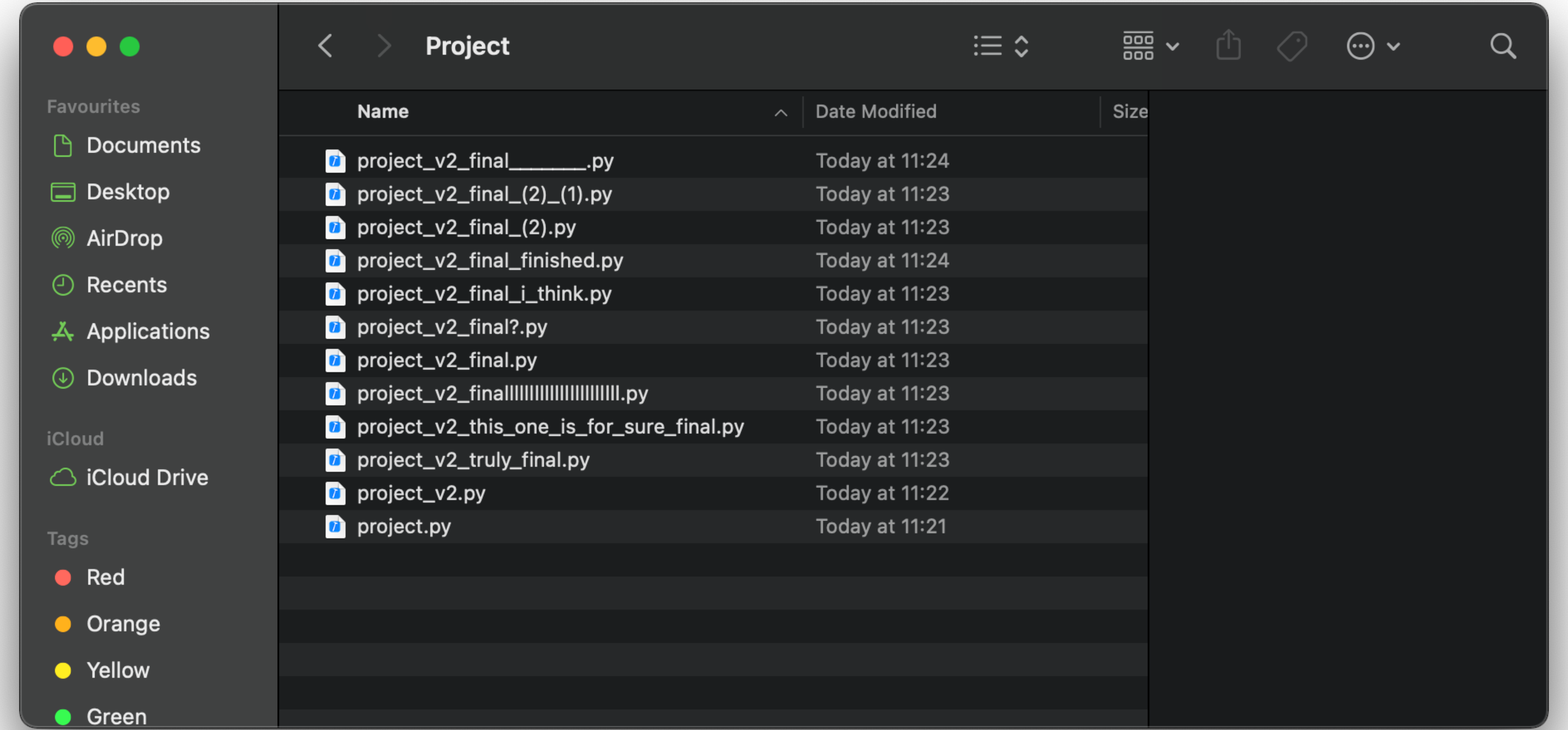
git + GitHub

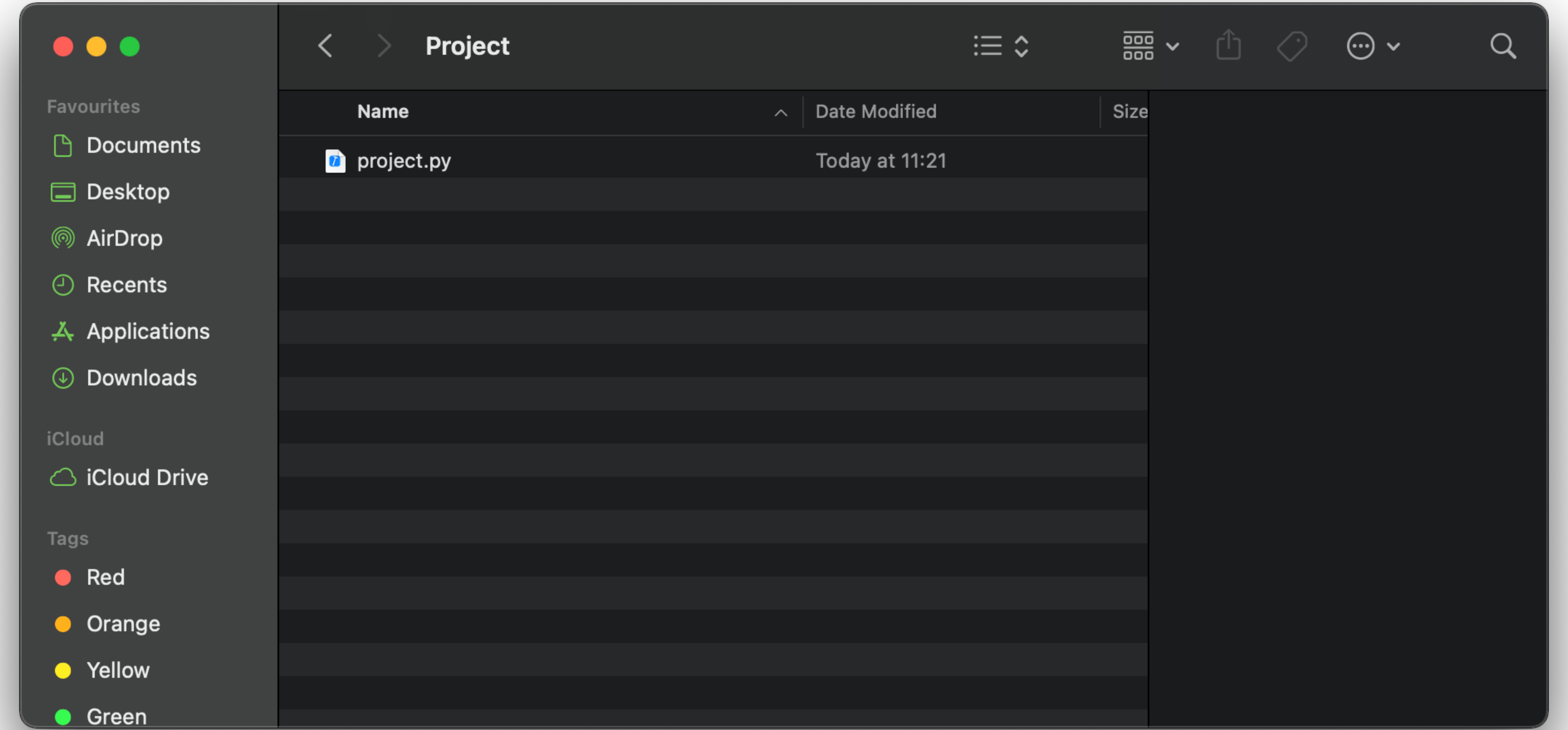
Data Science Tutorial

Version Control and why you need it



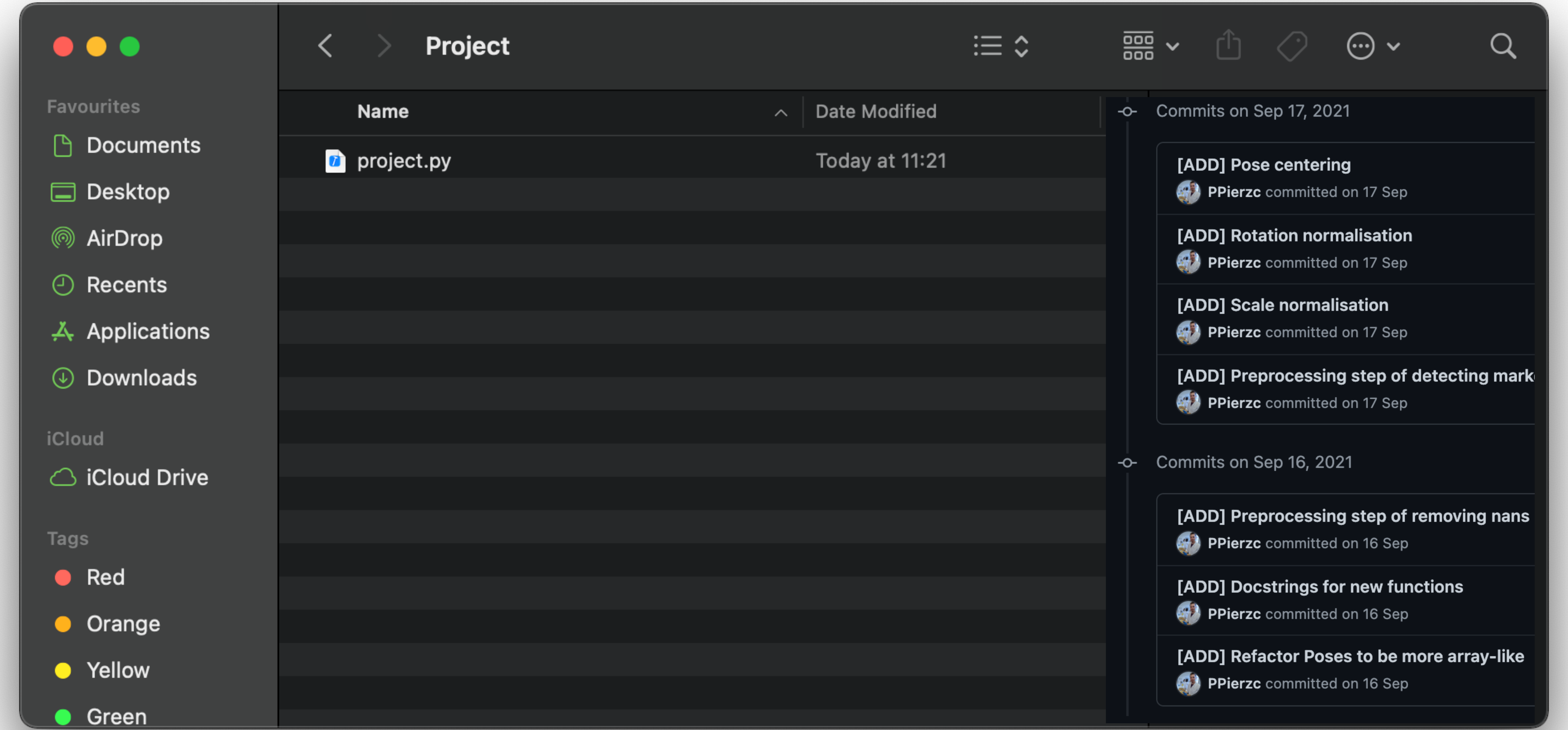
Name	Date Modified	Size
project_v2_final_________.py	Today at 11:24	
project_v2_final_(2)_(1).py	Today at 11:23	
project_v2_final_(2).py	Today at 11:23	
project_v2_final_finished.py	Today at 11:24	
project_v2_final_i_think.py	Today at 11:23	
project_v2_final?.py	Today at 11:23	
project_v2_final.py	Today at 11:23	
project_v2_finallllllllllllllllllllllll.py	Today at 11:23	
project_v2_this_one_is_for_sure_final.py	Today at 11:23	
project_v2_truly_final.py	Today at 11:23	
project_v2.py	Today at 11:22	
project.py	Today at 11:21	

Version Control and why you need it



Name	Date Modified	Size
project.py	Today at 11:21	

Version Control and why you need it



Name	Date Modified
project.py	Today at 11:21

Commits on Sep 17, 2021

[ADD] Pose centering
PPierzc committed on 17 Sep

[ADD] Rotation normalisation
PPierzc committed on 17 Sep

[ADD] Scale normalisation
PPierzc committed on 17 Sep

[ADD] Preprocessing step of detecting mark
PPierzc committed on 17 Sep

Commits on Sep 16, 2021

[ADD] Preprocessing step of removing nans
PPierzc committed on 16 Sep

[ADD] Docstrings for new functions
PPierzc committed on 16 Sep

[ADD] Refactor Poses to be more array-like
PPierzc committed on 16 Sep

Popular options for version control

Everyone has git installed?

https://github.com/git-guides/install-git

Popular options for version control

Everyone has git installed?

https://github.com/git-guides/install-git

Alternatives

Popular options for version control

Alternatives

Everyone has git installed?

https://github.com/git-guides/install-git

Repositories and where to find them

Definition

A collection of files and their different versions.

