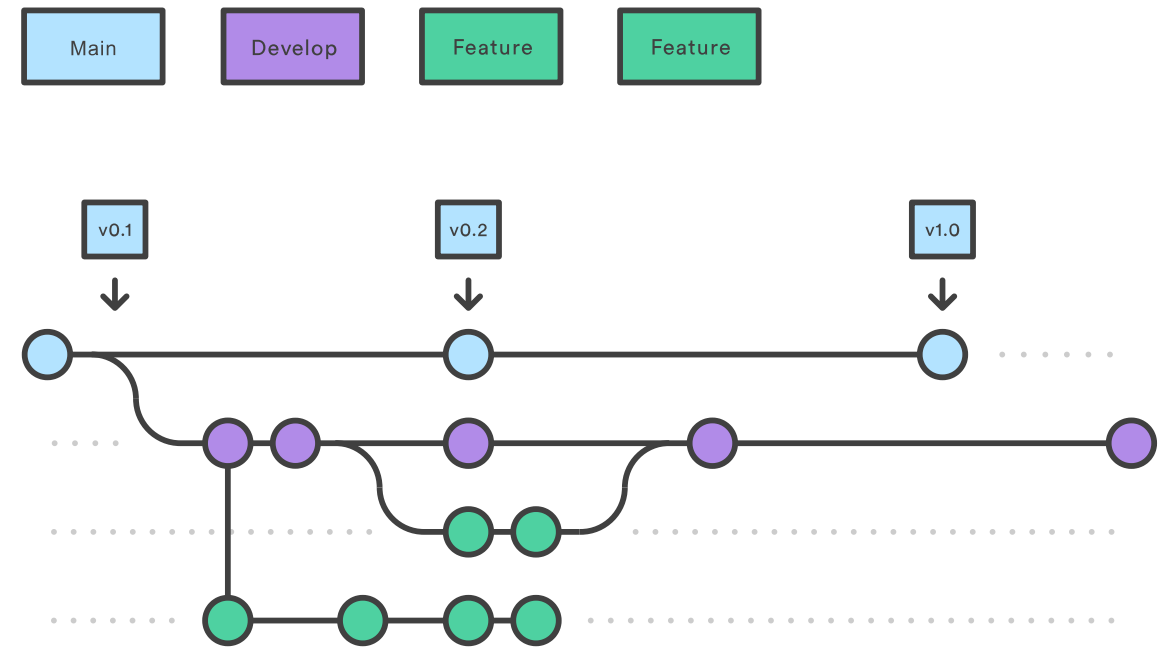
Commits, hashes and tags

- Commits represent snapshots of the repository
- Each commit has a unique, unalterable hash
- Tags serve as references, often for releases
- Commit signing enhances security



Branches

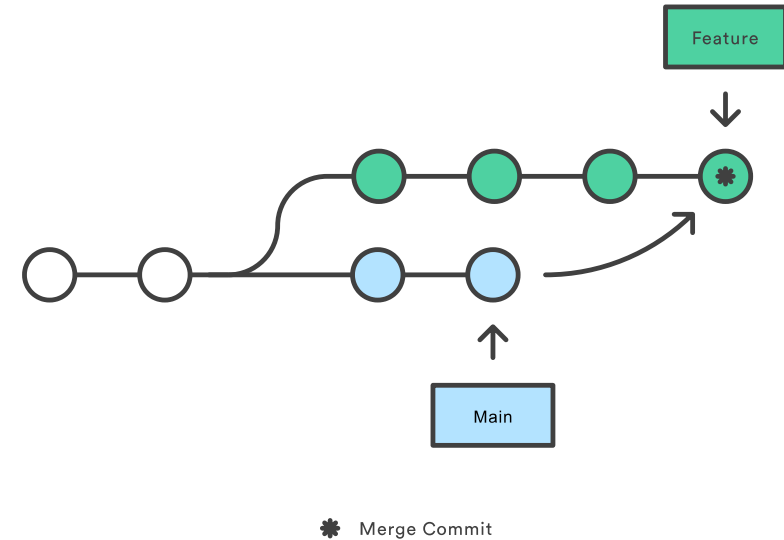
- Tracks distinct repository versions
- Default branch is typically named `main`
- Branches have names and commit pointers (heads)
- Allow to work on features or bug fixes without affecting the main branch



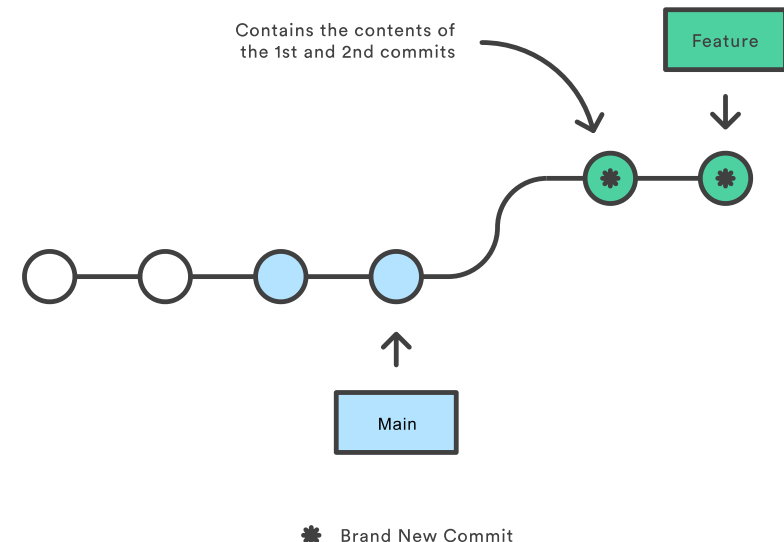
Merging branches

- Collaborative project workflow
- Three primary merging methods:
 - Merge
 - Rebase
 - Squash
- In this course, you will focus on the merge method

