



Git – Remote Repositories

Metodologias de Trabalho em Equipa

P.PORTO



Online hosting



- Git uses a “distributed” model that allows everyone working on a project to have their own independent copy of the entire repository.
- To collaborate effectively though we need a central version of the code base which is used to unify everyone's efforts.
- Typically, the best place for such a central repository is **online**

Source Code Hosting Services



- Code repositories for open-source software have been constantly maturing.
- There are many code hosting services and there is hardly any possibility to try them all and opt for an ultimate and final variant.
- However, let's focus on those that are most in demand.

Source Code Hosting Services



■ GitHub

- GitHub is the largest, and the most popular code hosting service. It started its work in 2008.
- It offers the following features:
 - A personal page for the project
 - Collaborative code review
 - Syntax highlighted feature for more than 200 programming languages
 - Wiki section
 - Code is handled through Git, but also there are options for importing SVN and Mercurial repositories
 - Branch comparison views
 - Open source Jabber, IRC, Bugzilla, Jira, integration.

Source Code Hosting Services



■ Bitbucket

- This code repository was also launched in 2008, and it takes up another leading position in the branch
- It offers the following features:
 - Mercurial and Git support
 - Unlimited number of public repositories
 - Unlimited number of private repositories for small teams (up to 5 persons)
 - Inline comments and threaded conversations
 - Bug tracking system
 - Pull requests for facilitation of code reviews
 - Branch permission for control over access levels
 - Branch comparison and commit history

Source Code Hosting Services



■ Source Forge

- It started as one of the most powerful sources in the field, but later it has been dislodged by the giants – GitHub and Bitbucket.
- Provides users with collaborative development tools and discussion and support instruments, forum and wiki.
- Supports OpenID and gives users possibility to have their unique sub-domain like *http://project-name.sourceforge.net*.
- It doesn't imply any paid subscriptions now.

Source Code Hosting Services



■ Gitlab

- Is one more popular web-based hosting service for projects.
- Its functions and structure are similar to GitHub, and the source itself is hosted on GitHub. However, there is one difference: Gitlab, unlike GitHub, allows using its software on third-party servers.
- It provides unlimited number of public and private repositories and unlimited number of collaborators (both public and private).
- Many notable companies use the source code repository tools it offers.



Github

Metodologias de Trabalho em Equipa

P.PORTO



Introducing ... GitHub!

“

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 50 million developers.

<https://github.com/>



GitHub – First steps



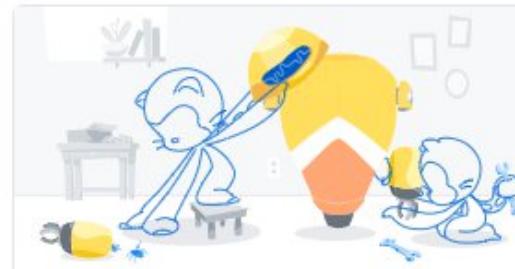
- Access <https://github.com/>
- Create an account
- Verify email address
 - An email containing verification instructions will be sent to you registered email

GitHub – First steps



What do you want to do first?

Every developer needs to configure their environment, so let's get your GitHub experience optimized for you.



Start a new project

Start a new repository or bring over an existing repository to keep contributing to it.

[Create a repository](#)

Collaborate with your team

Improve the way your team works together and get access to more features with an organization.

[Create an organization](#)

Learn how to use GitHub

Get started with an "Introduction to GitHub" course in our Learning Lab.

[Start Learning](#)[Skip this for now >](#)

Create a new repository

- Choose a repository name
- Choose Private
- Initialize repository
 - Check “Add a README file”
 - Check “Add a .gitignore”
 - Template: Java
 - Check “Choose a license”
 - GNU General Public ...
- Click “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 * **Repository name ***

 prpUpSkill  / MyUpSkillRepo

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

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.](#)
 .gitignore template: VisualStudio

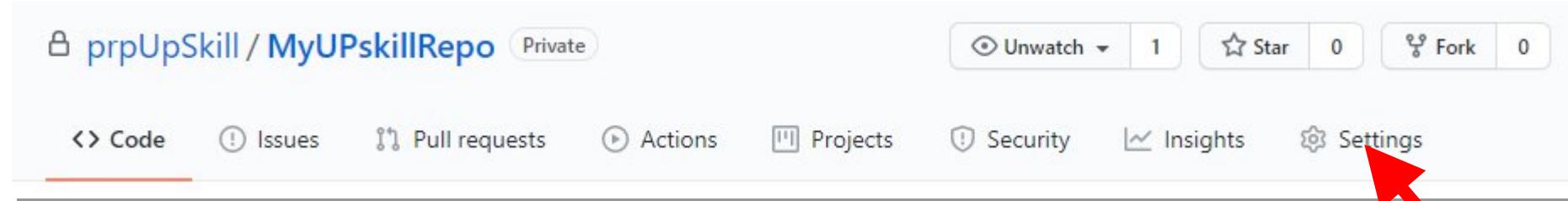
Choose a license
A license tells others what they can and can’t do with your code. [Learn more.](#)
 License: GNU General Public ...

This will set  main as the default branch. Change the default name in your [settings](#).