Everyone has a GitHub account and access token?

Repositories and where to find them

Definition

A collection of files and their different versions.

Popular git cloud solutions for storing repositories

GitHub

GitLab

Bitbucket

Everyone has a GitHub account and access token?

Repositories and where to find them

Definition

A collection of files and their different versions.

Popular git cloud solutions for storing repositories

GitHub

GitLab

Bitbucket

Everyone has a GitHub account and access token?

Repositories and where to find them

Definition

A collection of files and their different versions.

Popular git cloud solutions for storing repositories

GitHub

GitLab

Bitbucket

https://github.com/PPierzc/gh-tutorial

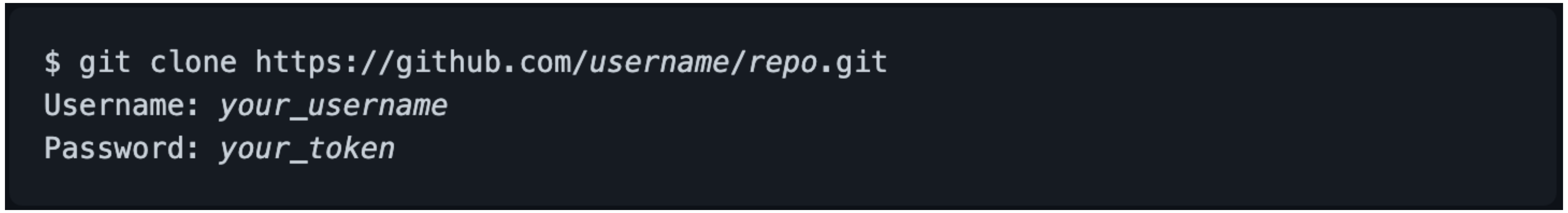
Everyone has a GitHub account and access token?

Install Git and create a Github account

https://github.com/git-guides/install-git

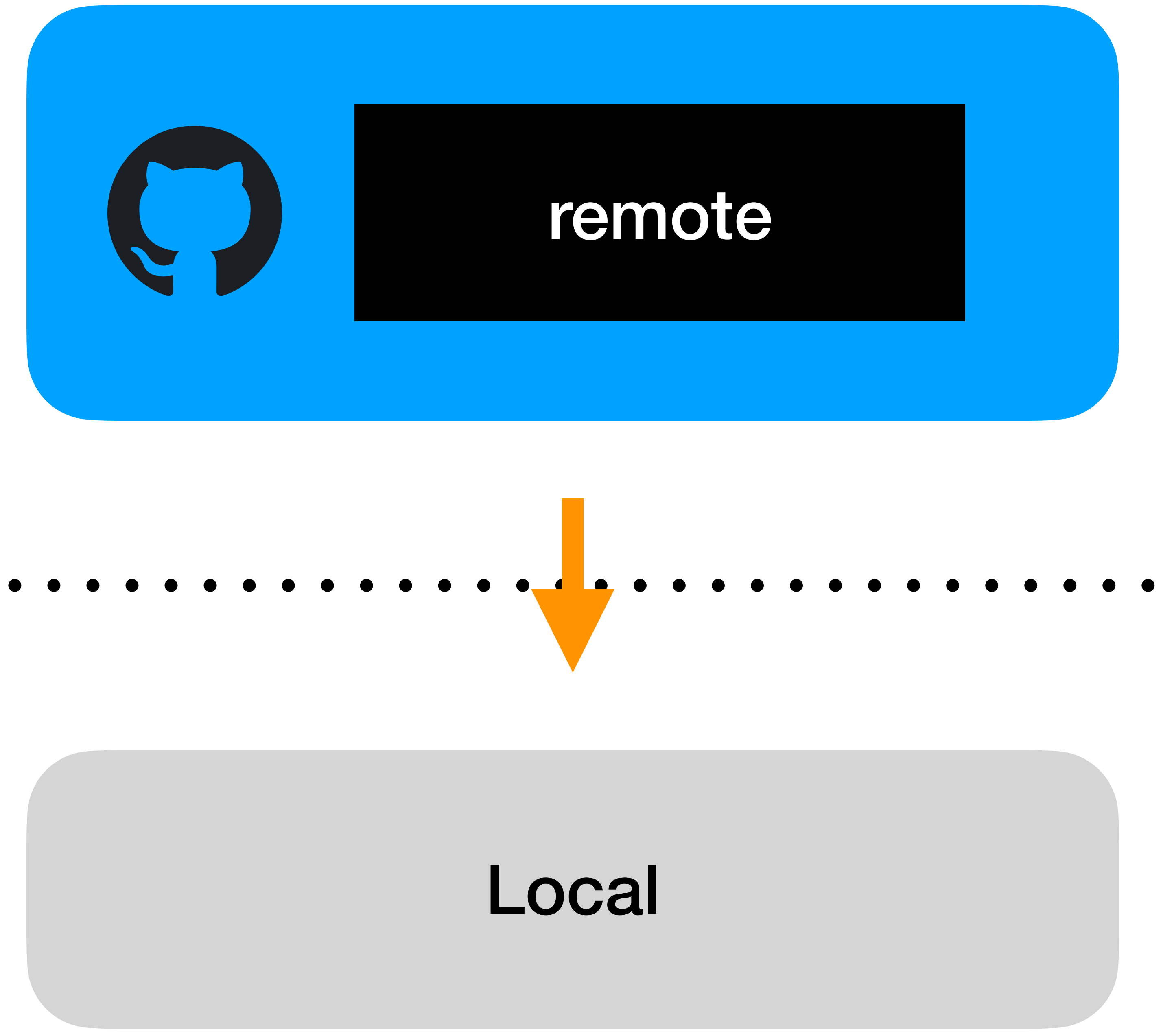
Create GitHub account

Generate access token



```
$ git clone https://github.com/username/repo.git
Username: your_username
Password: your_token
```

Getting files: cloning repositories

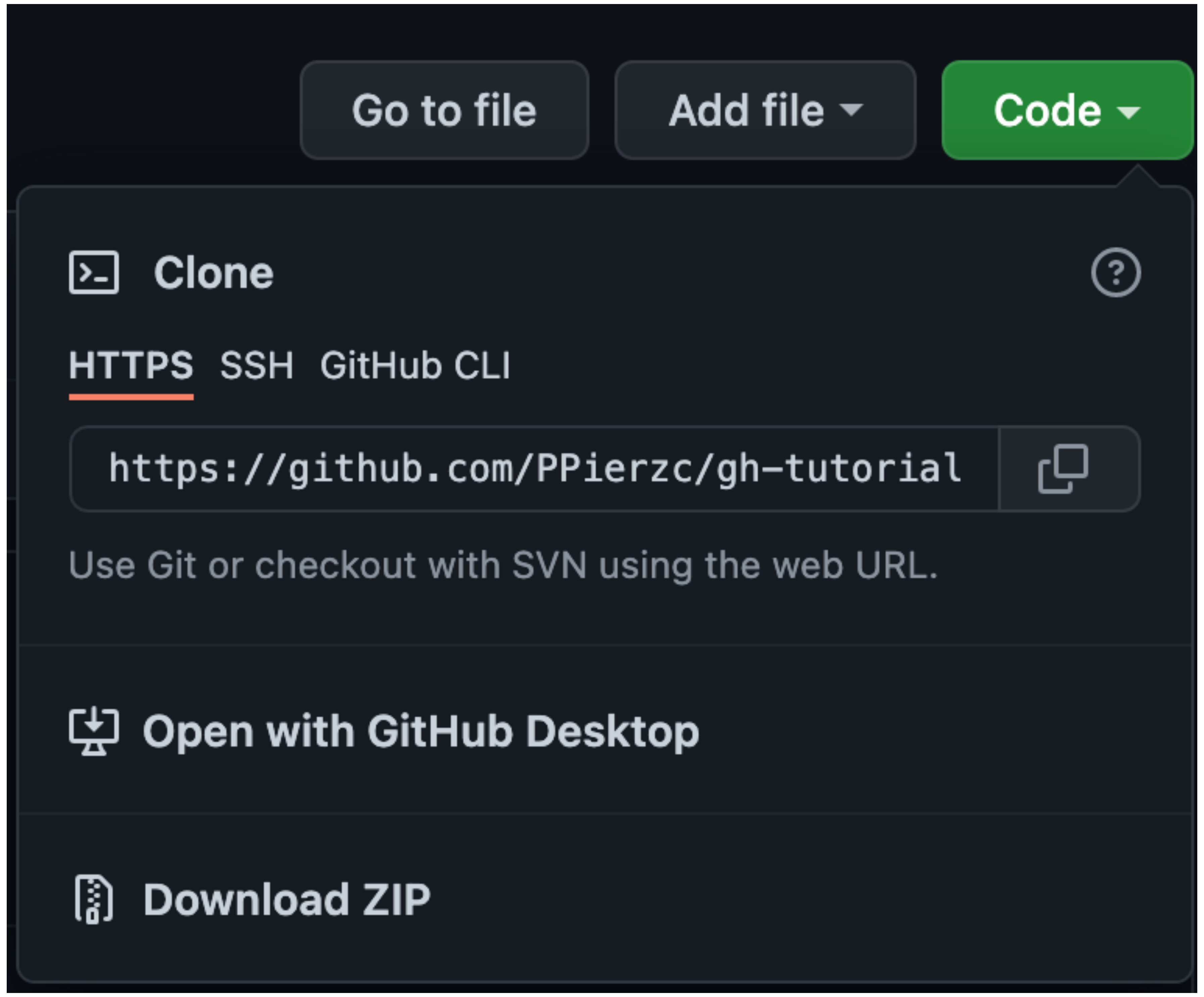


remote

Local

Cloning
Makes a copy of a ***remote*** git repository

```
git clone <repository_url>
```



https://github.com/PPierzc/gh-tutorial

Use Git or checkout with SVN using the web URL.