Merging main into the feature branch



Squashing a commit with an interactive rebase



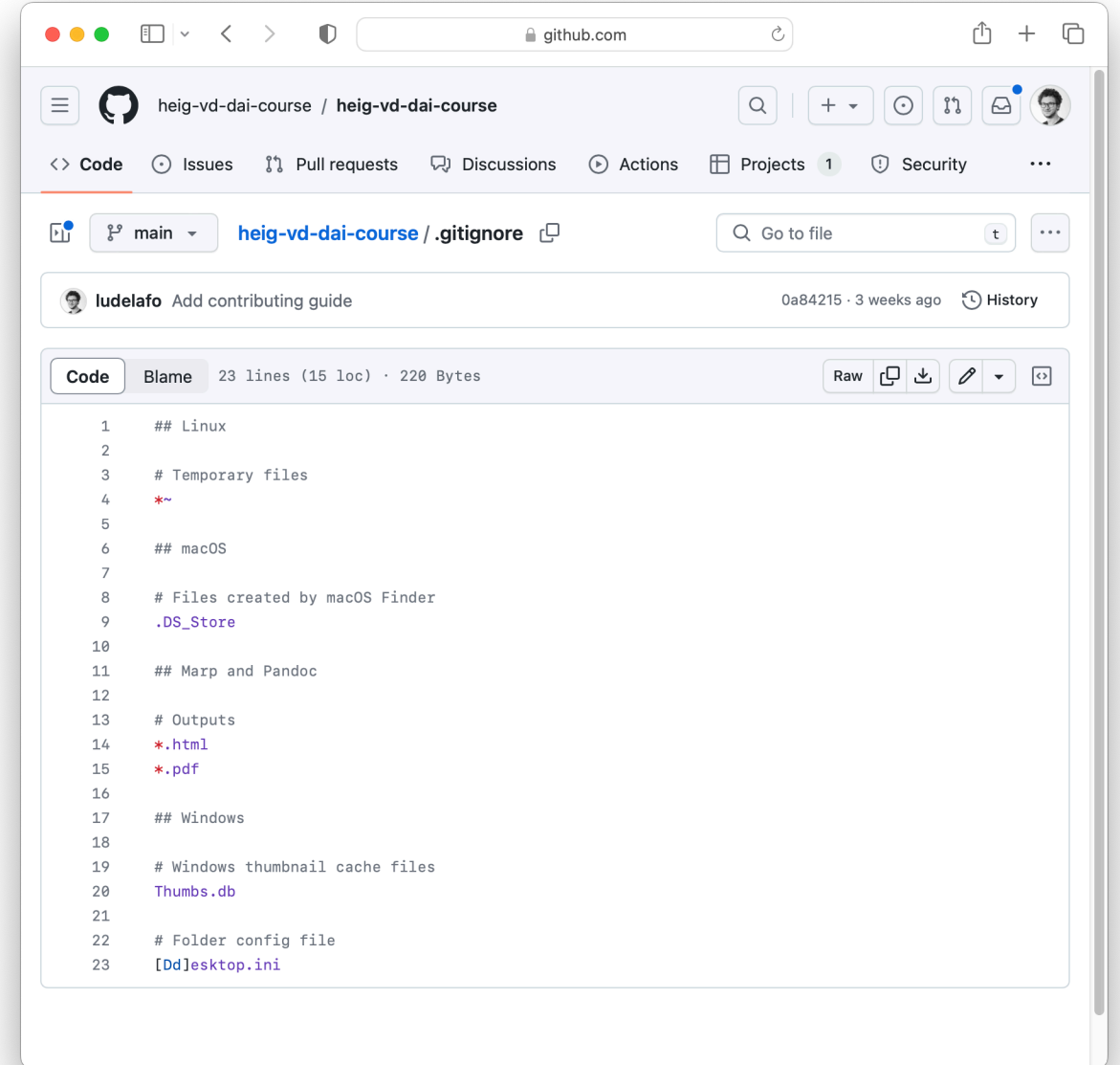
Conflicts

- Arise from simultaneous file edits
- Git detects and requires resolution
- Conflicts must be resolved to decide which changes to keep
- It can be tricky but is a crucial skill



Ignore Files

- gitignore files omit specific files that should not be on the repository
- Improves repository cleanliness and security (avoid sensitive data leaks)
- Tip: avoid online gitignore generators, they generate a lot of unnecessary rules



The screenshot shows a web browser displaying the GitHub repository page for 'heig-vd-dai-course / heig-vd-dai-course'. The file '.gitignore' is selected, showing its content. The file is 23 lines long (15 loc) and 220 bytes. The content of the .gitignore file is as follows:

```
1  ## Linux
2
3  # Temporary files
4  *~
5
6  ## macOS
7
8  # Files created by macOS Finder
9  .DS_Store
10
11 ## Marp and Pandoc
12
13 # Outputs
14 *.html
15 *.pdf
16
17 ## Windows
18
19 # Windows thumbnail cache files
20 Thumbs.db
21
22 # Folder config file
23 [Dd]esktop.ini
```

Summary

- Git is a distributed VCS
- Git uses commits to track changes
- Git uses branches to track different versions of the repository
- Git can pull changes from a remote repository
- Git can merge, rebase or squash branches/commits
- Git can detect and resolve conflicts

We encourage you to learn how to use Git from the command line, so you can still use it everywhere (even on a server).

GitHub

More details for this section in the [course material](#). You can find other resources and alternatives as well.