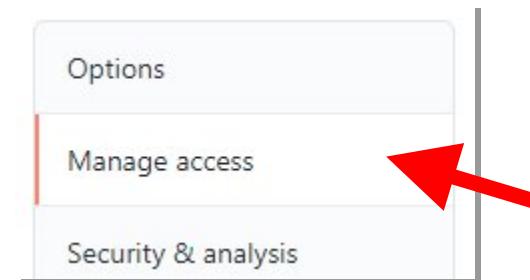
Create repository

Add collaborators to the project

■ Select Settings



■ Choose Manage access

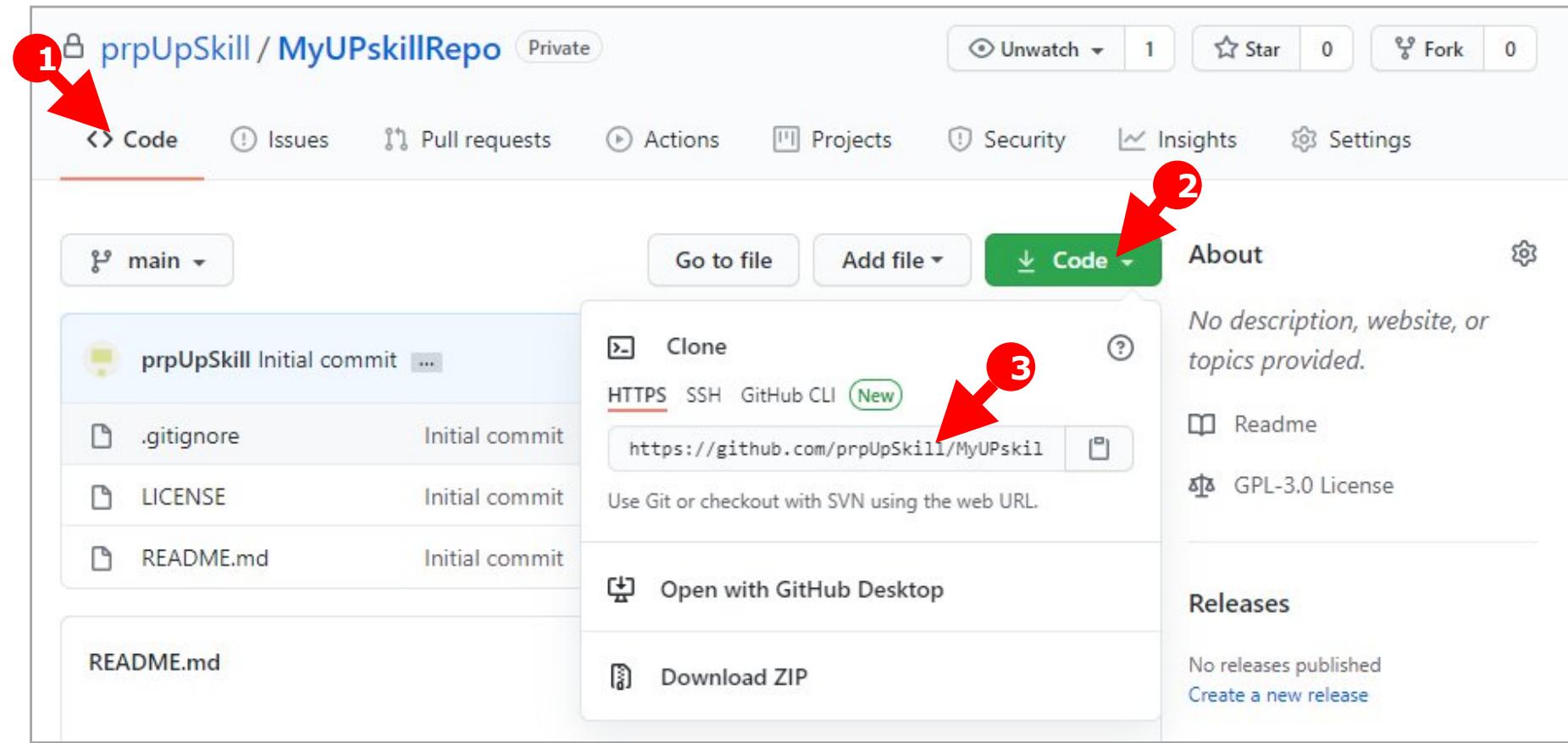


■ Invite a collaborator

- Search a collaborator by email



Clone the repository



Code & commits



A screenshot of a GitHub repository interface. The top navigation bar includes 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Security', 'Insights', and a settings gear icon. Below this, a dropdown shows 'main' branch, '1 branch', and '0 tags'. Buttons for 'Go to file', 'Add file', and a green 'Code' button are present. The main area displays a commit titled 'prpUpSkill Initial commit' made 24 minutes ago. A red arrow points to the 'Code' button. Below the commit, three files are listed: '.gitignore', 'LICENSE', and 'README.md', all marked as 'Initial commit' and 24 minutes ago. A red box highlights these three files.

File	Type	Time
.gitignore	Initial commit	24 minutes ago
LICENSE	Initial commit	24 minutes ago
README.md	Initial commit	24 minutes ago

.gitignore

- This file specifies intentionally untracked files that Git should ignore
 - A line starting with a **#** serves as a comment
 - Each line specifies a pattern

```
# excludes everything in directory foo
/foos/*
```

- The prefix **!** negates the pattern

```
# excludes everything except directory
foo/bar
/foos/*
!/foos/bar
```

A typical .gitignore for java IntelliJ Idea



```
# IntelliJ Idea specific
.idea/*

# File-based project format
*.iws

# Class Files
*.class

# Package Files
*.jar
*.war
*.ear
```

Adding a remote



- To add a remote use `git remote add` command in the directory your repository is stored at.
- This command takes two arguments:
 - a remote name, for example, *origin*
 - a remote URL

```
$ git remote add origin https://github.com/prpUpSkill/MyUPskillRepo.git  
  
$ git remote -v  
origin https://github.com/prpUpSkill/MyUPskillRepo.git (fetch)  
origin https://github.com/prpUpSkill/MyUPskillRepo.git (push)
```

Cloning a repository



- If you have the address (and correct permissions) for an online repository, then you can grab your own copy using the `git clone` command.

```
$ git clone https://github.com/{user}/Project.git
```

- Now you have your own copy of the repository and can do whatever you want with it.

Collaboration basic work-flow



- Make your changes in your own personal copy of the repo, ideally in a new branch.
- “Pull” the most recent version of the remote repo into your master branch.

```
$ git pull
```

- Merge your changes from your new branch into master.
- Once any conflicts are resolved you can update the remote repo with your code.