Creating your own repository

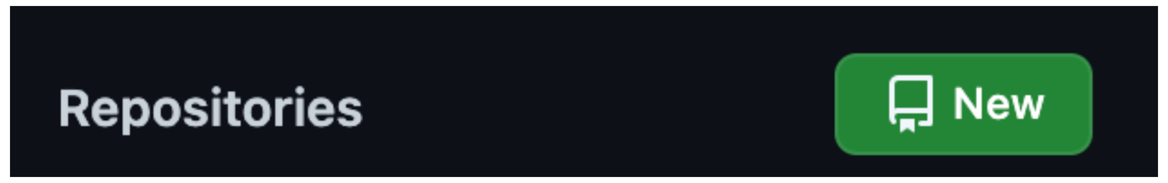
Easiest option is through

GitHub

Creating your own repository

Easiest option is through

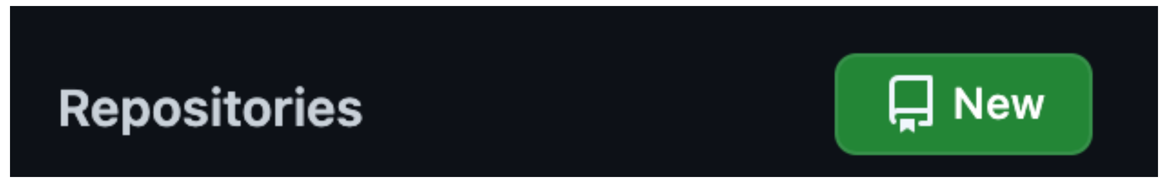
GitHub



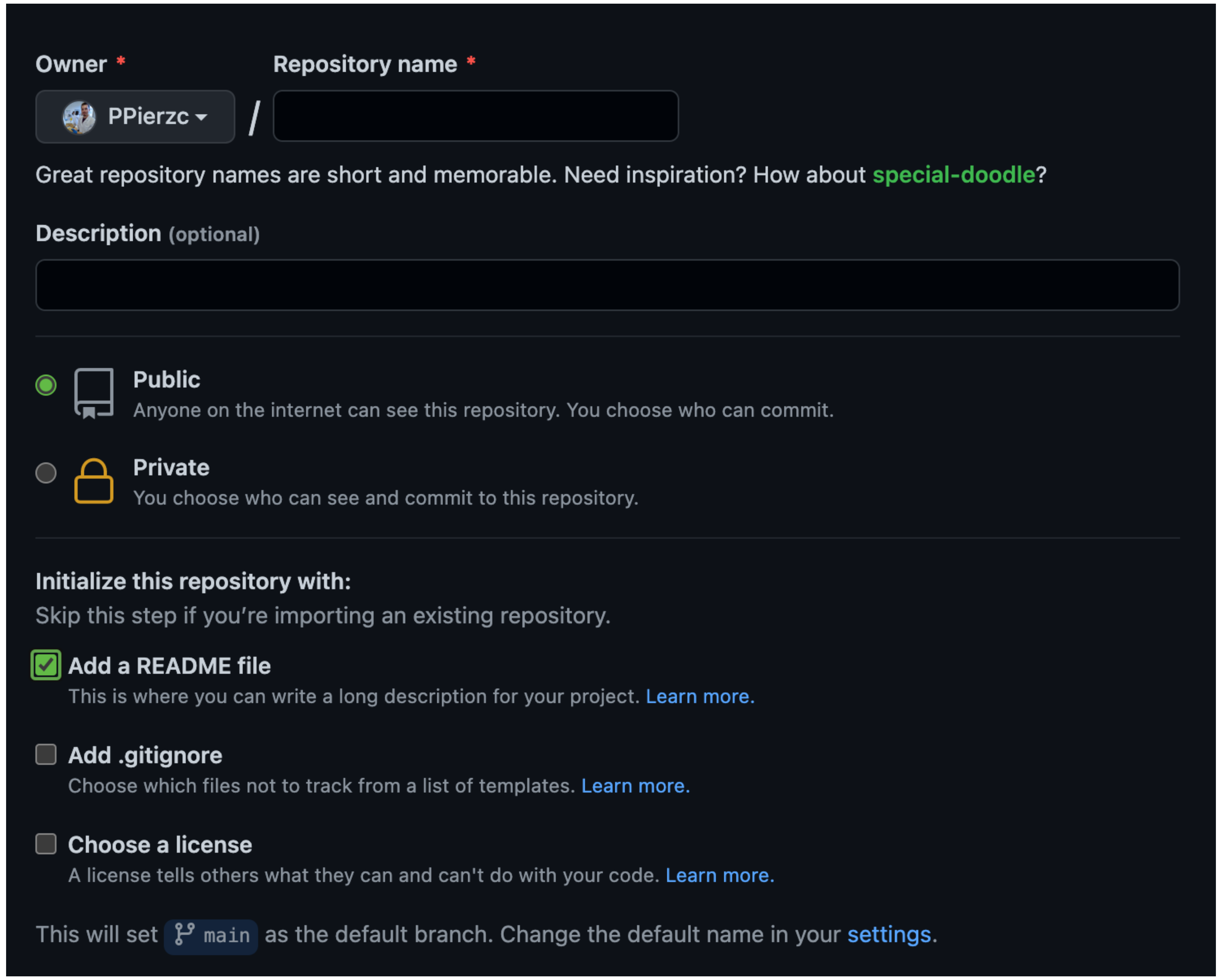
Creating your own repository

Easiest option is through

GitHub



Repositories



Owner *

Repository name *

PPierzc /

Great repository names are short and memorable. Need inspiration? How about special-doodle?

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.

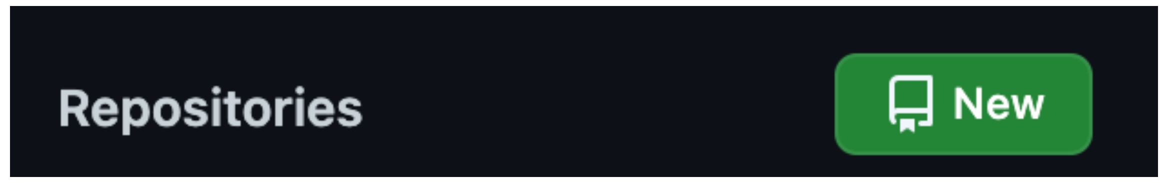
□ Choose a license
A license tells others what they can and can't do with your code. Learn more.

This will set `main` as the default branch. Change the default name in your settings.

Creating your own repository

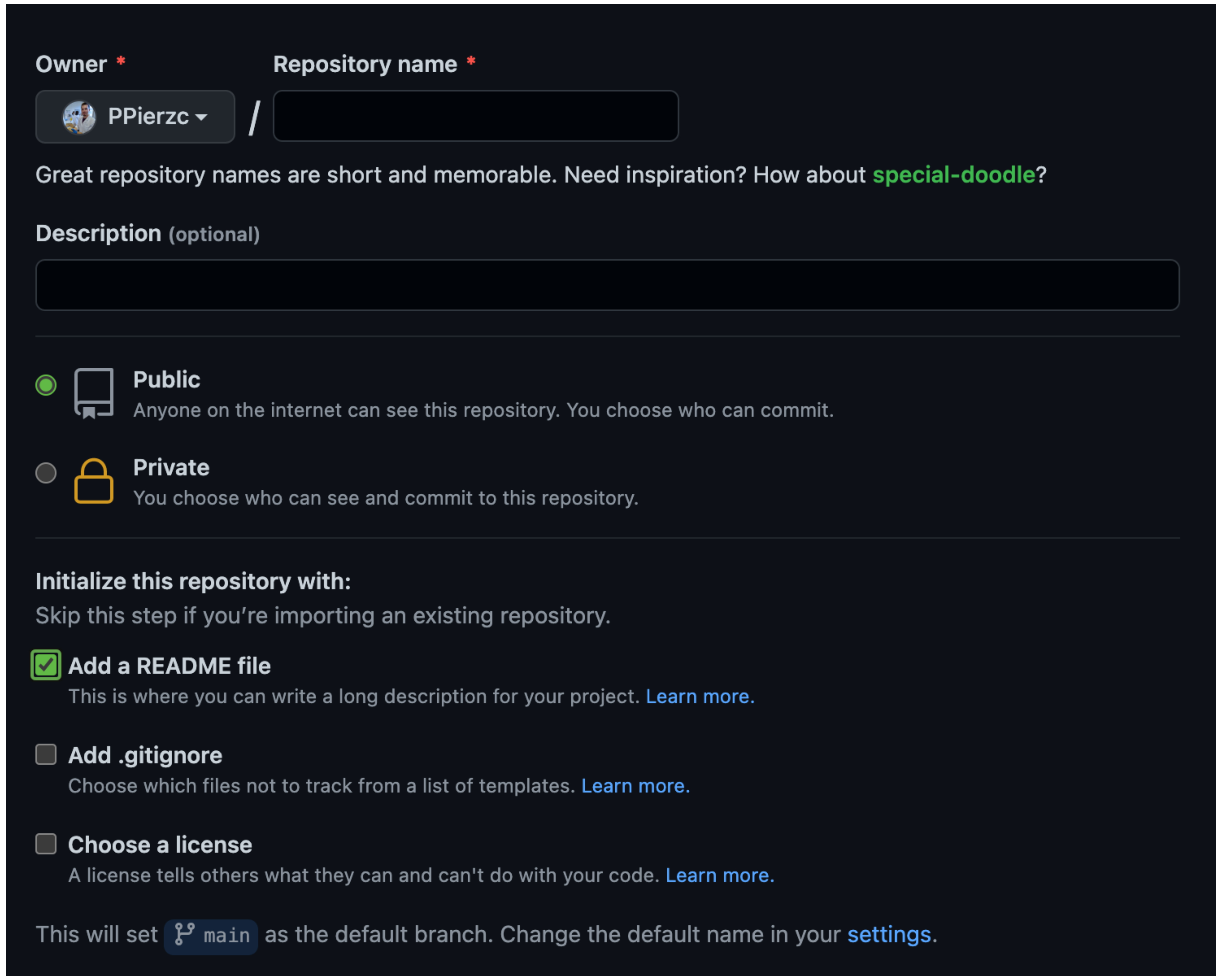
Easiest option is through

GitHub



Repositories

New



Owner *

Repository name *

PPierzc ▾ /

Great repository names are short and memorable. Need inspiration? How about special-doodle?

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.

This will set `main` as the default branch. Change the default name in your settings.

```
git clone REMOTE_URL
```

Add files: add, commit and push

Status
Shows you the tracked and untracked files, committed and uncommitted changes etc.

```
git status
```

Add files: add, commit and push

Status
Shows you the tracked and untracked files, committed and uncommitted changes etc.

```
git status
```

Adding
Adds a file to be tracked

```
git add <filename>
```

Add files: add, commit and push

Status
Shows you the tracked and untracked files, committed and uncommitted changes etc.

```
git status
```

Adding
Adds a file to be tracked

```
git add <filename>
```

Committing
Adding files as a commit i.e. version

```
git commit -m “commit message”
```

Add files: add, commit and push

Status
Shows you the tracked and untracked files, committed and uncommitted changes etc.

```
git status
```

Adding
Adds a file to be tracked

```
git add <filename>
```

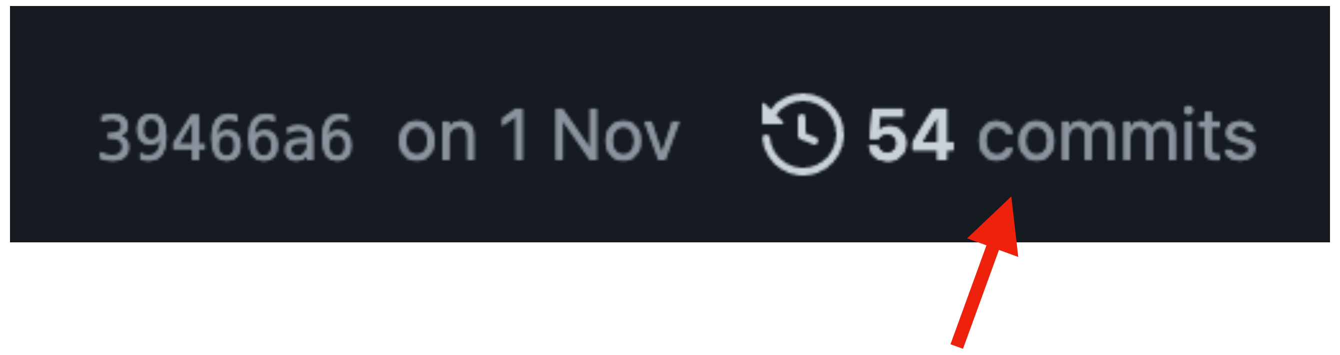
Committing
Adding files as a commit i.e. version

```
git commit -m “commit message”
```

