WEEK ONE - B

TERMINAL AND GIT

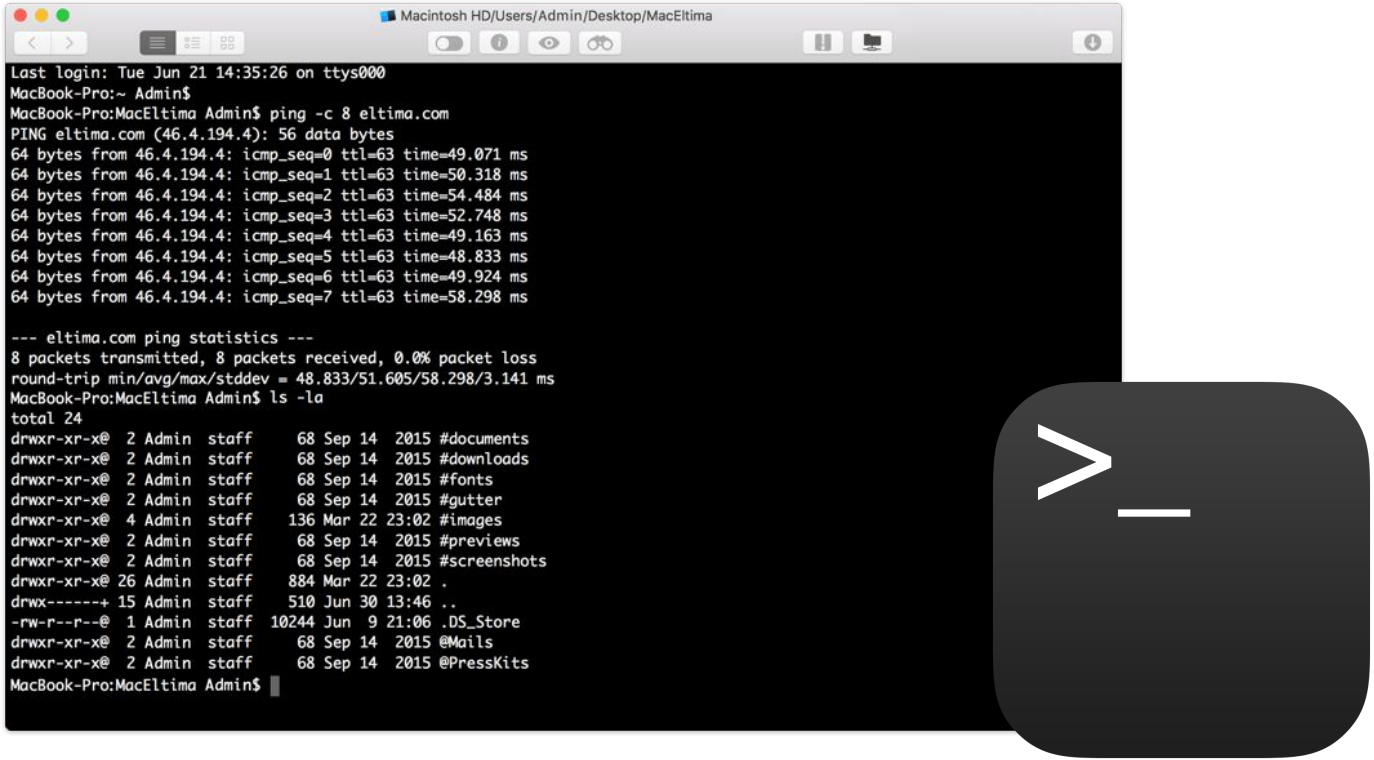
TERMINAL

What's under the hood

Terminal

What's under the hood

Terminal



What's under the hood

Terminal

Directories ?

- change directory (cd)
- print working directory (pwd)
- list files and directories inside current working directory (ls)

What's under the hood

Terminal

Files ?

- Create file: touch <filename>
- Copy file: cp <filenameToCopy> <copiedName>
- Delete file: rm <filename>

What's under the hood

Terminal

Managing Directory ?

- Create directory: mkdir <directoryName>
- Delete directory: rm -rf <directoryName>

What's under the hood

Terminal

GIT

Real World Issue

Collaboration

Real World Issue

Collaboration

ISSUES

- Conflicts and overwritten files
- Revision issues

SOLUTION ?