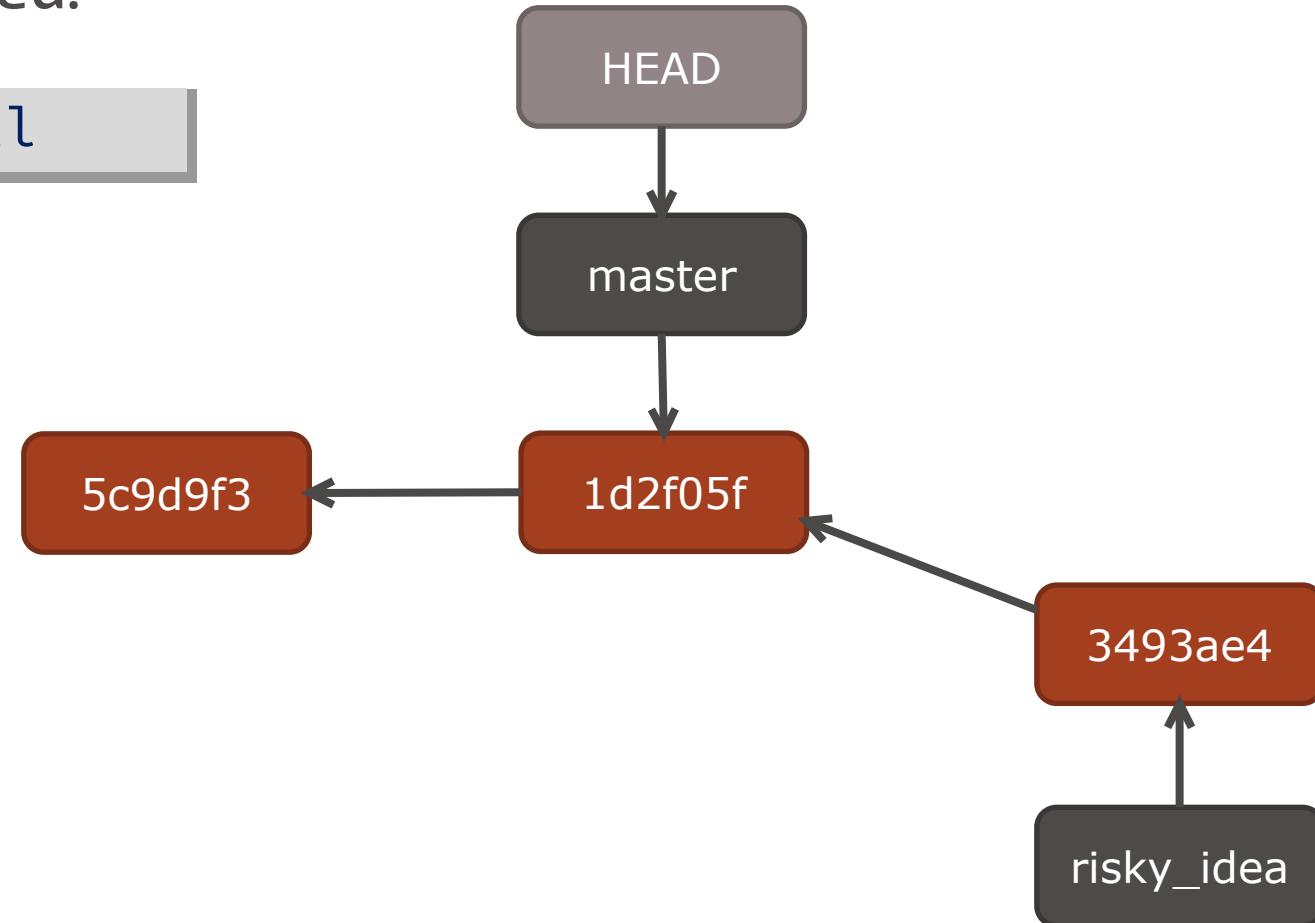
```
$ git push
```

How git pull works

Someone else pushed.