Pushing
Uploads the commits to the remote

```
git push
```

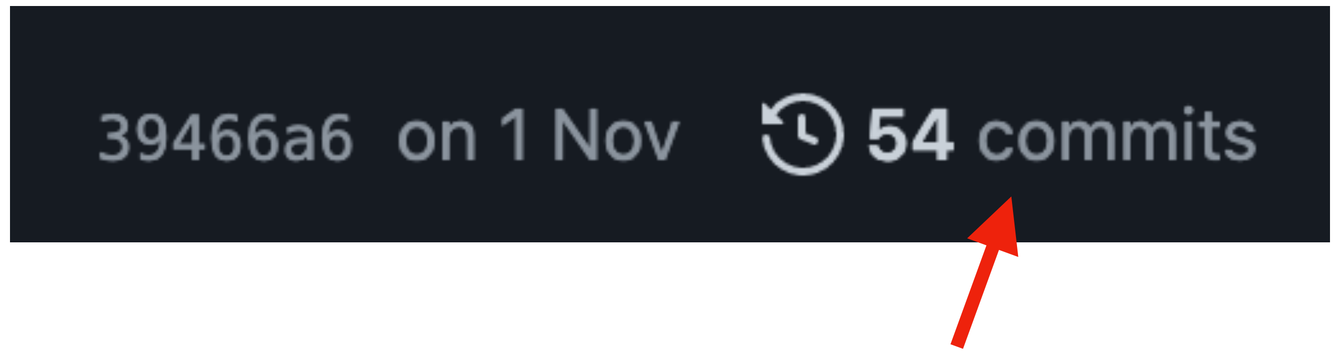
You can check the history on Github



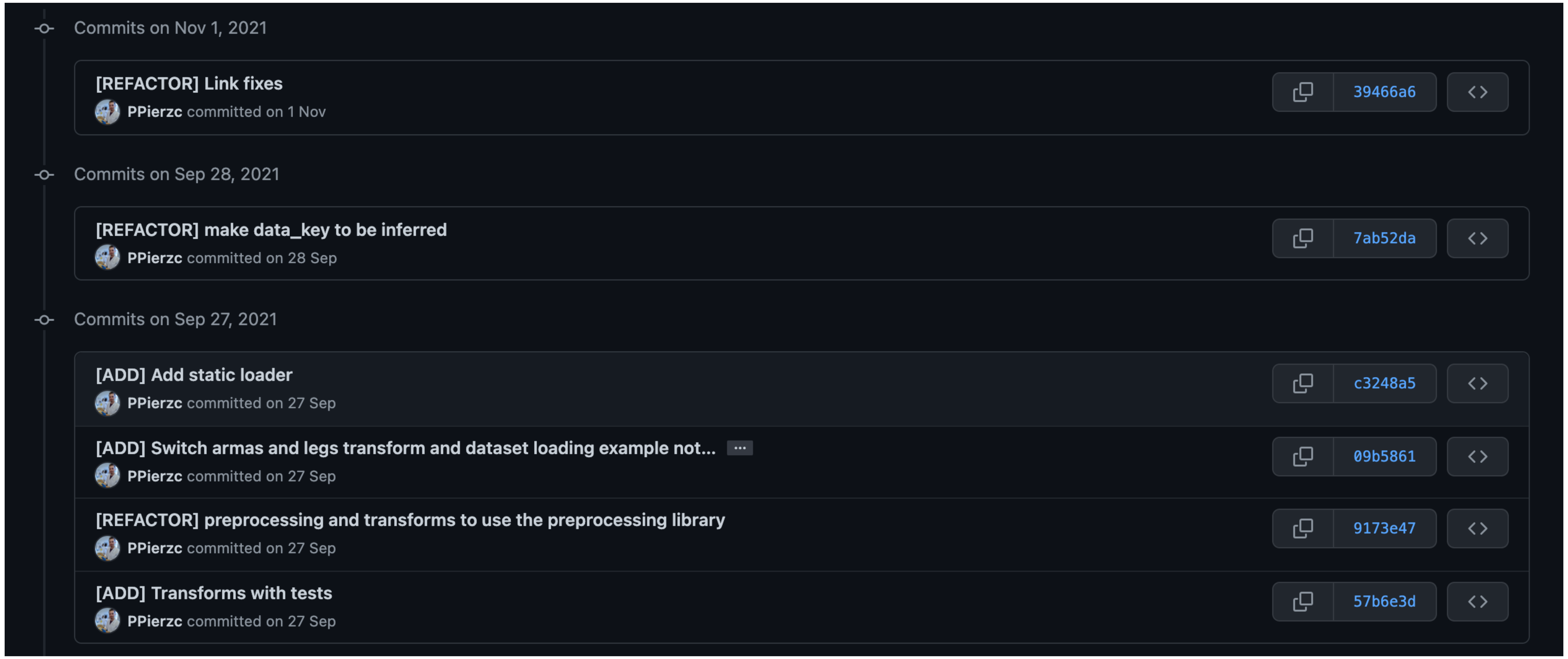
39466a6 on 1 Nov

54 commits

You can check the history on Github



39466a6 on 1 Nov 54 commits



Commits on Nov 1, 2021

[REFACTOR] Link fixes
PPierzc committed on 1 Nov — 39466a6

Commits on Sep 28, 2021

[REFACTOR] make data_key to be inferred
PPierzc committed on 28 Sep — 7ab52da

Commits on Sep 27, 2021

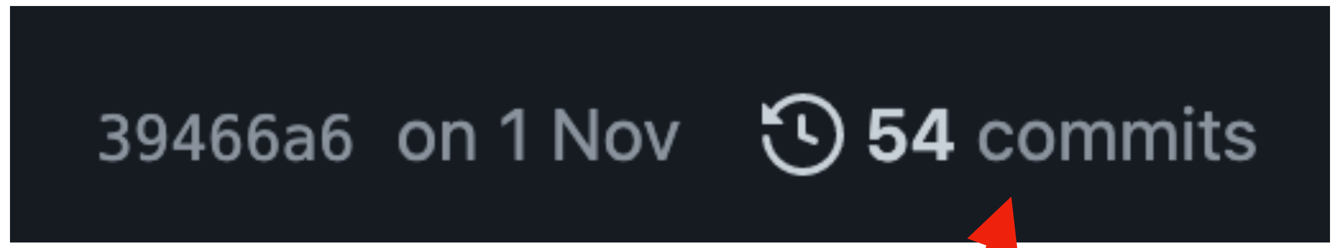
[ADD] Add static loader
PPierzc committed on 27 Sep — c3248a5

[ADD] Switch armas and legs transform and dataset loading example not... ···
PPierzc committed on 27 Sep — 09b5861

[REFACTOR] preprocessing and transforms to use the preprocessing library
PPierzc committed on 27 Sep — 9173e47

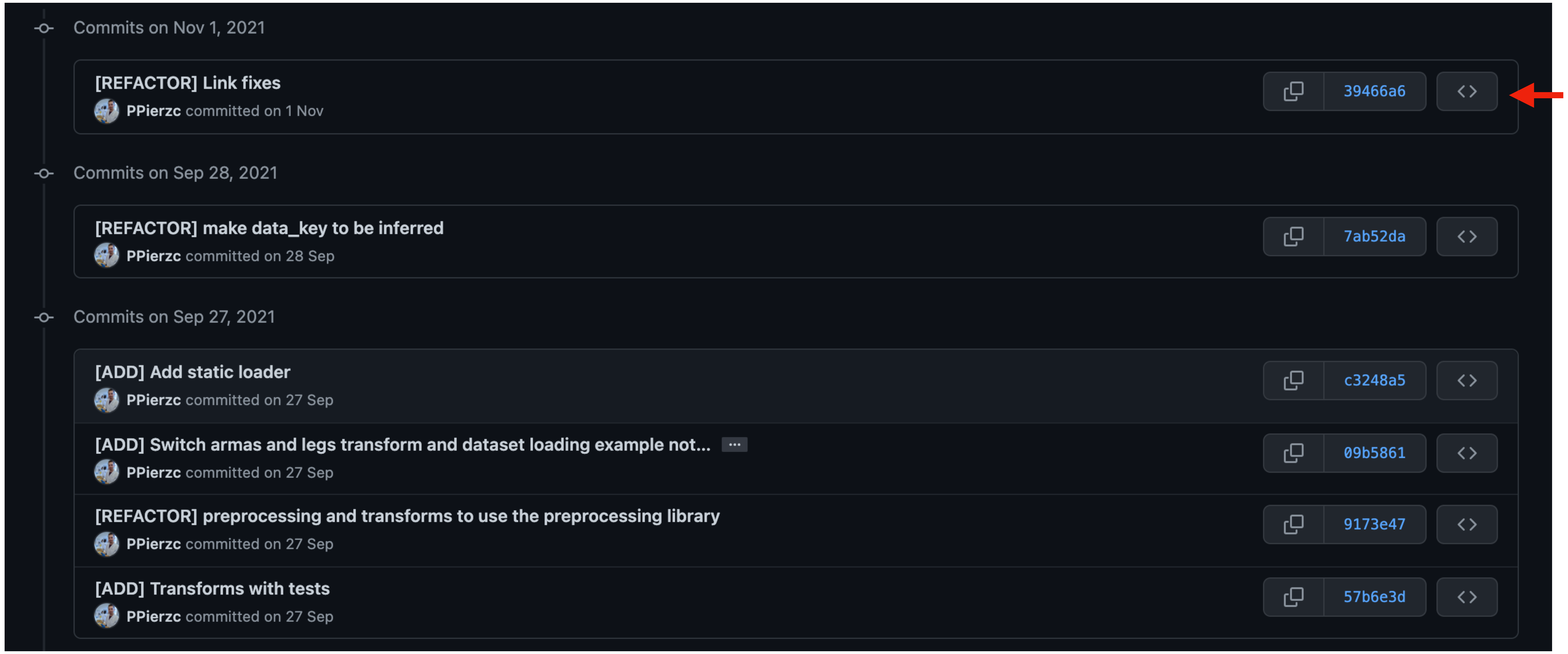
[ADD] Transforms with tests
PPierzc committed on 27 Sep — 57b6e3d

You can check the history on Github



39466a6 on 1 Nov

54 commits



Commits on Nov 1, 2021

[REFACTOR] Link fixes
PPierzc committed on 1 Nov
39466a6

Commits on Sep 28, 2021

[REFACTOR] make data_key to be inferred
PPierzc committed on 28 Sep
7ab52da

Commits on Sep 27, 2021

[ADD] Add static loader
PPierzc committed on 27 Sep
c3248a5

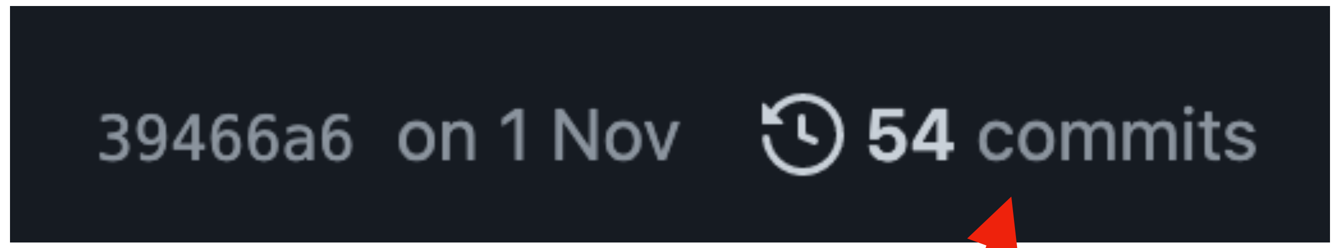
[ADD] Switch armas and legs transform and dataset loading example not… …
PPierzc committed on 27 Sep
09b5861