Real World Issue

Collaboration

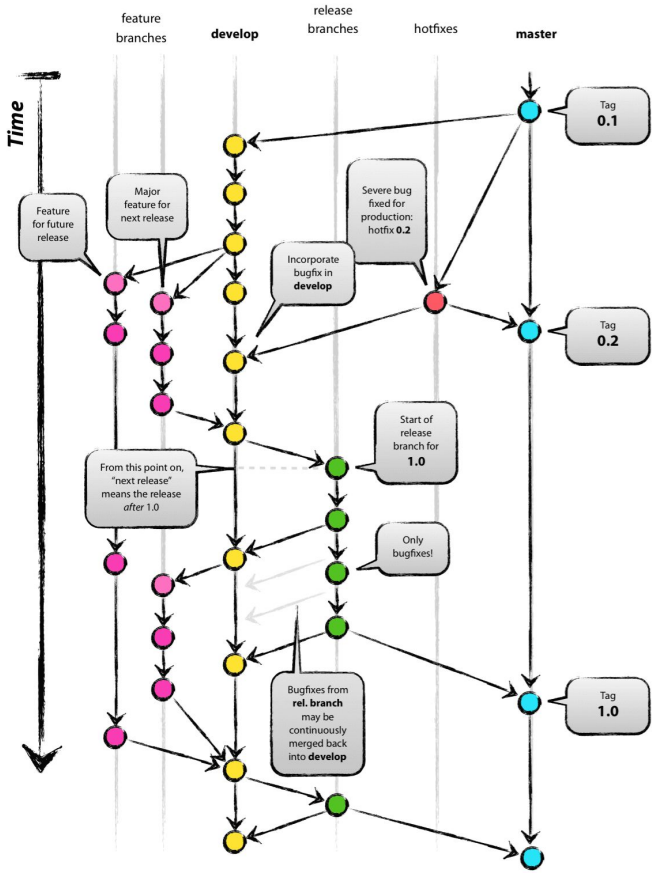
Real World Issue

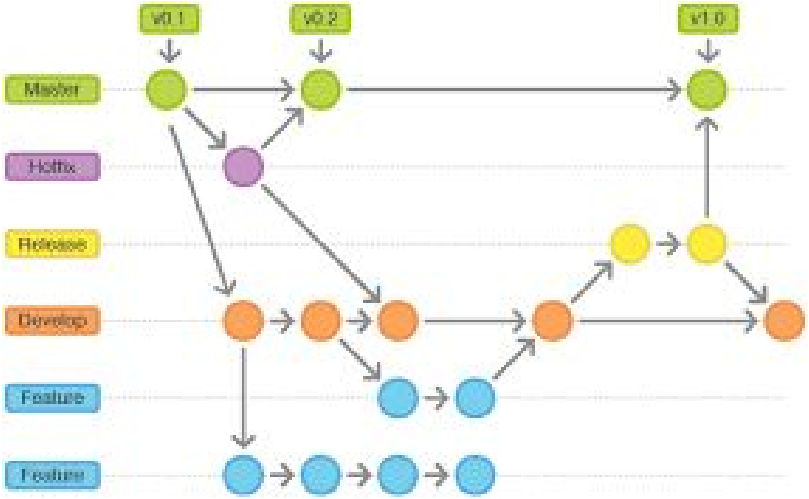
Collaboration

git

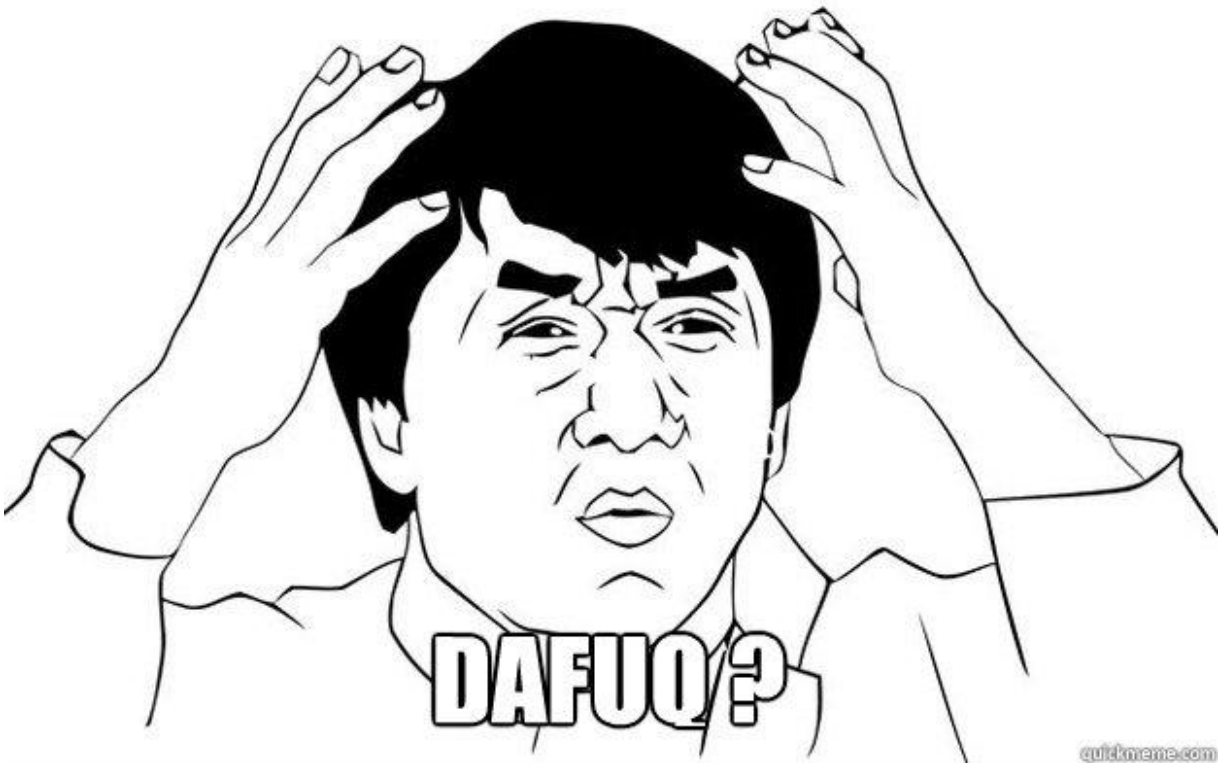
Git = version control system

What is Git


feature branches
develop
release branches
hotfixes
master
Time
Tag 0.1
Tag 0.2
Tag 1.0
Feature for future release
Major feature for next release
Severe bug fixed for production: hotfix 0.2