Before

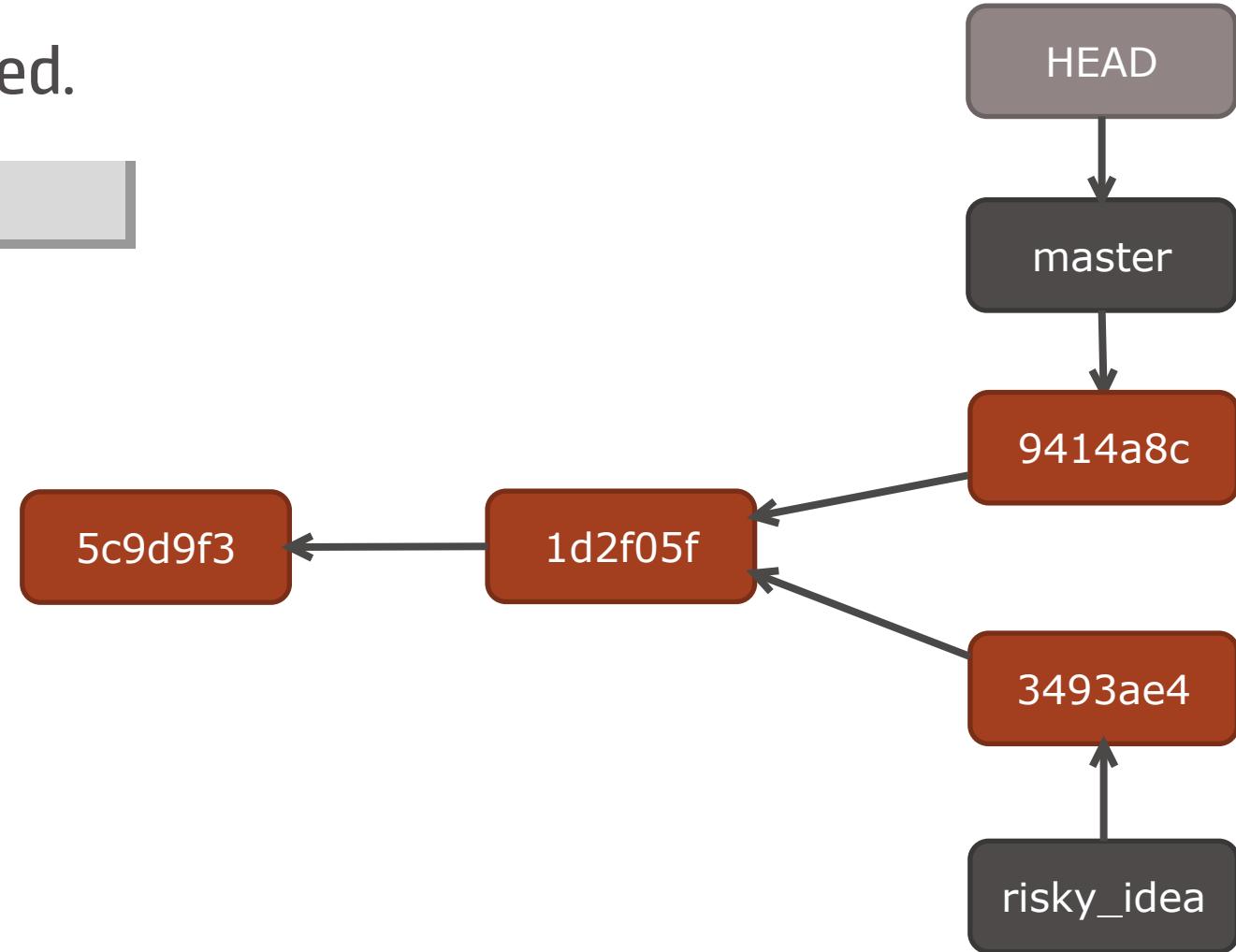
```
$ git pull
```



How git pull works

Someone else pushed.

After `$ git pull`



git-fetch



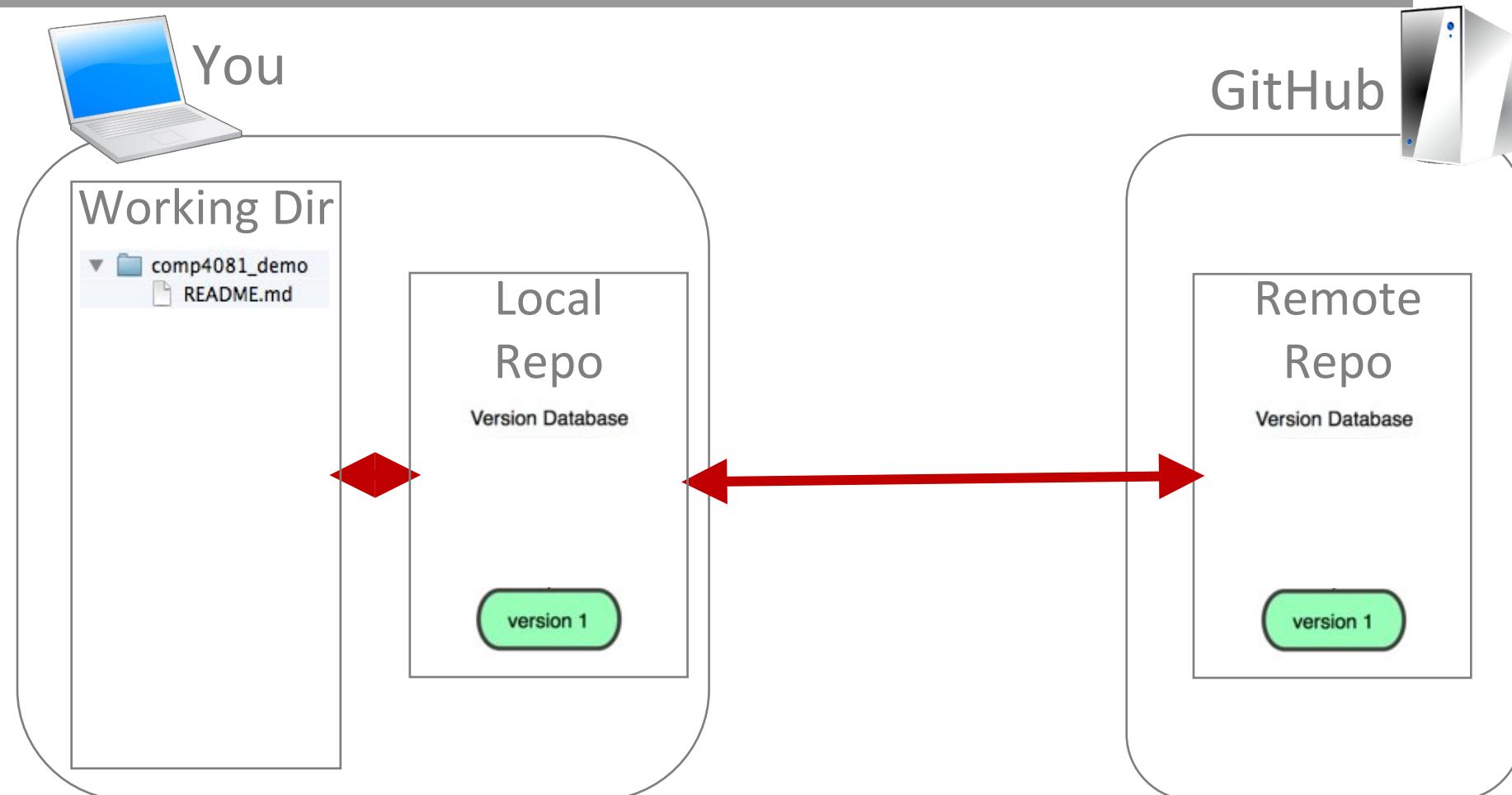
- In Git there are two versions of the local repository:
 - One contains your code with its changes
 - The other mirrors the remote repository
- These two versions of the local repository support diff and merge commands
- `git fetch` can be used to download all commits from the remote repository without affecting the local code.
- Basically, `git pull` does a `git fetch` first followed by a `git merge origin / master`.