[REFACTOR] preprocessing and transforms to use the preprocessing library
PPierzc committed on 27 Sep
9173e47

[ADD] Transforms with tests
PPierzc committed on 27 Sep
57b6e3d

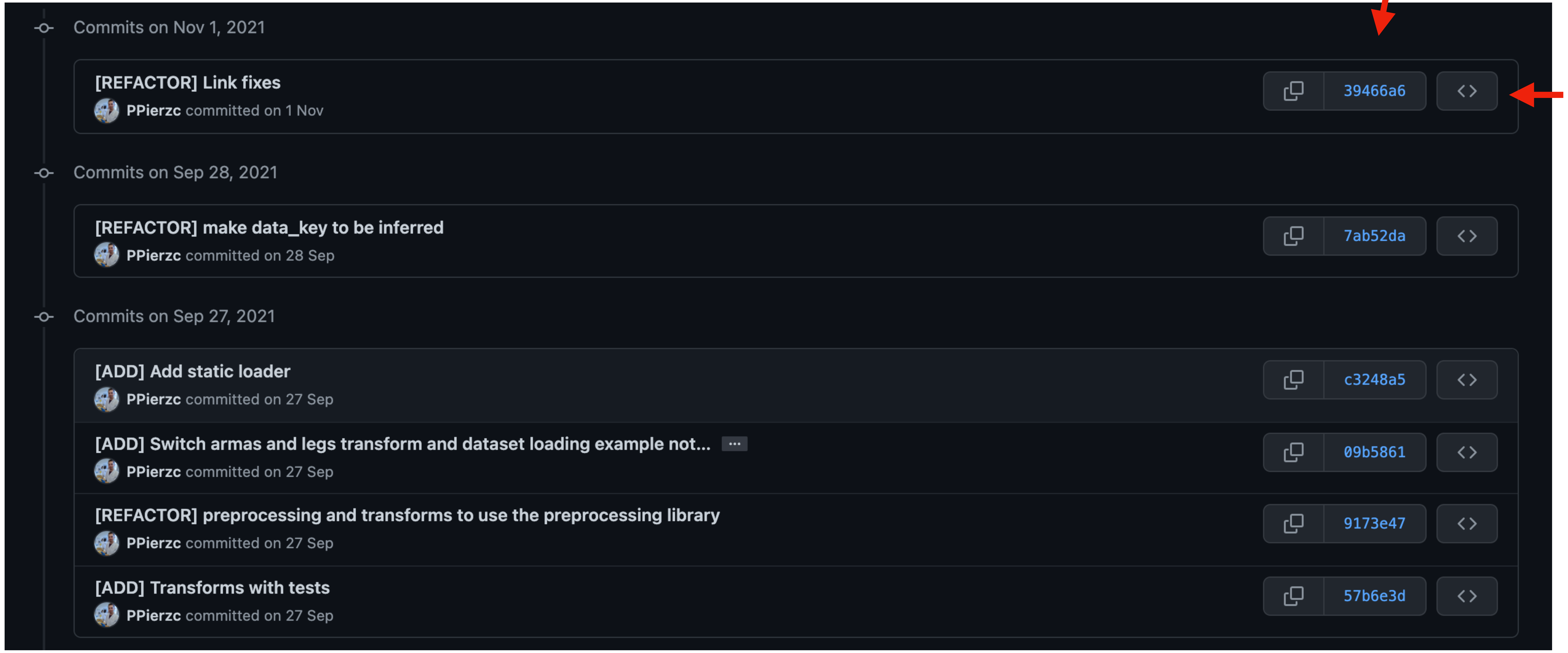
Explore repository at the state of this commit

You can check the history on Github



39466a6 on 1 Nov 54 commits

Explore the changes made



Commits on Nov 1, 2021

[REFACTOR] Link fixes
PPierzc committed on 1 Nov — 39466a6

Commits on Sep 28, 2021

[REFACTOR] make data_key to be inferred
PPierzc committed on 28 Sep — 7ab52da

Commits on Sep 27, 2021

[ADD] Add static loader
PPierzc committed on 27 Sep — c3248a5

[ADD] Switch armas and legs transform and dataset loading example not... …
PPierzc committed on 27 Sep — 09b5861

[REFACTOR] preprocessing and transforms to use the preprocessing library
PPierzc committed on 27 Sep — 9173e47

[ADD] Transforms with tests
PPierzc committed on 27 Sep — 57b6e3d

Explore repository at the state of this commit

Let’s talk about commit messages

Conventional Commits

A specification for adding human and machine readable meaning to commit messages

https://www.conventionalcommits.org/en/v1.0.0/

Let’s talk about commit messages

Conventional Commits

A specification for adding human and machine readable meaning to commit messages

https://www.conventionalcommits.org/en/v1.0.0/

Suggested message structure

```
<type>[optional scope]: <description>

[optional body]

[optional footer(s)]
```

Let’s talk about commit messages

Conventional Commits

A specification for adding human and machine readable meaning to commit messages

https://www.conventionalcommits.org/en/v1.0.0/

Suggested message structure

Title of commit →

```
<type>[optional scope]: <description>

[optional body]

[optional footer(s)]
```

Let’s talk about commit messages

Conventional Commits

A specification for adding human and machine readable meaning to commit messages

https://www.conventionalcommits.org/en/v1.0.0/

Suggested message structure

Title of commit →

```
<type>[optional scope]: <description>

[optional body]

[optional footer(s)]
```

types:

- fix
- feat
- BREAKING CHANGE
- refactor
- test

Let’s talk about commit messages

Conventional Commits

A specification for adding human and machine readable meaning to commit messages

https://www.conventionalcommits.org/en/v1.0.0/

Suggested message structure

Title of commit →

```
<type>[optional scope]: <description>

[optional body]

[optional footer(s)]
```

types:
- fix
- feat
- BREAKING CHANGE
- refactor
- test

Example commit with scope

```
feat(lang): add polish language button

Adds Polish language button to
switch between English and Polish language

refs: #100
```

How often should I commit?

How often should I commit?

Short Answer: Whenever you have something that works

How often should I commit?

Short Answer: Whenever you have something that works

Don’t be afraid to commit often.

It is better to commit a lot of small changes than one massive change.

Makes it easier to revert.

How often should I commit?

Short Answer: Whenever you have something that works

Don’t be afraid to commit often.

It is better to commit a lot of small changes than one massive change.

Makes it easier to revert.

How to control yourself?

How often should I commit?

Short Answer: Whenever you have something that works

Don’t be afraid to commit often.

It is better to commit a lot of small changes than one massive change.

Makes it easier to revert.

How to control yourself?

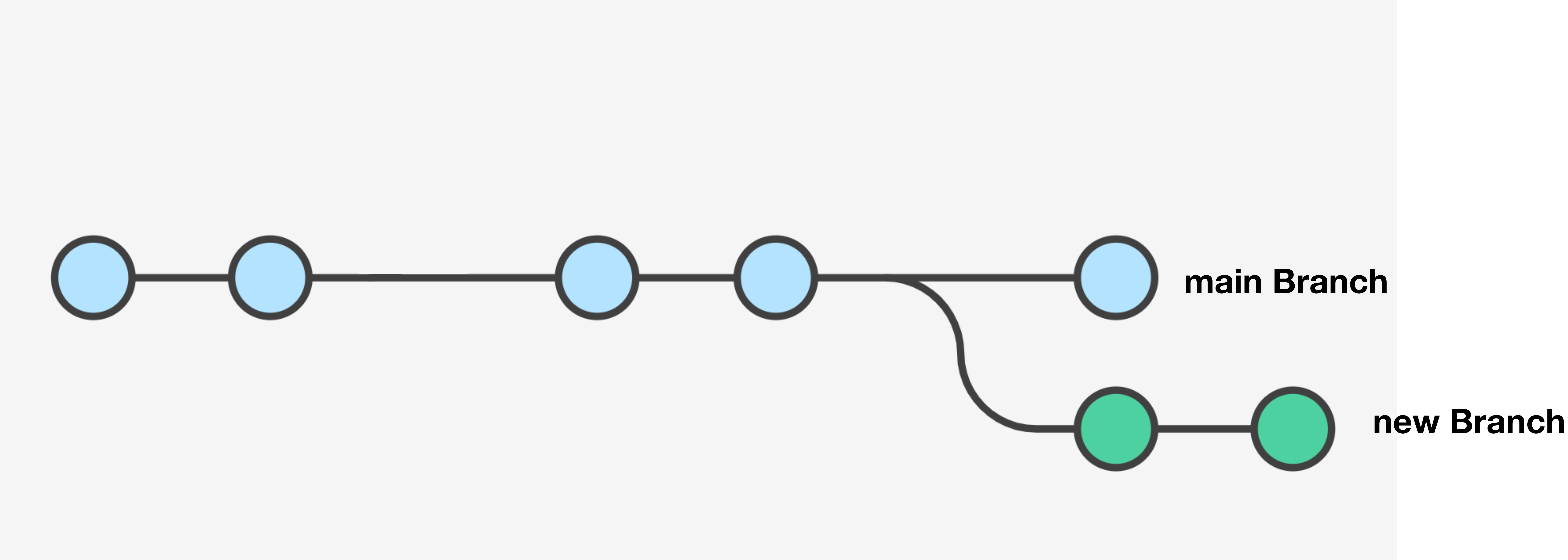
Can you still define the type of the commit you will make?

I.e. does this add a new feature or fix a bug or refactors some code etc.

If not you have gone too long without a commit!

Branching: making copies

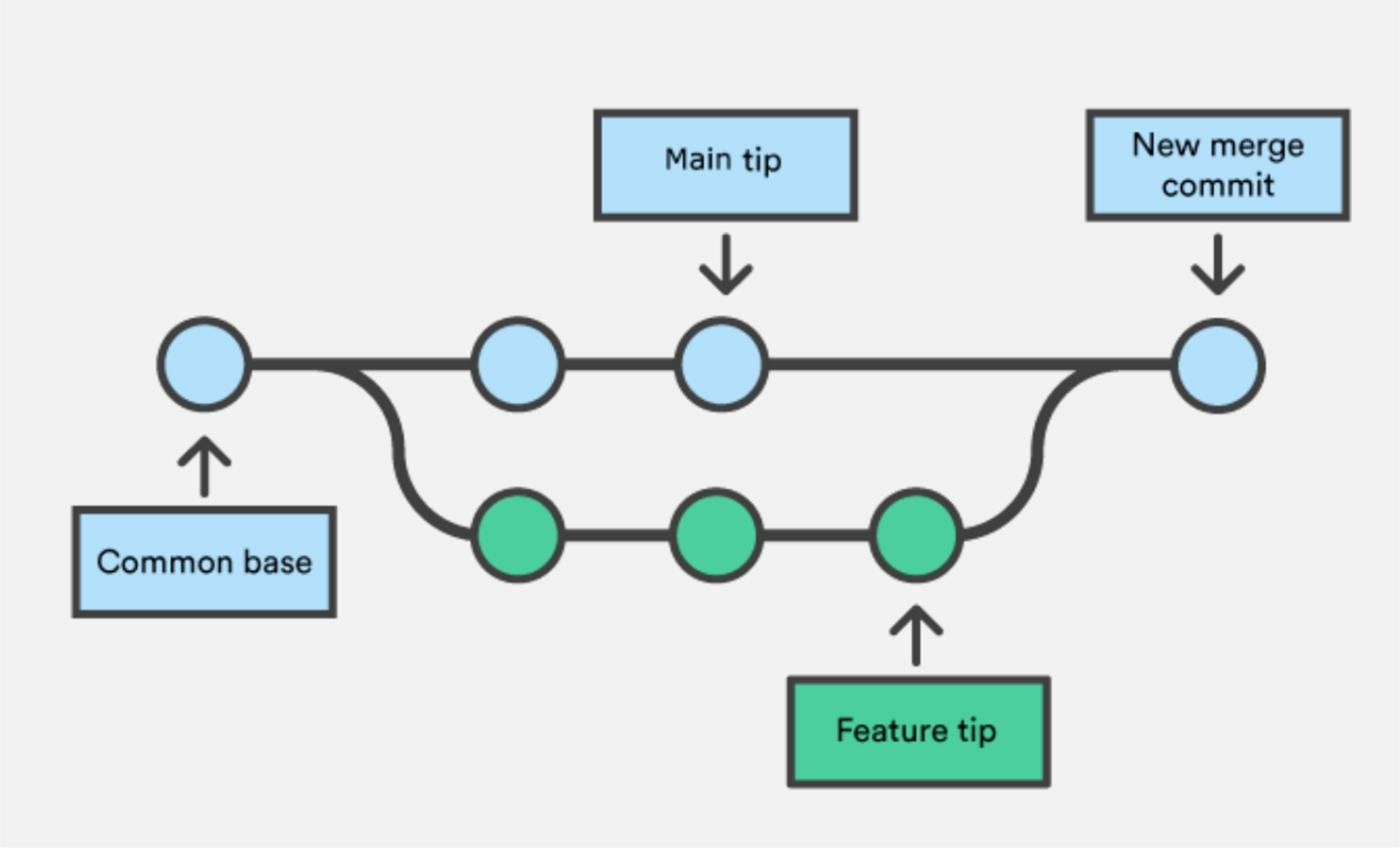
Sometimes you want to make a copy of your repository.



```
git checkout -b <branch_name>
```