Incorporate bugfix in develop
Start of release branch for 1.0
From this point on, "next release" means the release after 1.0
Only bugfixes!
Bugfixes from rel. branch may be continuously merged back into develop

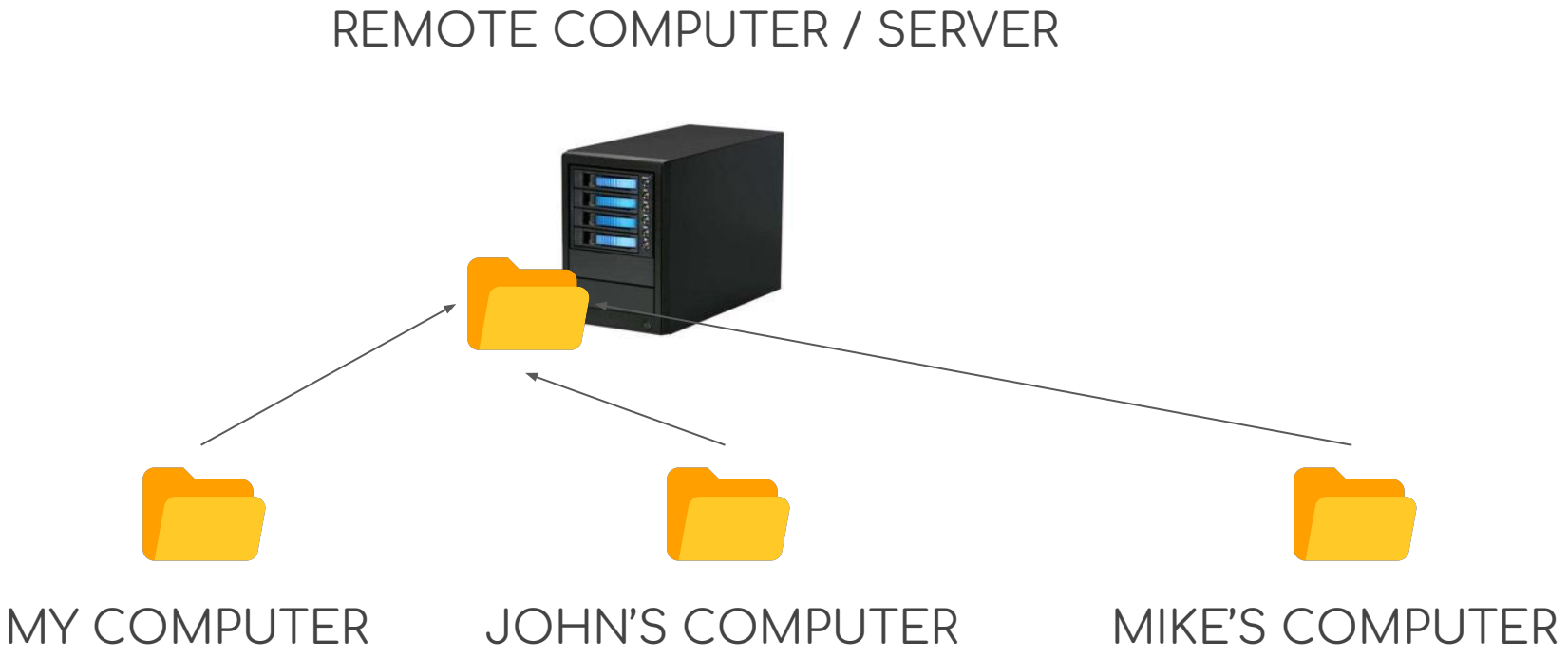

v0.1
v0.2
v1.0
Master
Hotfix
Release
Develop
Feature
Feature

What is
Git



What is
Git

EVERYONE SHOULD SYNC TO THE REMOTE SERVER!