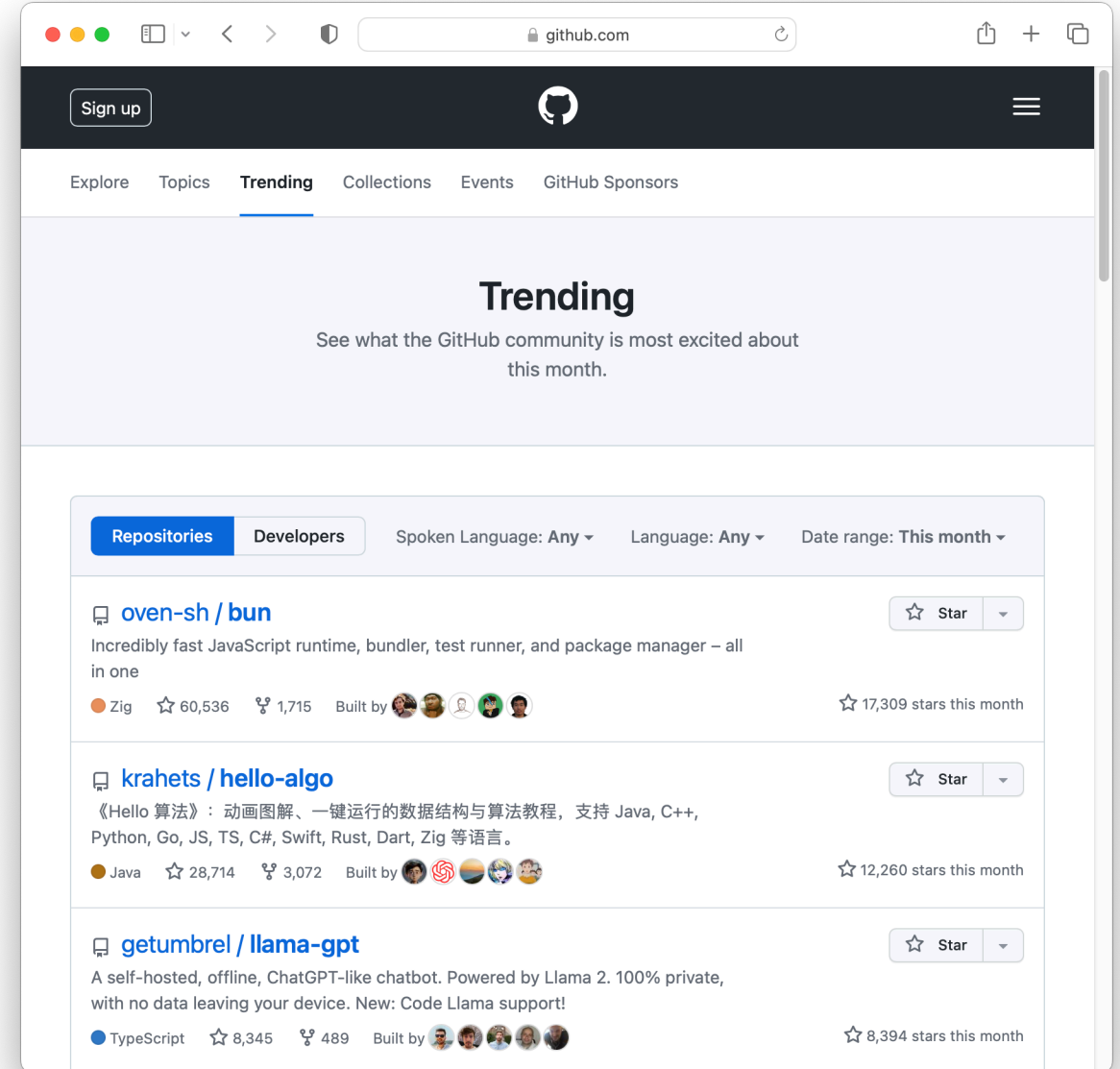
GitHub

- Git hosting platform
- Created in 2008
- Web-based Git hosting
- Popular for open source projects