Useful for developing a new feature while maintaining working code

Merging joining your work from before



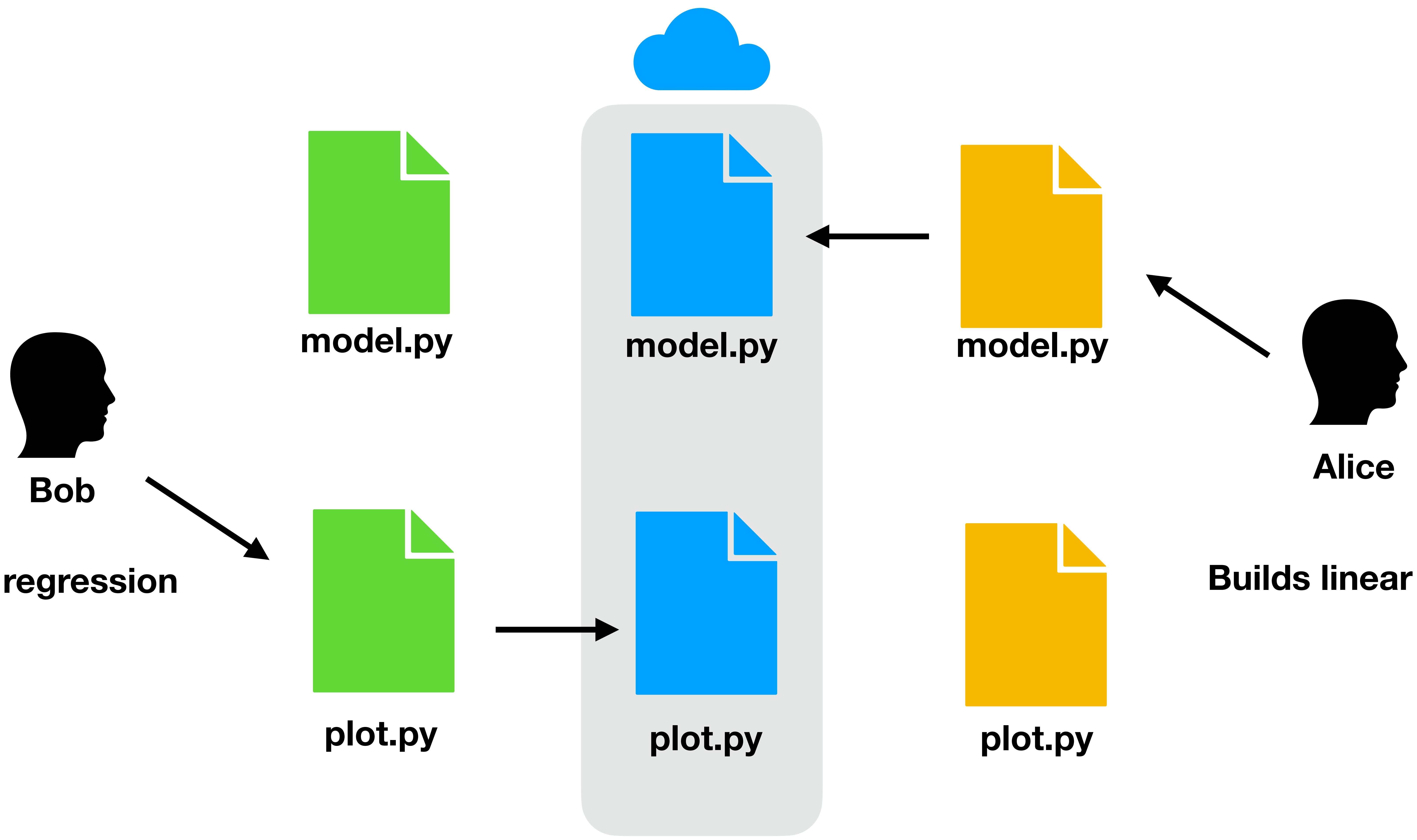
Merging two branches together

`git merge <branch_name>`

https://www.atlassian.com/git/tutorials/using-branches/git-merge

Making your project teamwork friendly

How can you get multiple people to work on the same codebase at the same time?

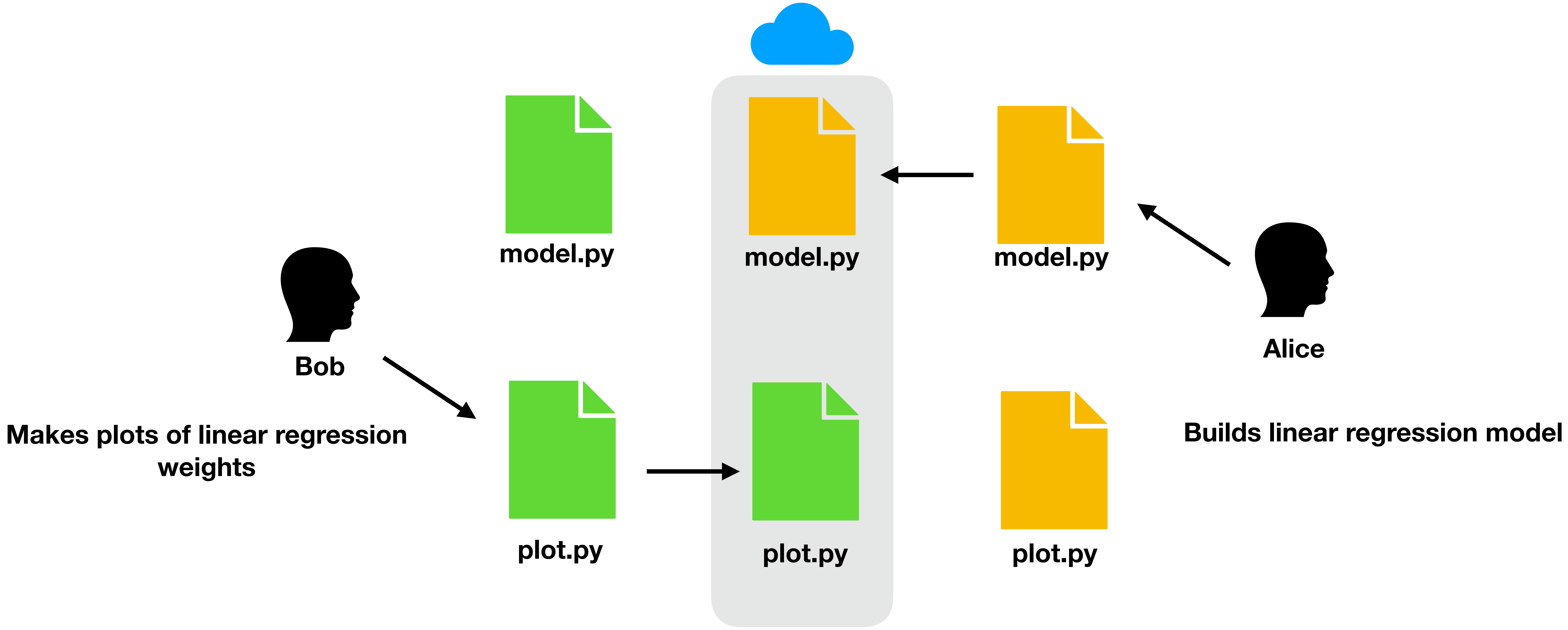


Makes plots of linear regression weights

Builds linear regression model

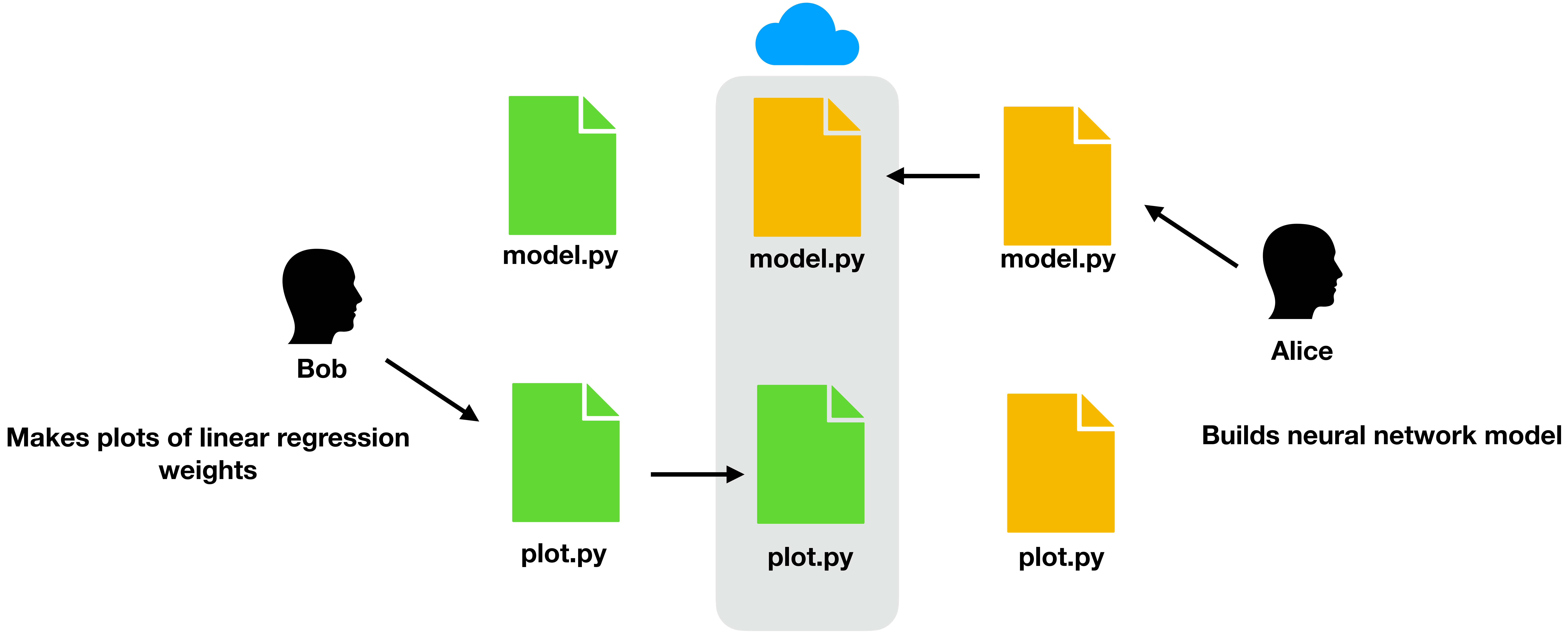
Making your project teamwork friendly

How can you get multiple people to work on the same codebase at the same time?


model.py
model.py
model.py
Bob
Alice
Makes plots of linear regression weights
plot.py
plot.py
plot.py
Builds linear regression model

Making your project teamwork friendly

How can you get multiple people to work on the same codebase at the same time?