What is
Git

Git Repository

Folders and Files

History

What is Git

Git tracks every file changes.

Your changes, John's changes, everyone!

What is
Git

Git can undo to some “points”

We call it as **Commit**

Commit = the record of changes

GITHUB

What is
Git

IT IS QUITE COMPLICATED TO SETUP A GIT SERVER !

REMOTE COMPUTER / SERVER

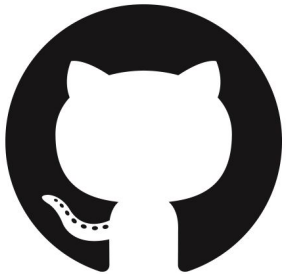
What is
Git

WE NEED A SERVICE TO BE THE SERVER!

REMOTE COMPUTER / SERVER

What is
Git

HELLO GITHUB!



Github = git hosting service

HOW GIT WORKS

The Flow of
Git

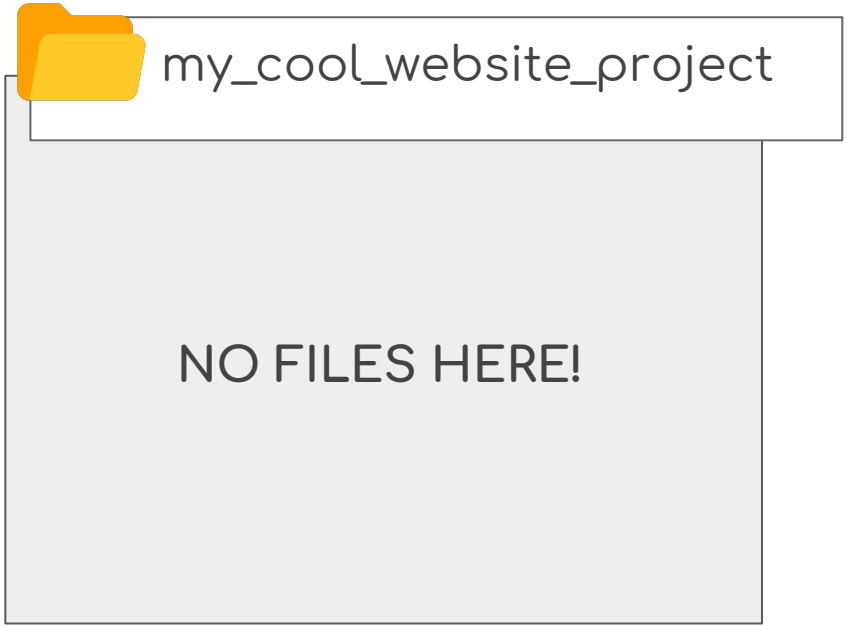
Create empty repository in github

Go to
github.com