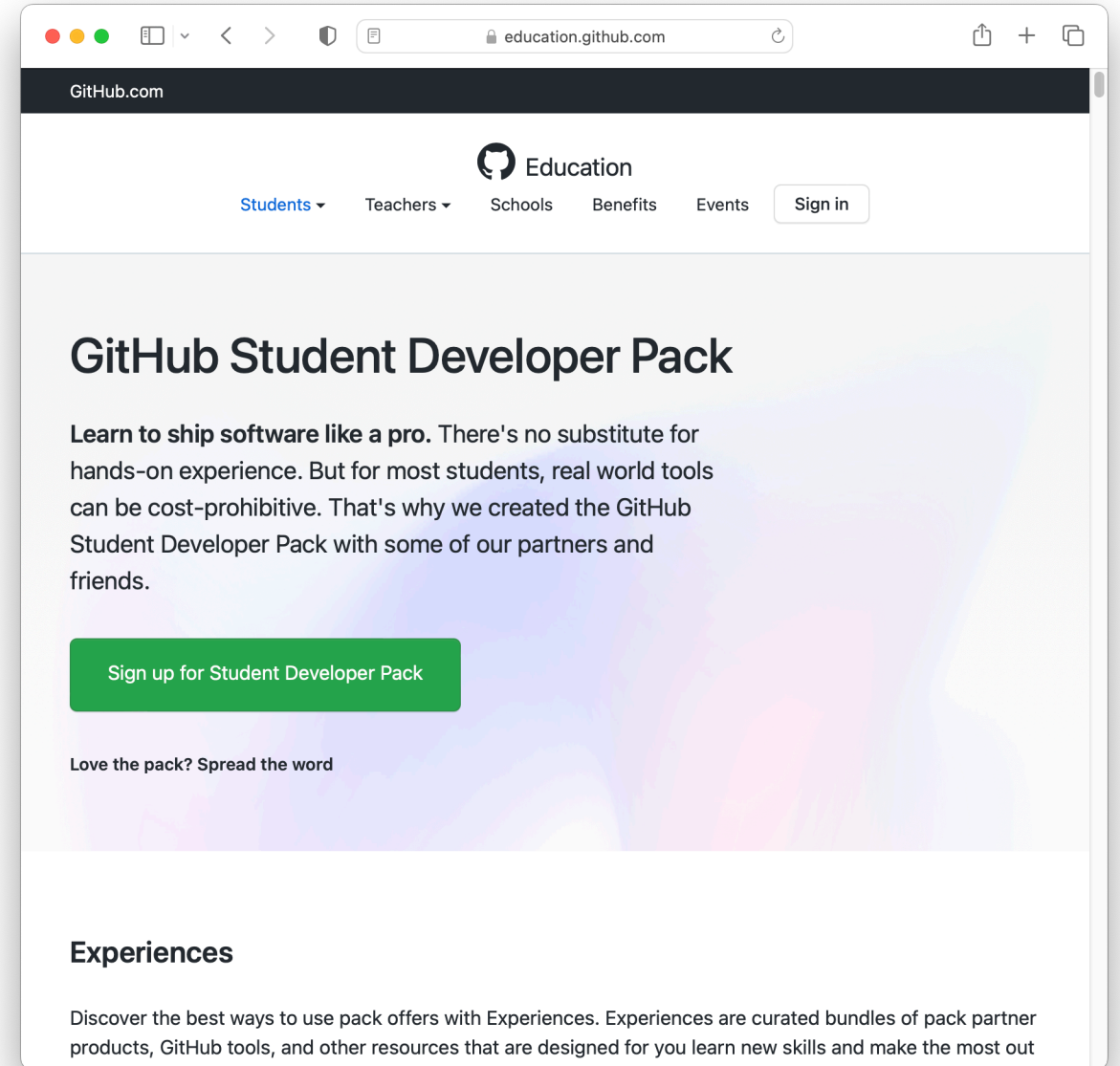
Social platform

- Social interactions:
 - Follow users
 - Star projects
 - Discover new projects
- Forking repositories
 - Create a copy of a repository
 - Make changes and contribute back



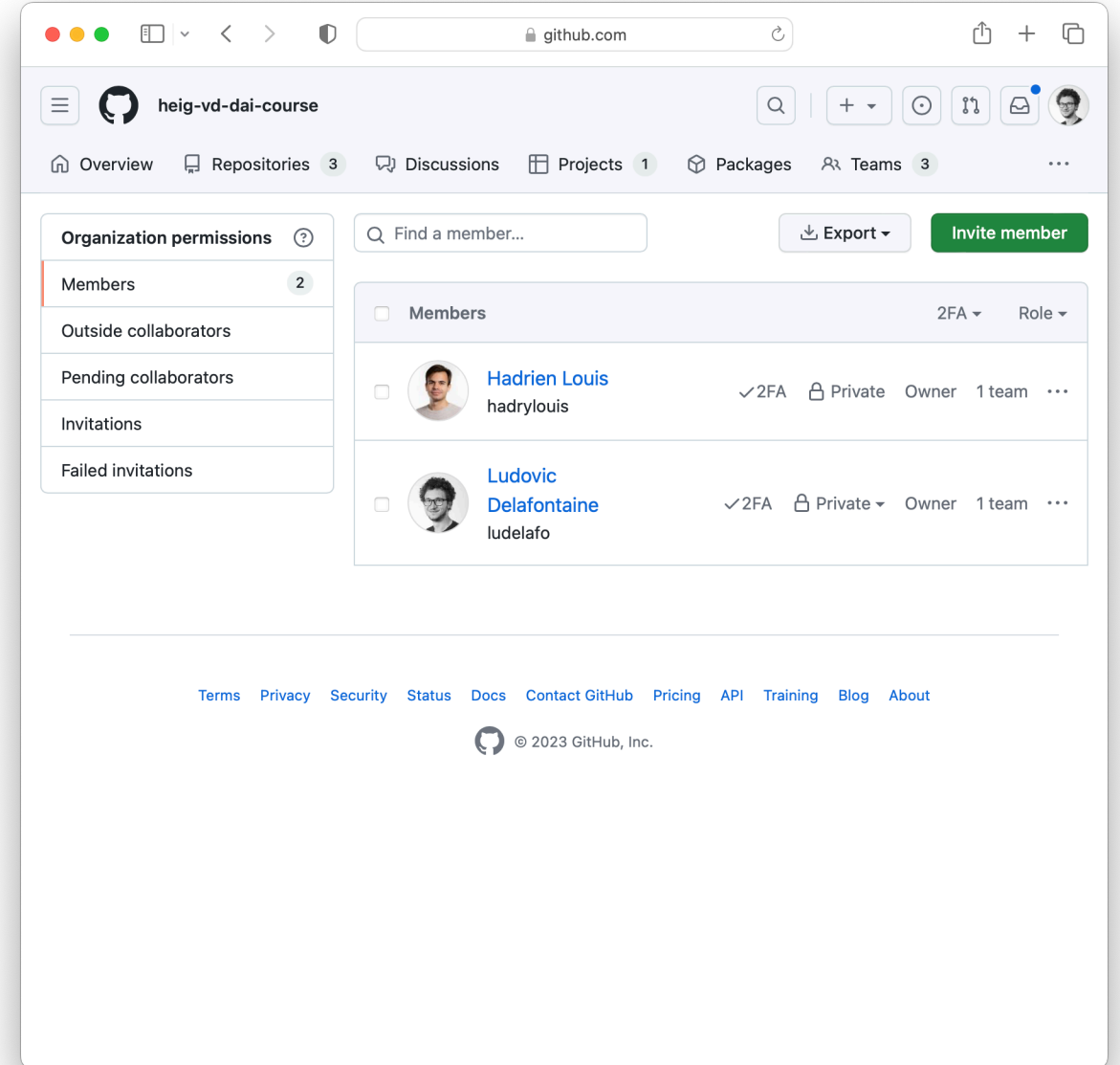
Free features and PRO features

- Enhanced features with PRO
 - Access to premium features and software
- Free for students - you can apply!