Example



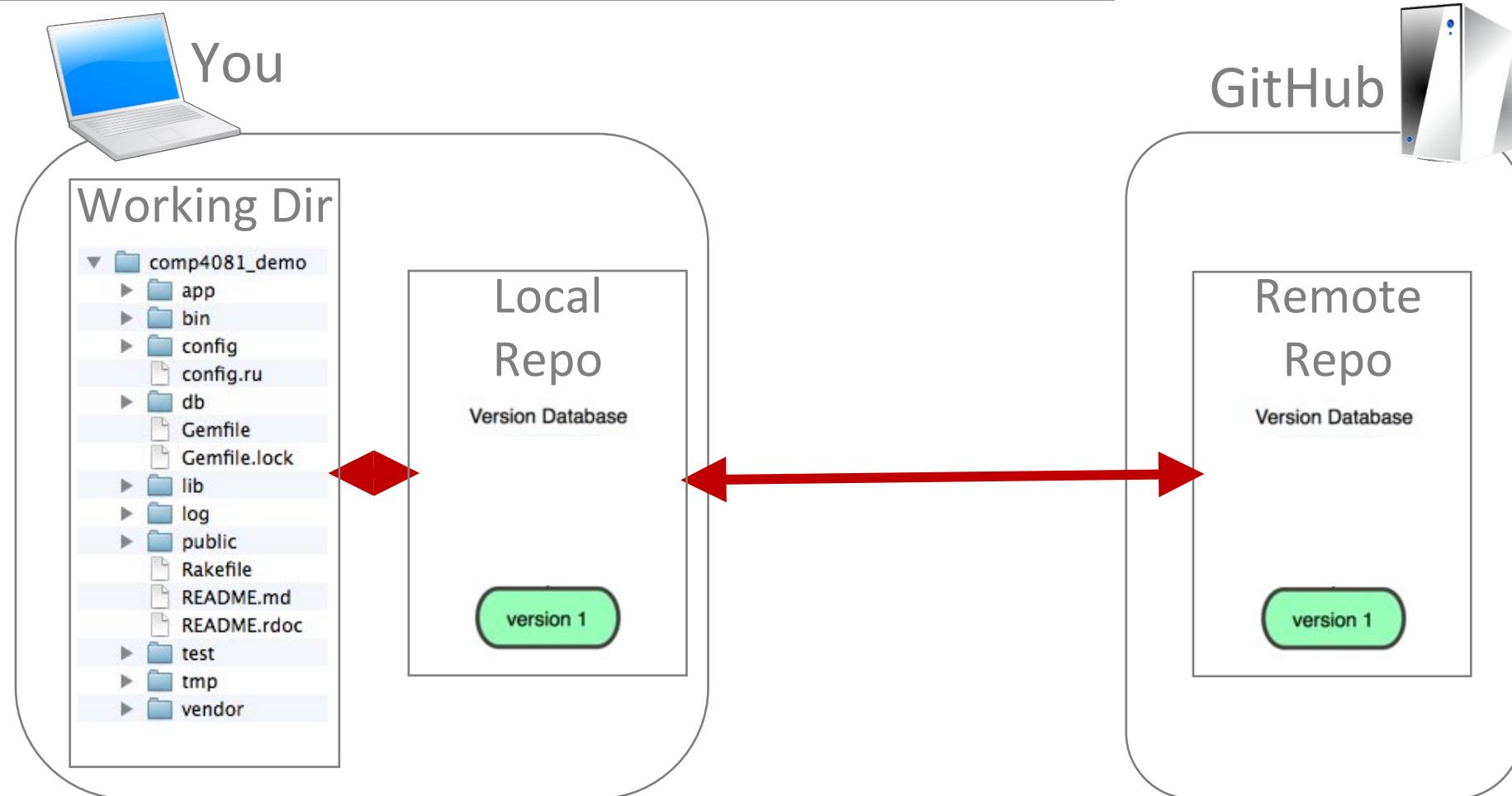
Example

```
$ git clone https://github.com/{user}/comp4081_demo.git
```