and create a new repository

The Flow of
Git

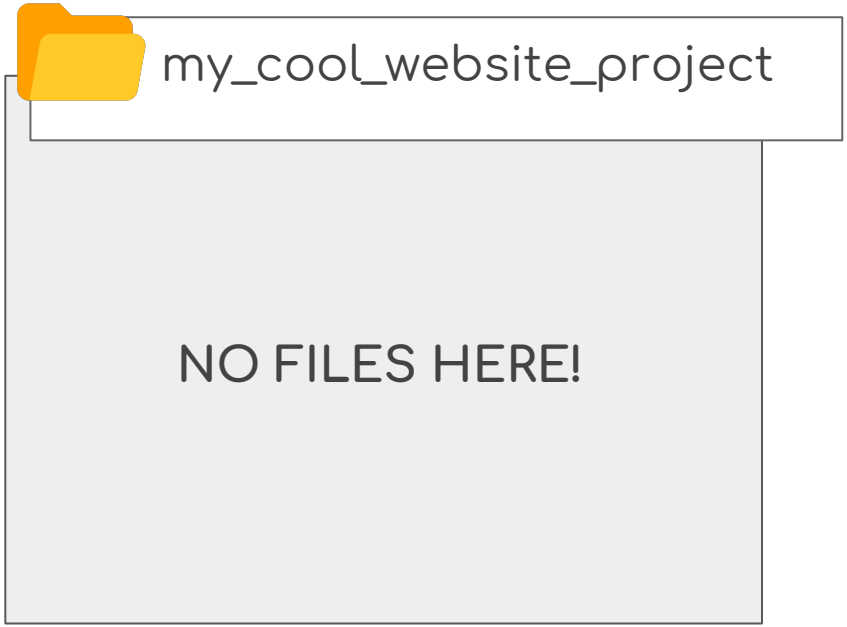
Prepare the Project Folder

MY COMPUTER



The Flow of
Git

MY COMPUTER



Initialize the git using

TERMINAL: $ git init

What is Git

Your Folder is now a Git Repository!

What is
Git

BUT still no connection to the remote repository!

The Flow of
Git

TERMINAL: $ git remote add origin <HTTPS LINK TO GITHUB REPO>

MY COMPUTER

my_cool_website_project

NO FILES HERE!

https://github.com/super-cool-repository

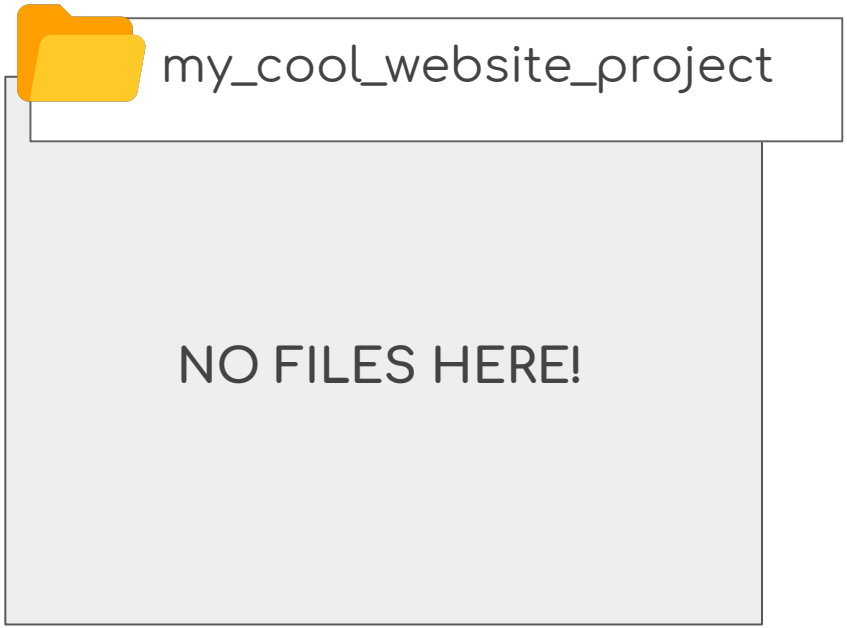
What is
Git

Yay... your local git is now connected to remote repository!

Now let's focus on your local computer..