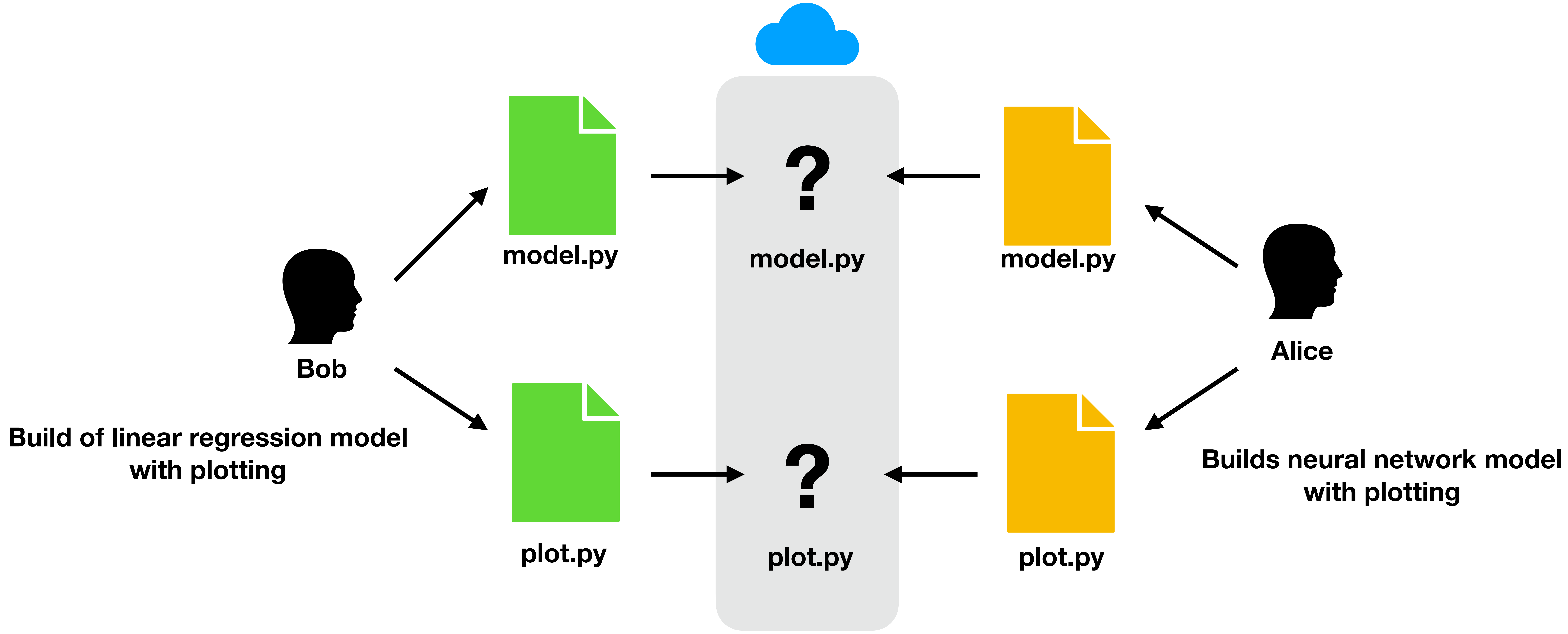


Makes plots of linear regression weights

Builds neural network model

Making your project teamwork friendly

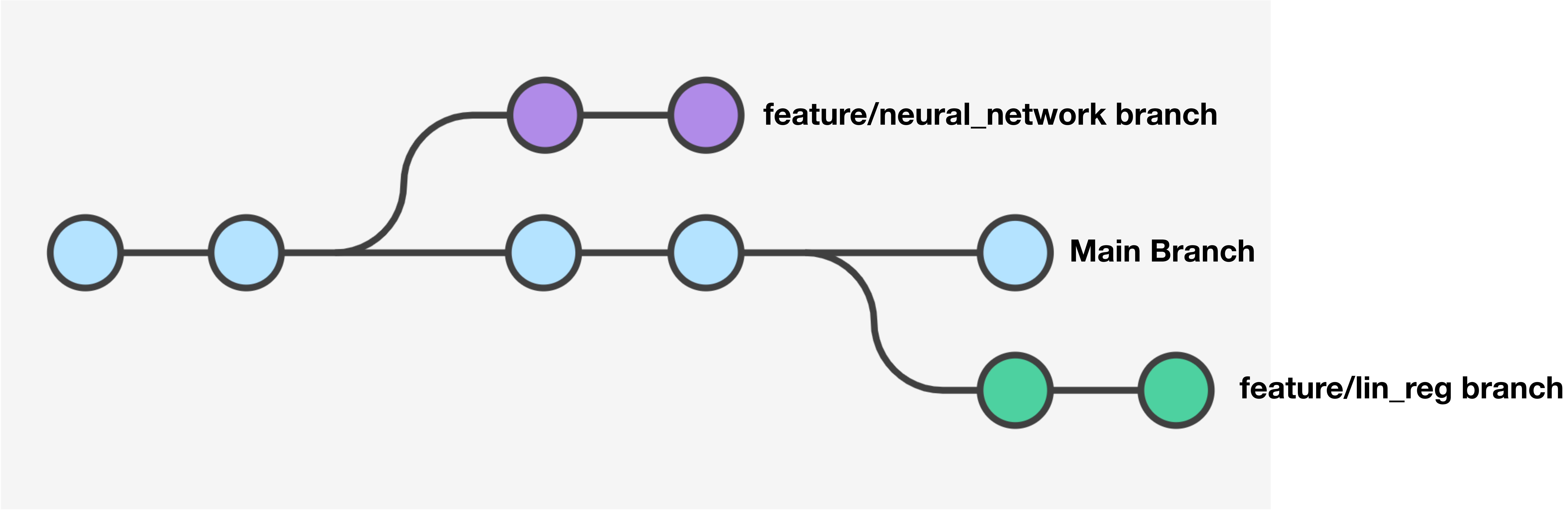
How can you get multiple people to work on the same codebase at the same time?



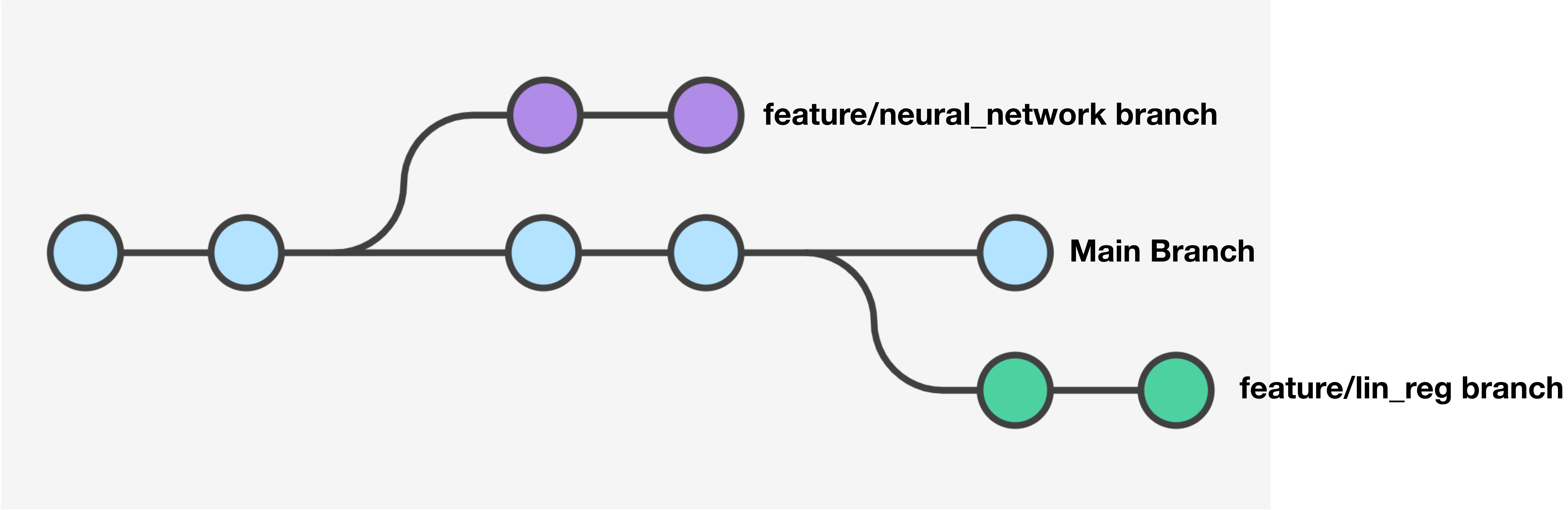
Build of linear regression model with plotting

Builds neural network model with plotting

Feature Branches

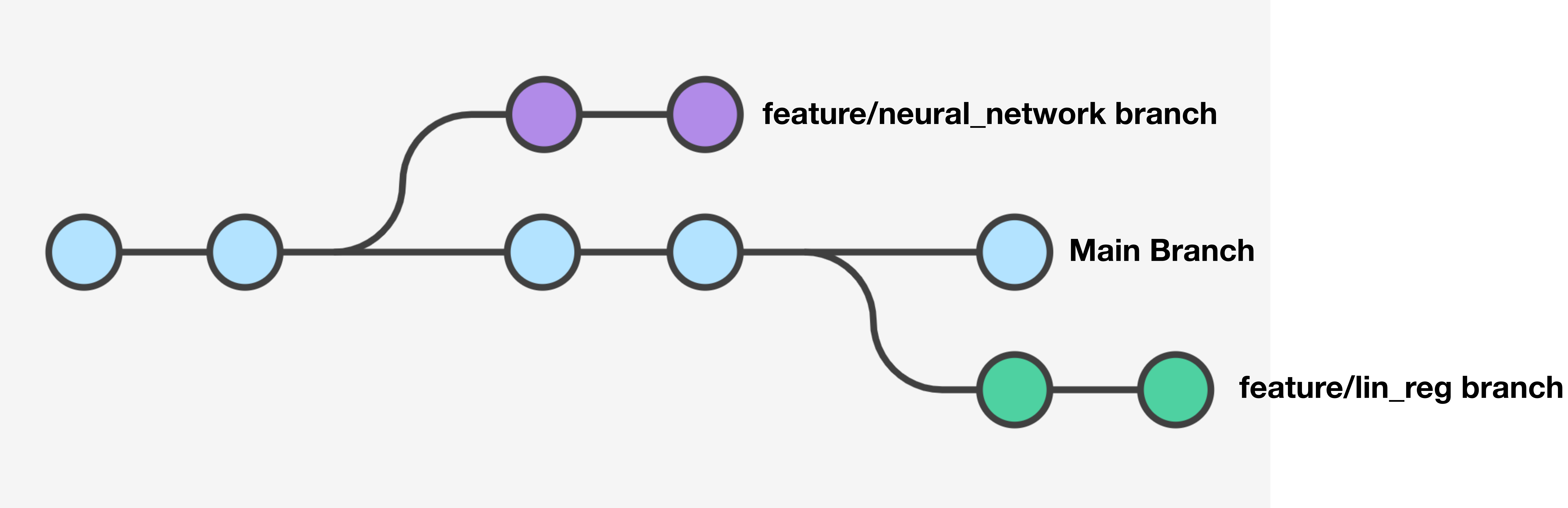


Feature Branches



This allows you to merge only the features that are ready/working to the main branch