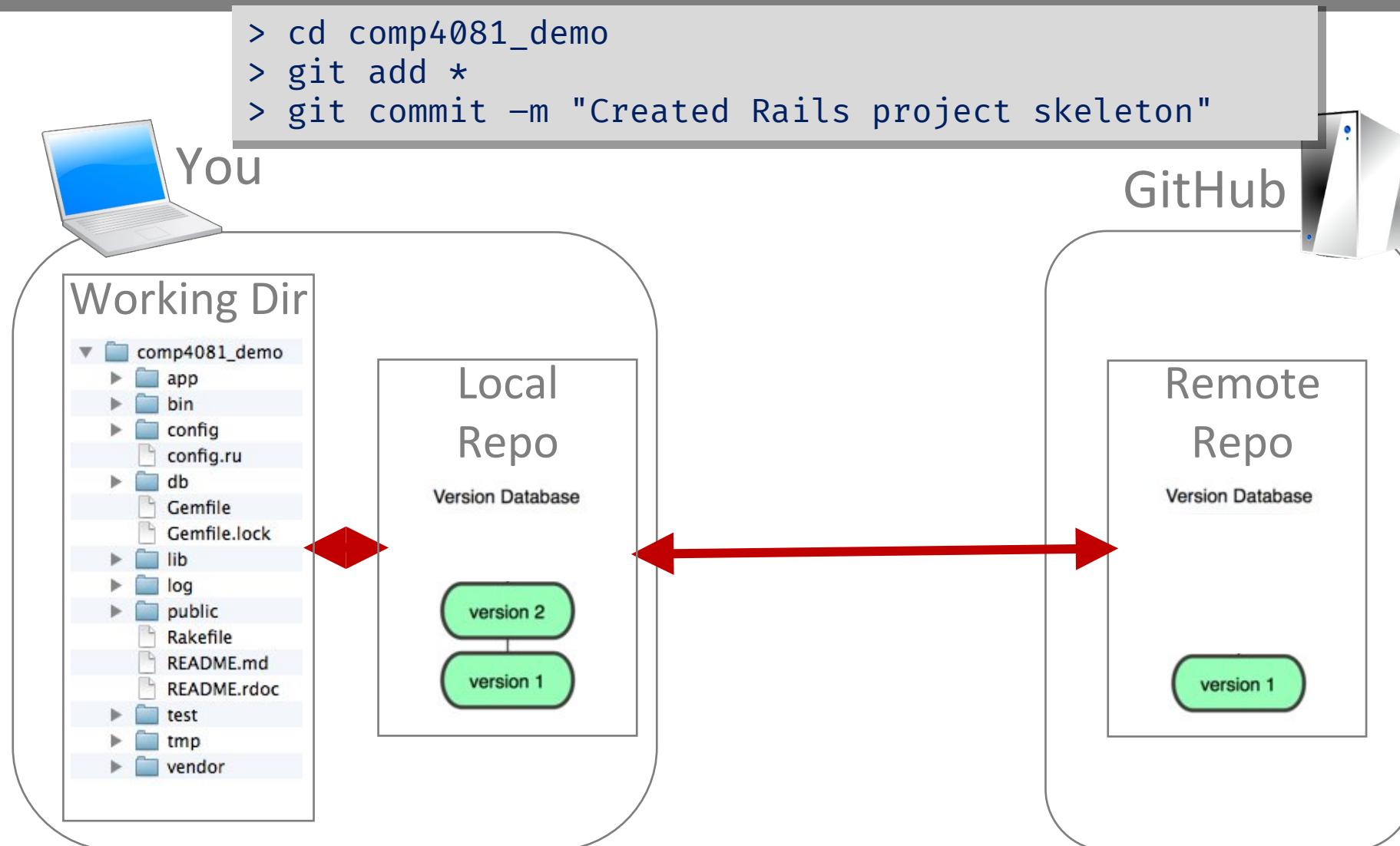


Example

```
$ rails new comp4081_demo
```

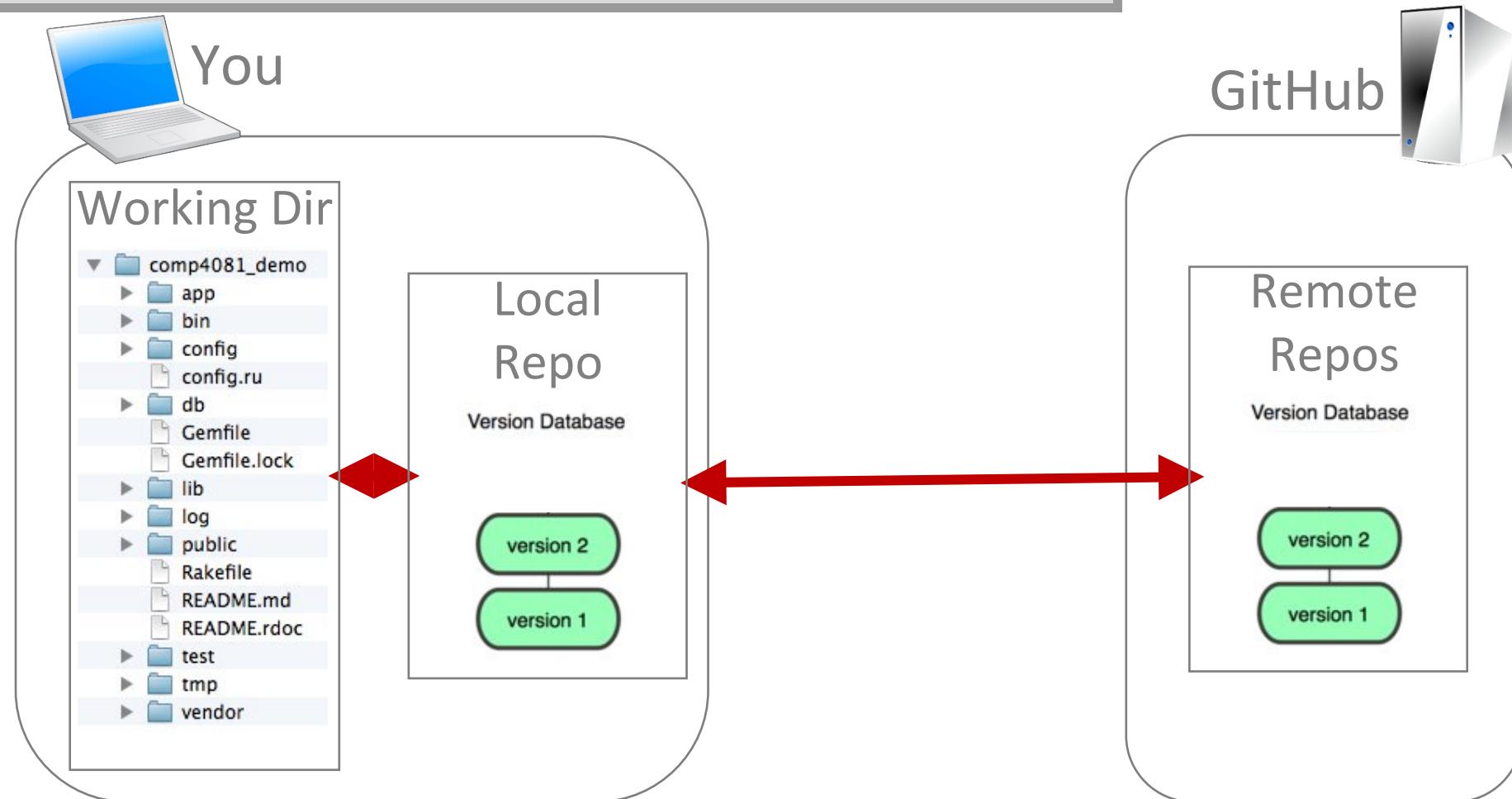


Example



Example

```
$ git push
```



Git conflicts



■ Merge the branch

Hello.html

```
<head>
<<<<< HEAD
    <link type="text/css" rel="stylesheet" media="all" href="style.css" />
=====
    <!-- no style -->
>>>>> master
</head>
<body>
    <h1>Hello,World! Life is great!</h1>
</body>
</html>
```