The Flow of
Git

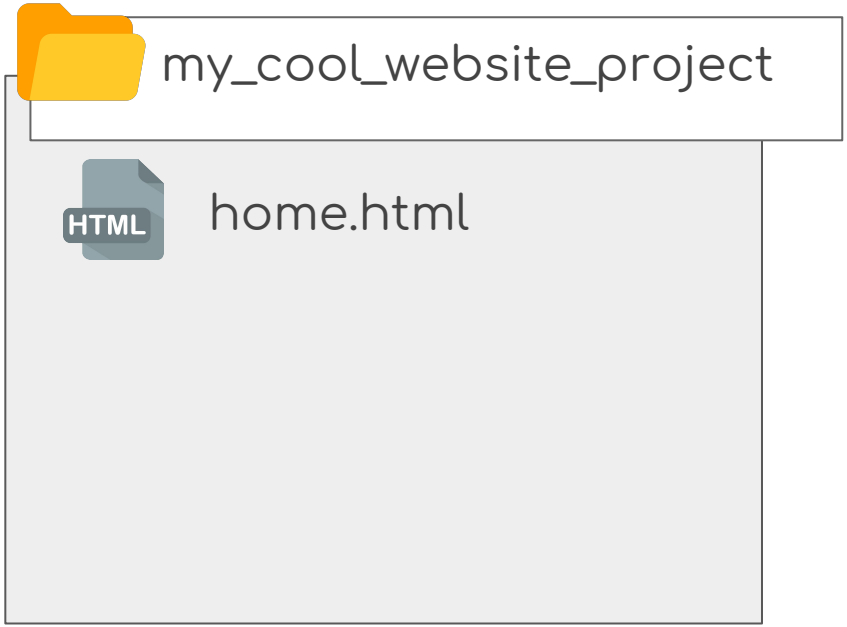
MY COMPUTER



The Flow of
Git

Add new file called **home.html**

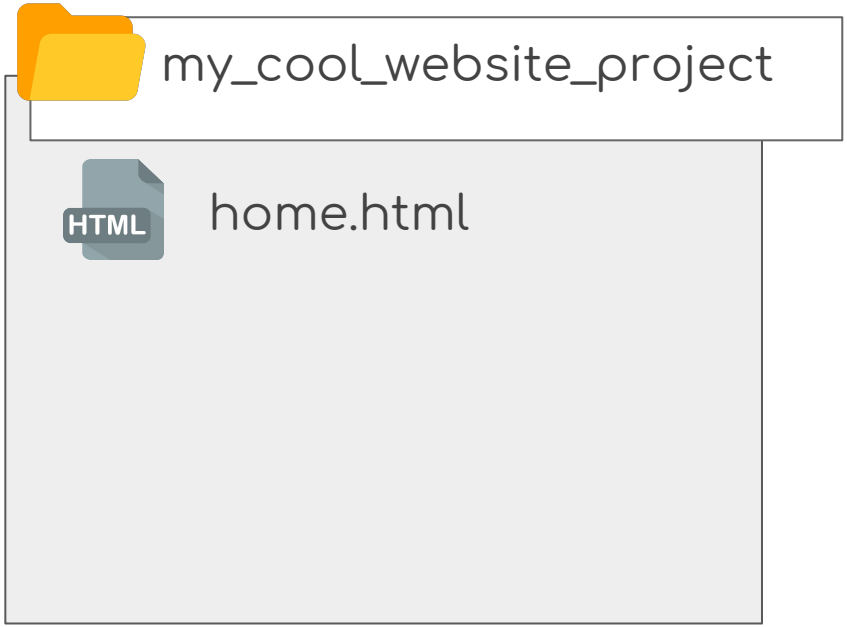
MY COMPUTER



The Flow of
Git

At this point, **home.html** is UNSTAGED

MY COMPUTER



What is
Git

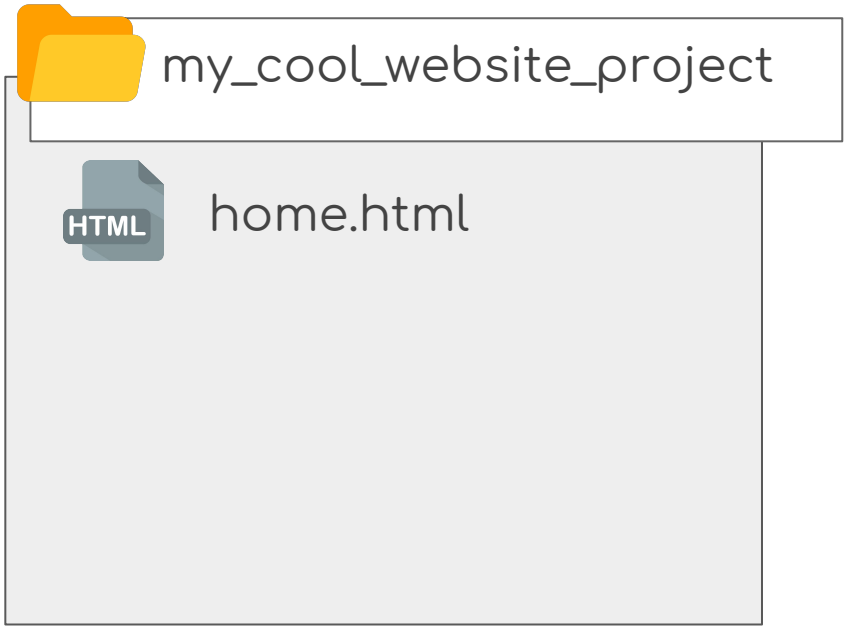
UNSTAGED ?