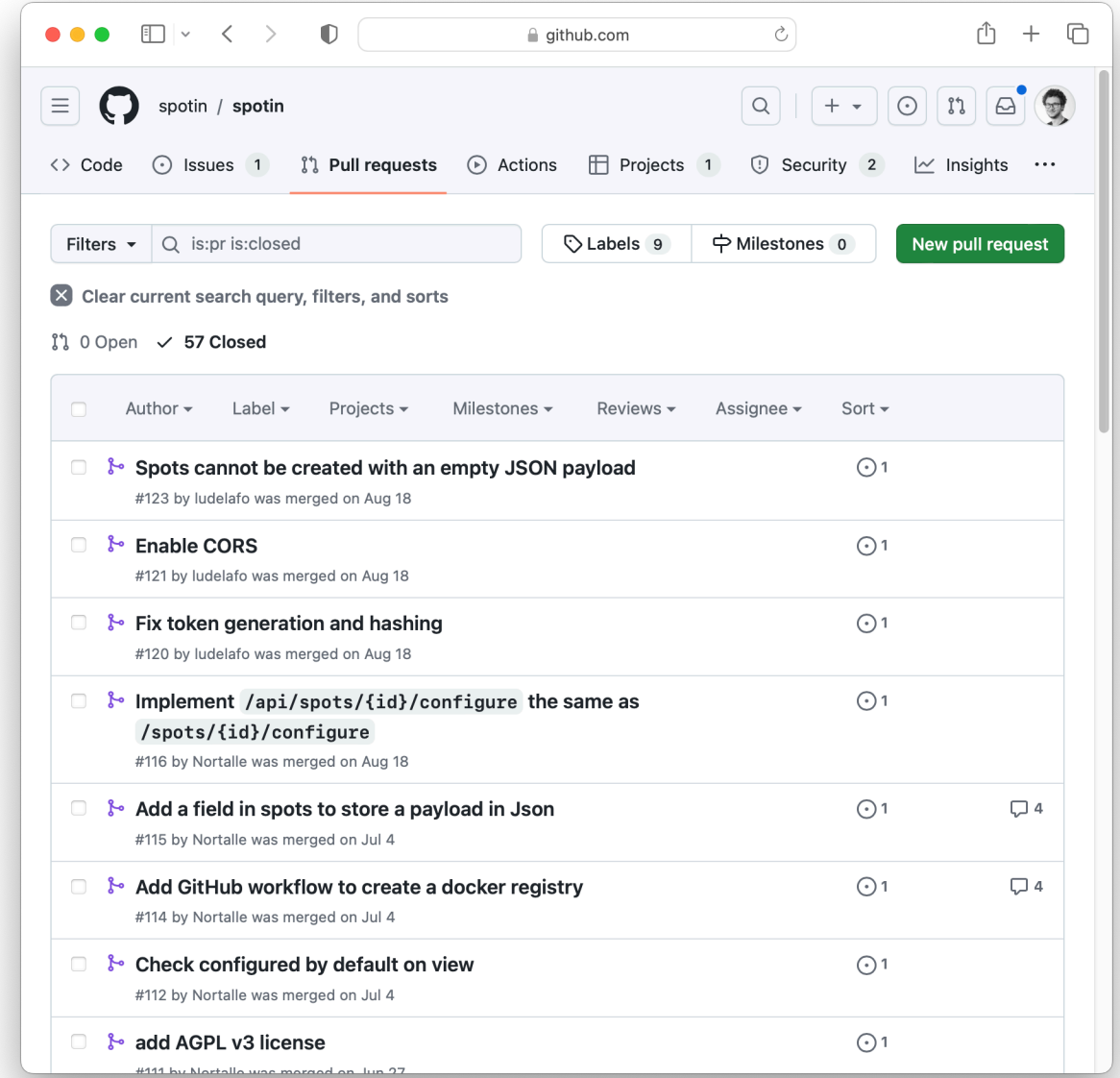
Users and organizations

- Users or organizations ownership
- Users can be part of multiple organizations



Issues, pull requests and forks

- Contribution workflow
 - Issues
 - Forks
 - Pull requests
- Approval process
- Project management tools (wiki, planning, etc.)



Summary

The whole process would be as follow:

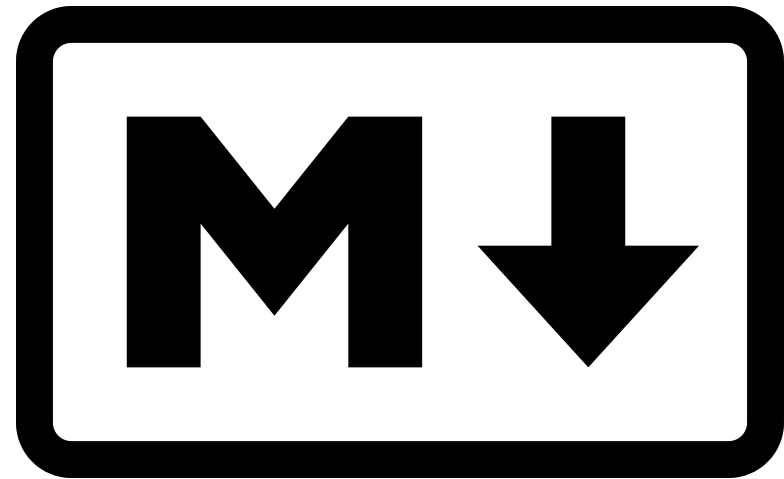
1. Open an issue to discuss the feature (optional, but recommended)
2. Clone or fork the project and checkout to a new branch
3. Make your changes, commit and push them as often as you want
4. Create the pull request
5. The maintainers review and merge if OK
6. You can delete the branch or the fork

Markdown

More details for this section in the [course material](#). You can find other resources and alternatives as well.