Git conflicts



■ Resolution of the conflict

Hello.html

```
<head>
    <link type="text/css" rel="stylesheet" media="all" href="style.css" />
</head>
<body>
    <h1>Hello,World! Life is great!</h1>
</body>
</html>
```

Git conflicts

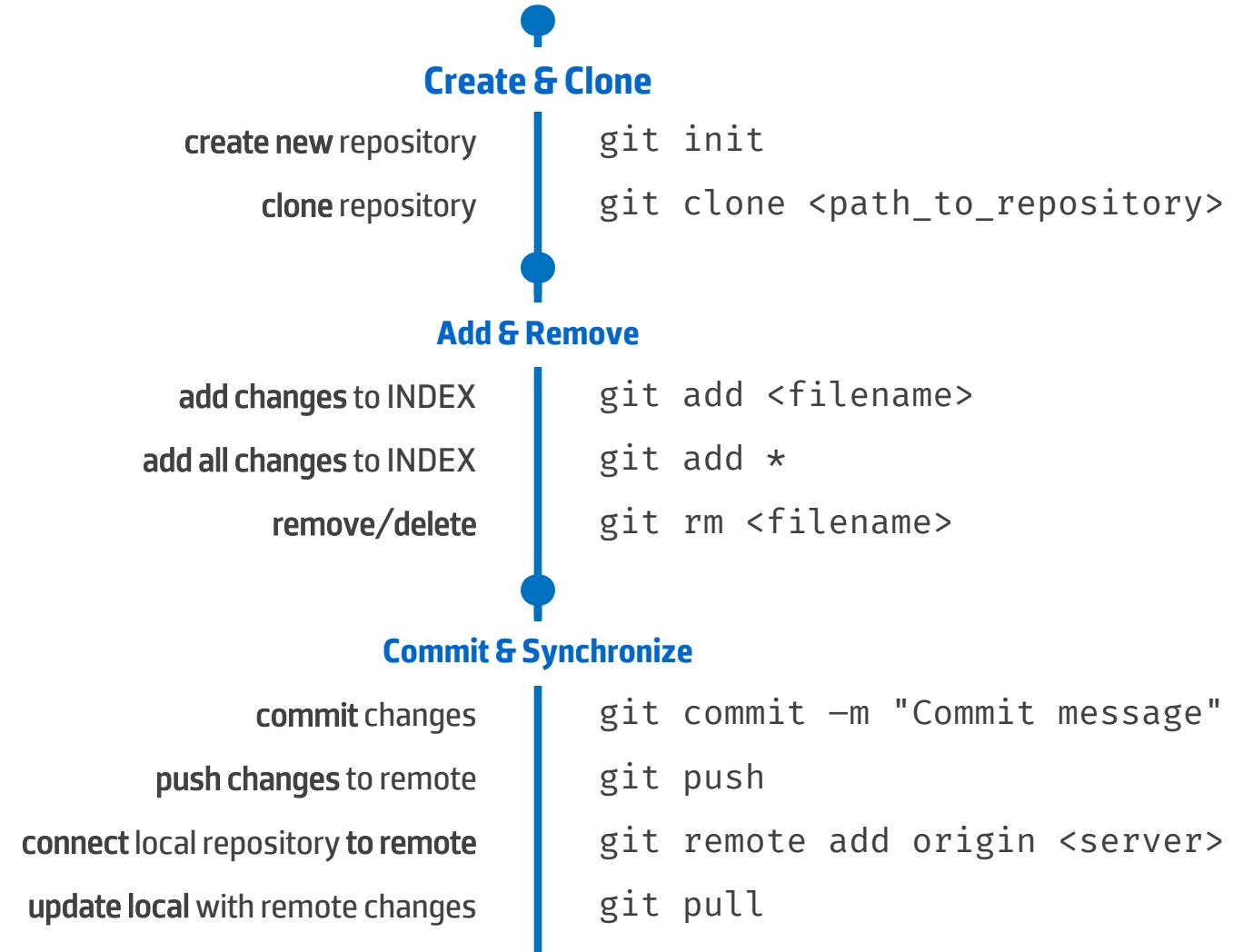


- Make a commit of conflict resolution

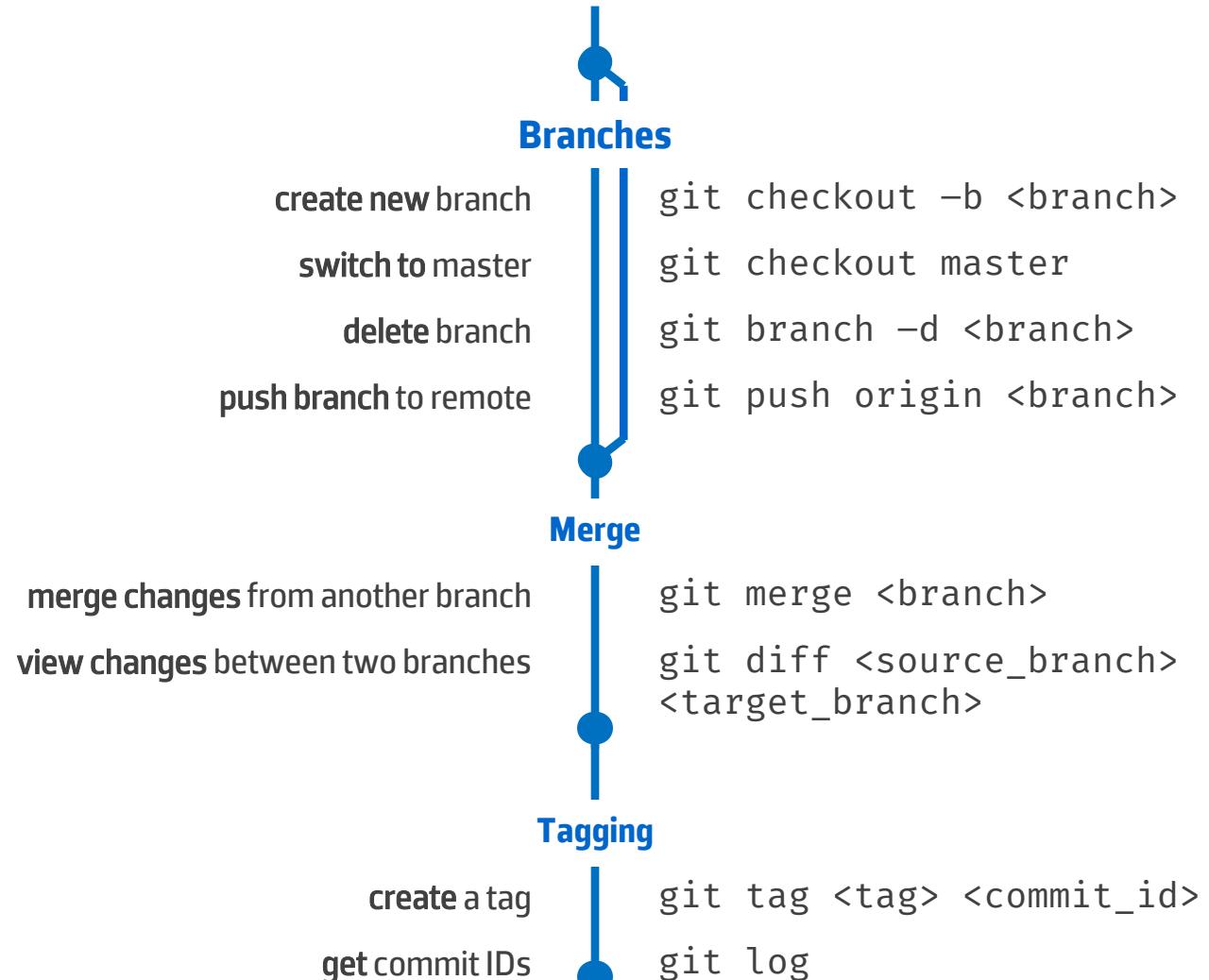