All changes in **MY COMPUTER**
But not yet recorded in Git

The Flow of
Git

TERMINAL: $ git add **home.html**

MY COMPUTER



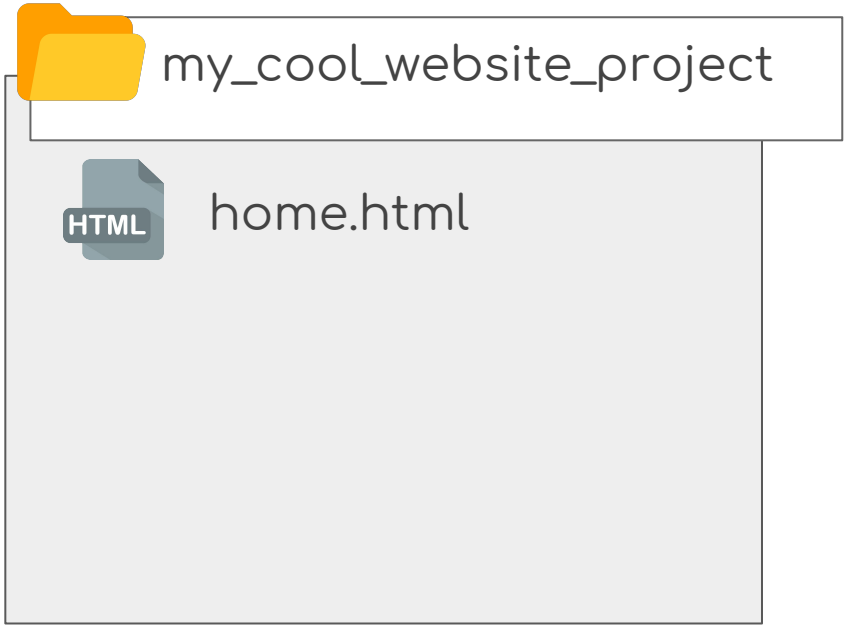
What is Git

Home.html is now staged !

The Flow of
Git

At this point, **home.html** is STAGED

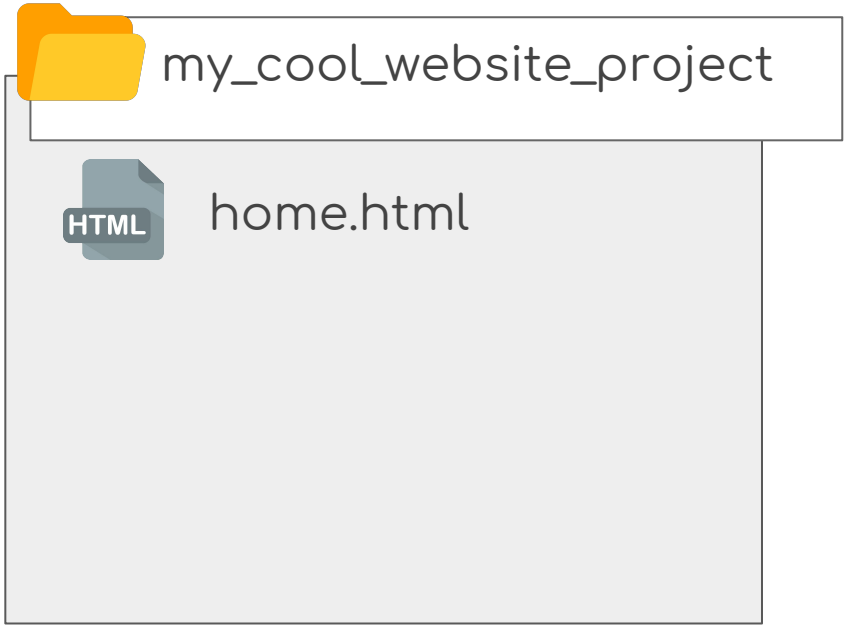
MY COMPUTER



The Flow of
Git

Commit the **home.html** file and set the **commit message**

MY COMPUTER



The Flow of
Git

COMMIT ?

Record some changes in **MY COMPUTER** to Git

The Flow of
Git

COMMIT MESSAGE ?

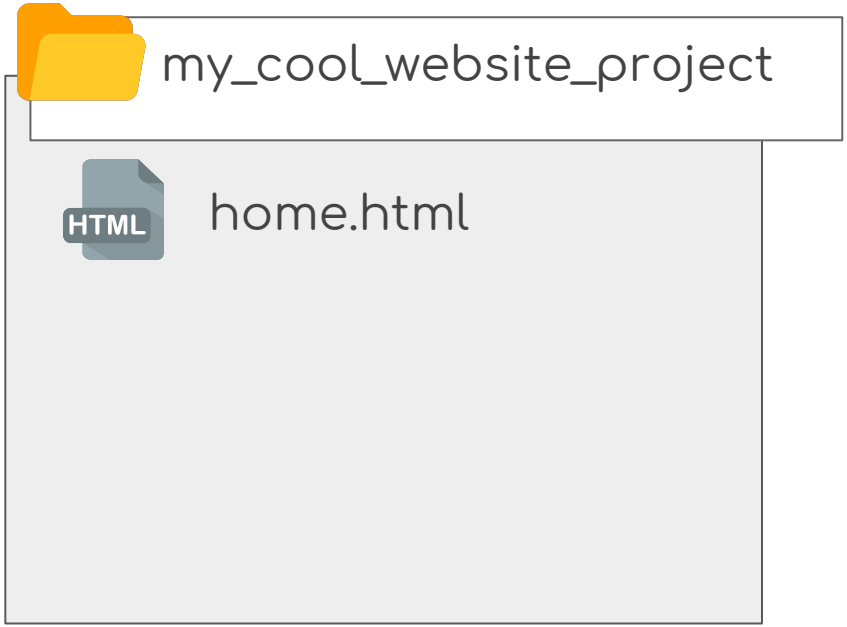
When I record something,
I need to name / label them!