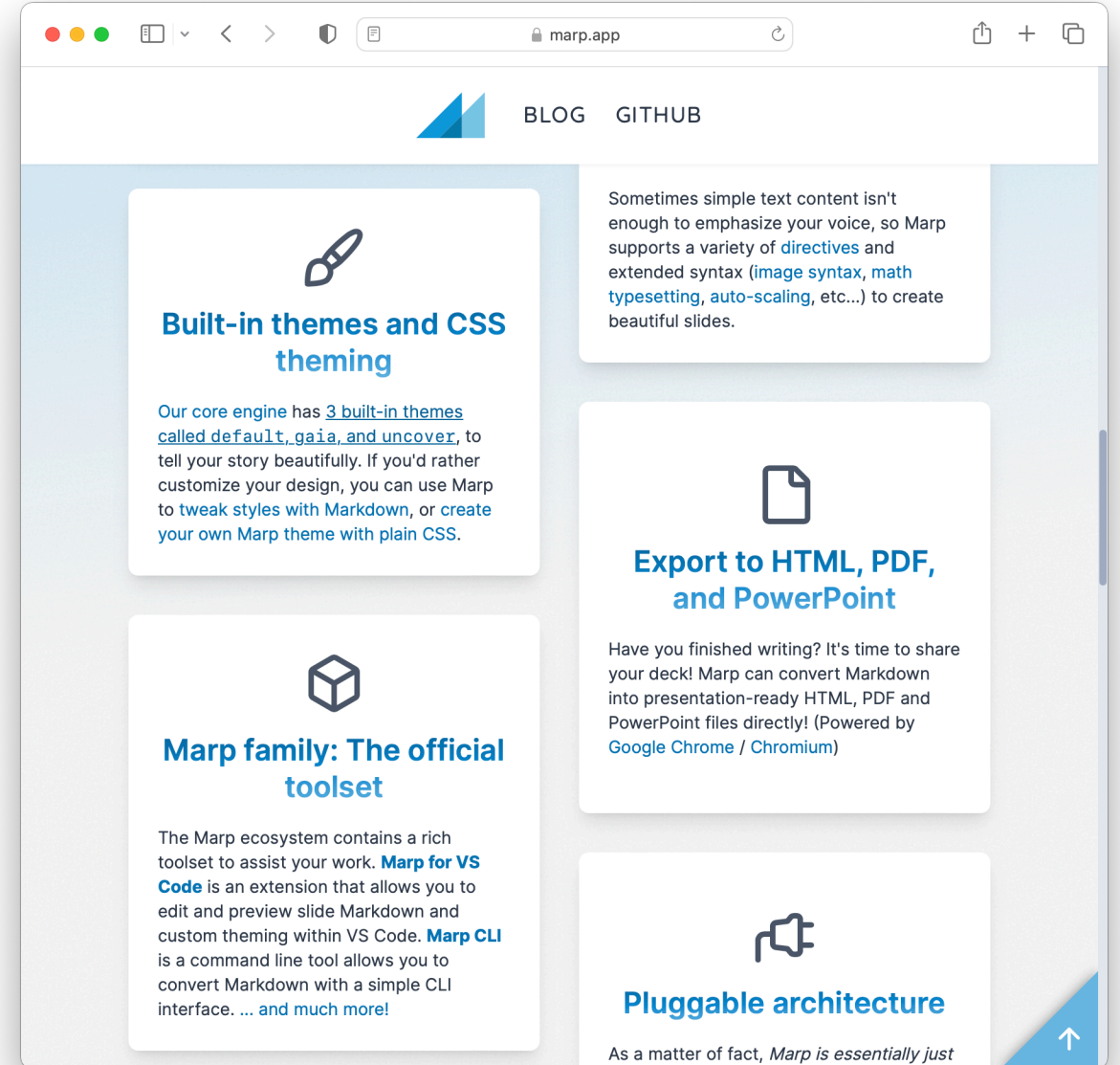
Markdown

- Lightweight markup language
- Created by John Gruber and Aaron Swartz in 2004
- Simple and versatile
- Used by many platforms and tools