Feature Branches



feature/neural_network branch

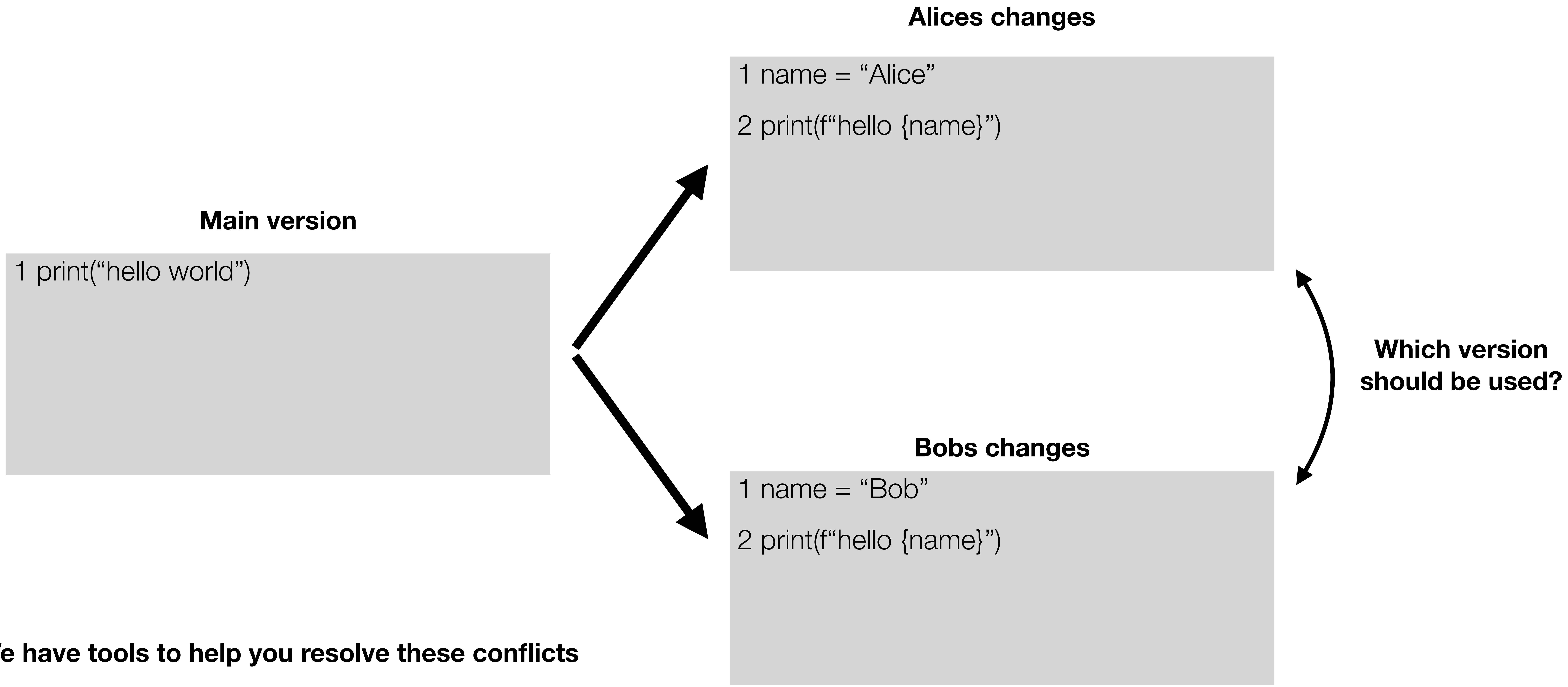
Main Branch

feature/lin_reg branch

This allows you to merge only the features that are ready/working to the main branch

Goal is to have the main branch as a stable working version and feature branches as developmental branches (not always working)

Merge conflicts



Main version

```
print(“hello world”)
```

Alices changes

```
name = “Alice”
print(f“hello {name}”)
```

Bobs changes

```
name = “Bob”
print(f“hello {name}”)
```

Which version should be used?

We have tools to help you resolve these conflicts

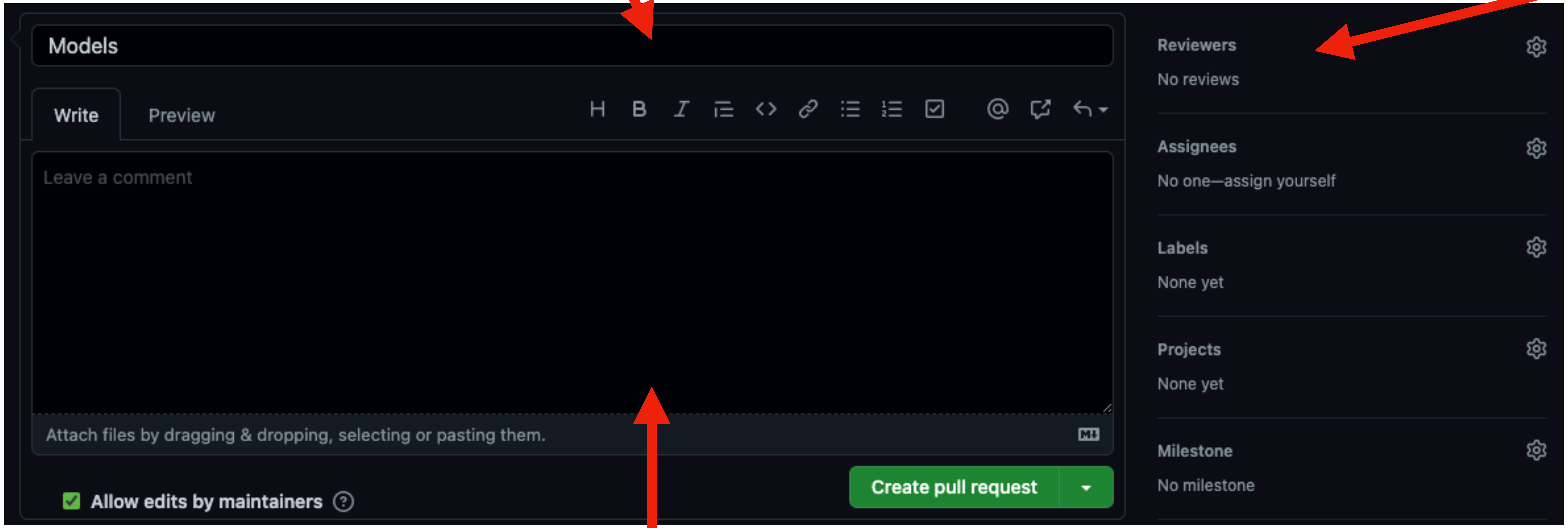
Pull Requests

Github allows for creating pull requests

You ask your collaborators for permission to merge your branch with another branch

Title of the pull request: Short and descriptive of the changes the PR is making

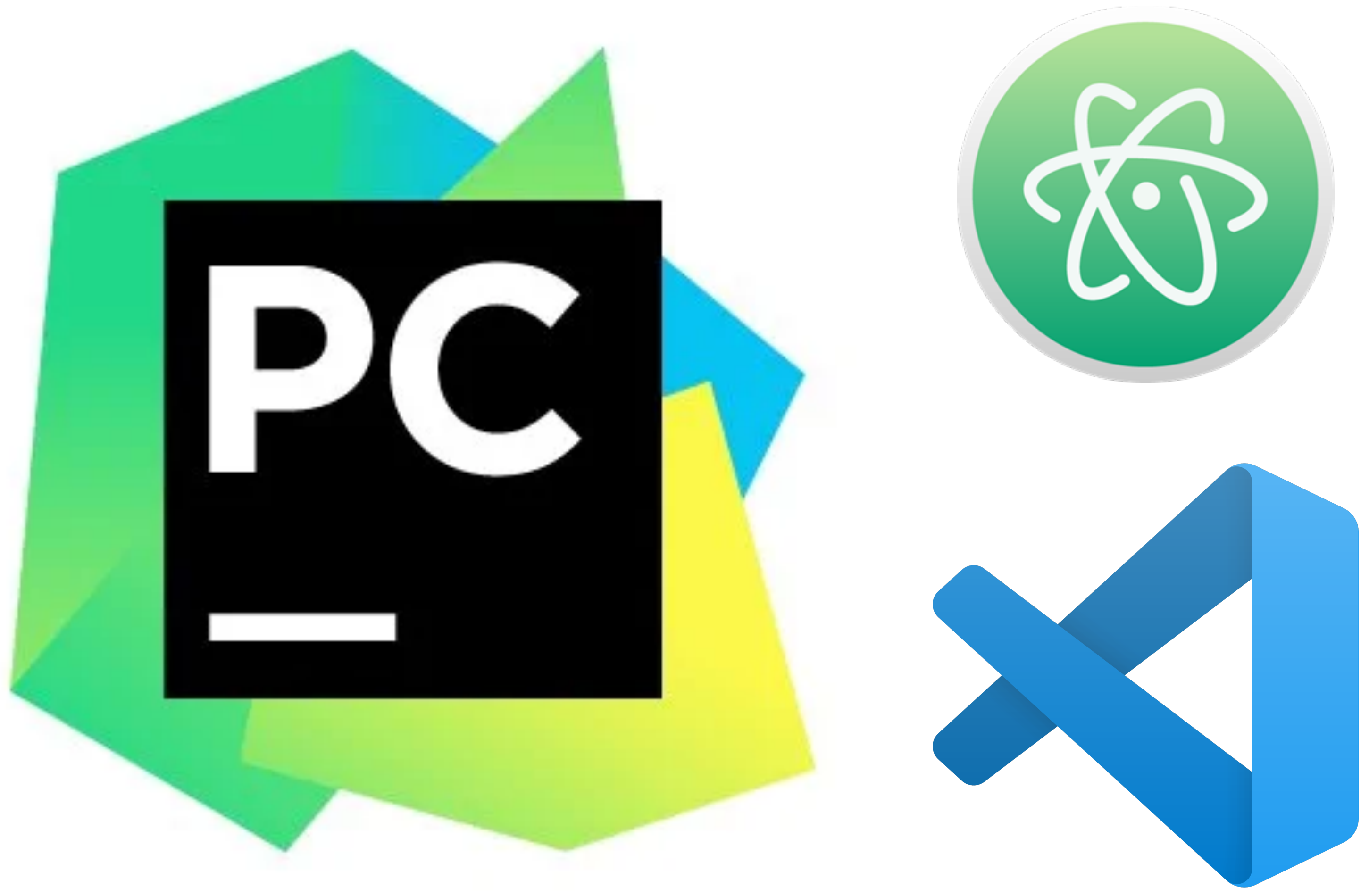
You can assign people to review and accept your pull request



Models

Description of the changes, what does this pull request add. Why should it be merged etc. (especially important for large projects)

Using Git with an IDE makes things a lot simpler



Conclusion

Conclusion

Git provides you with a solution for version control, no more copies of files!

Conclusion

Git provides you with a solution for version control, no more copies of files!

Git provides a solution for working on programming projects as a team

Conclusion

Git provides you with a solution for version control, no more copies of files!

Git provides a solution for working on programming projects as a team

Don’t be afraid to commit

Conclusion

Git provides you with a solution for version control, no more copies of files!

Git provides a solution for working on programming projects as a team

Don’t be afraid to commit

Commands worth remembering

```
git commit -m “commit message”
```

```
git push
```

Conclusion

Git provides you with a solution for version control, no more copies of files!

Git provides a solution for working on programming projects as a team

Don’t be afraid to commit

Commands worth remembering

```
git commit -m “commit message”
```

```
git push
```

https://ohshitgit.com/