```
> git add Hello.html
> git commit -m "Merged master fixed
conflict."
```

Recorded resolution for 'Hello.html'.
[style 645c4e6] Merged master fixed
conflict.

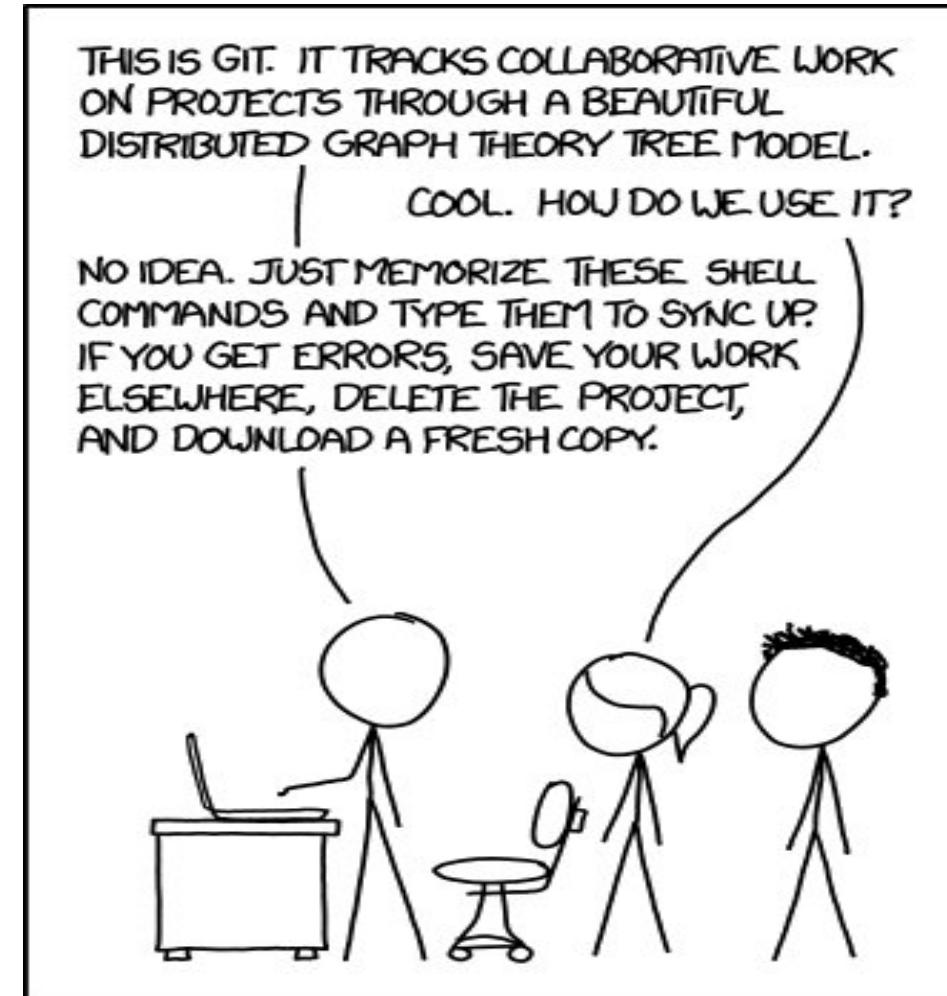
Git Cheat Sheet



Git Cheat Sheet



That's all ... about Git!



Git – Remote Repositories

Metodologias de Trabalho em Equipa

P.PORTO