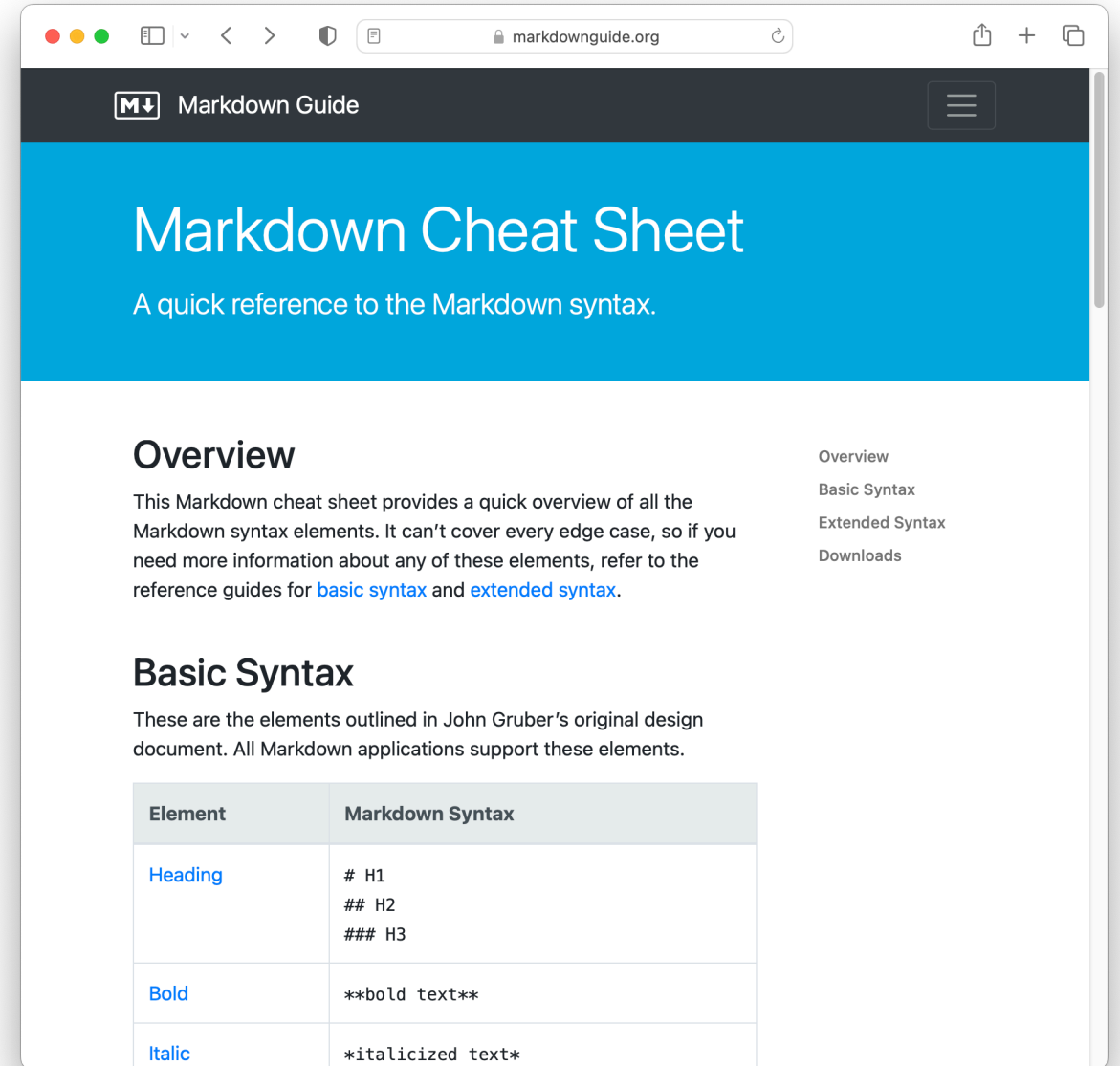
Markdown and output formats

- Multiple output formats
- Supports HTML integration
- Used in web pages, presentations, documentation



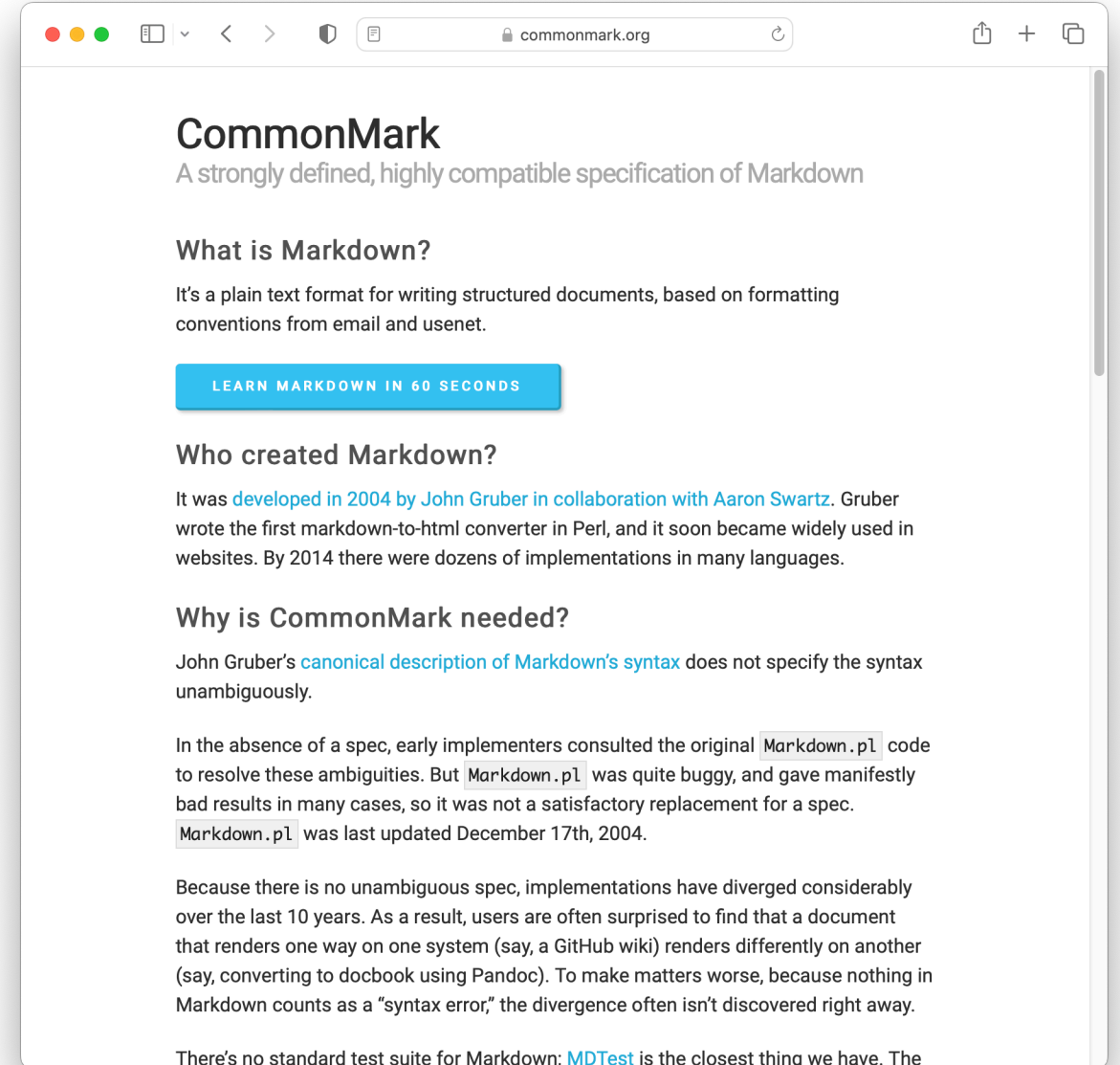
Syntax

- Markup language with special characters (`#` , `*` , `-` , `[]` , `!` `_` , etc.)
- Discord and Telegram uses Markdown for formatting
- Supports headings, text formatting, code blocks, lists, links, images, and tables



Specifications

- No single standard
- CommonMark and GitHub Flavored Markdown are the most popular
- Custom extensions for additional features on some platforms



Summary

- Markdown is easy to learn and use
- Markdown can be used to create documentation
- Markdown can be used to create slides
- Markdown can be exported to many formats

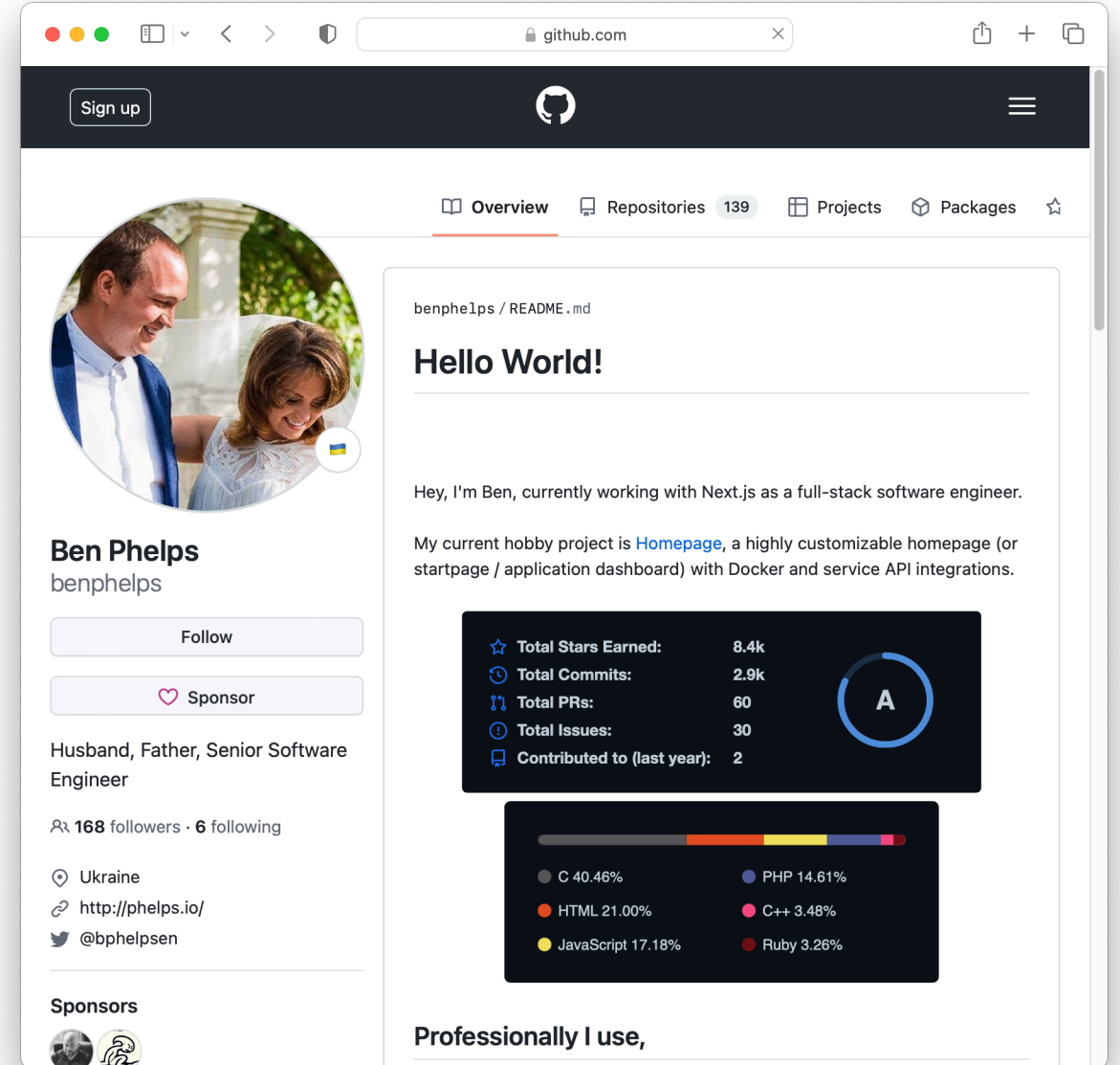
Questions

Do you have any questions?

Practical content

What will you do?

- Install and configure Git
- Create and configure your GitHub account
- Create your own GitHub profile README
- Add yourself to the GitHub organization README



Find the practical content

You can find the practical content for this chapter on [GitHub](#).



Finished? Was it easy? Was it hard?

Can you let us know what was easy and what was difficult for you during this chapter?

This will help us to improve the course and adapt the content to your needs. If we notice some difficulties, we will come back to you to help you.

 [GitHub Discussions](#)

You can use reactions to express your opinion on a comment!

What will you do next?

In the next chapter, you will learn the following topics:

- Java: why is Java a popular programming language?
- IntelliJ IDEA: tips and tricks for a better (collaborative) experience
- Maven: Java project structure, dependencies and build tool



Sources

- Main illustration by [Roman Synkevych](#) on [Unsplash](#)
- Illustration by [Aline de Nadai](#) on [Unsplash](#)
- Git logo by [Git](#)
- Git illustrations by [Atlassian](#) - [1](#), [2](#), [3](#), [4](#)
- Git strip by [xkcd](#)
- Illustration by [Nathan Dumlao](#) on [Unsplash](#)