Example: “Add Home Page“

The Flow of
Git

TERMINAL: $ git commit -m “Add home page”

MY COMPUTER



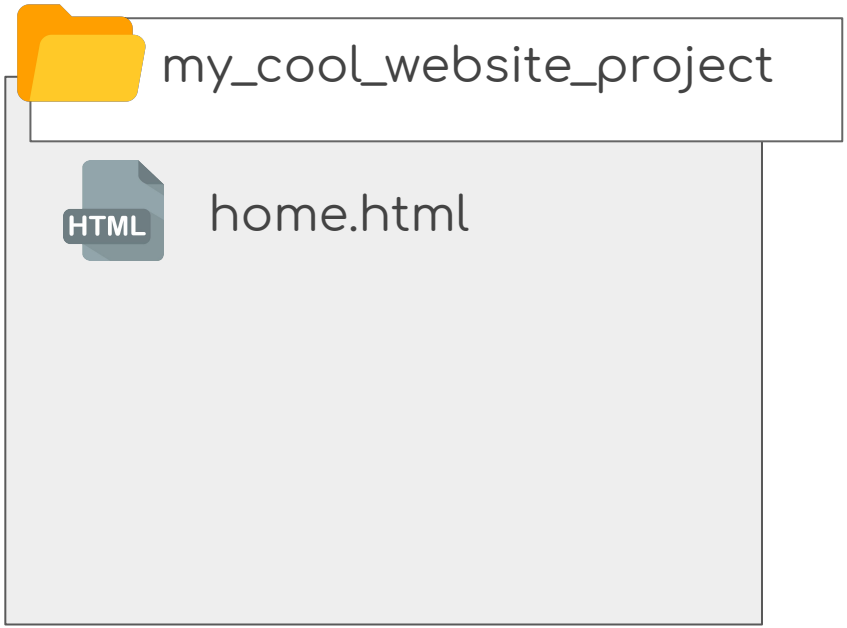
What is
Git

Home.html is now committed !

The Flow of
Git

home.html is now **staged** and committed

MY COMPUTER



The Flow of
Git

STAGED ?

Recorded changes of files in
MY COMPUTER'S GIT

The Flow of
Git

Changes committed and I guess my work's done (for now).

So what now?

The Flow of
Git

Time to sync them to the server!
(read: Remote Repository)

The Flow of
Git

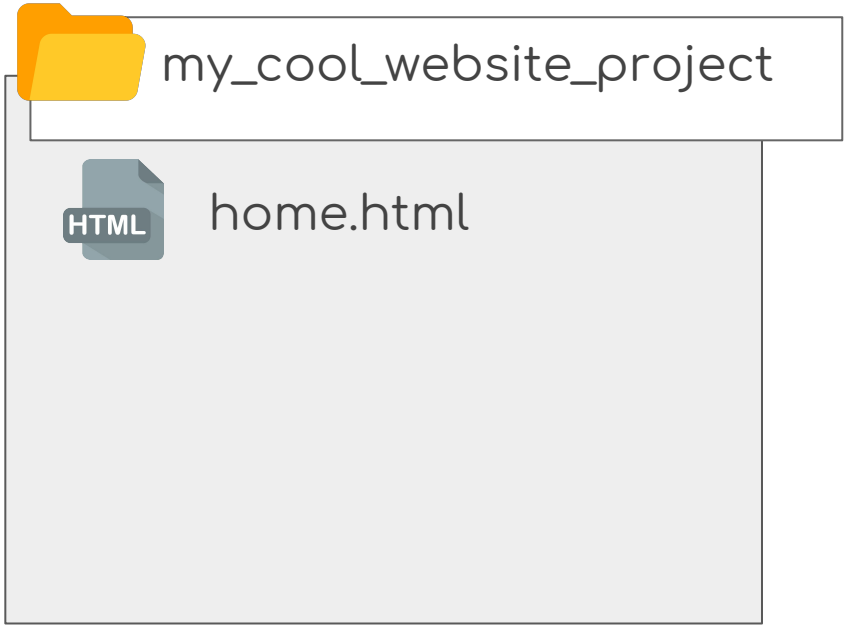
PUSH

"Push" changes in **My Computer** to Remote Repository, and automatically sync them with the remote server

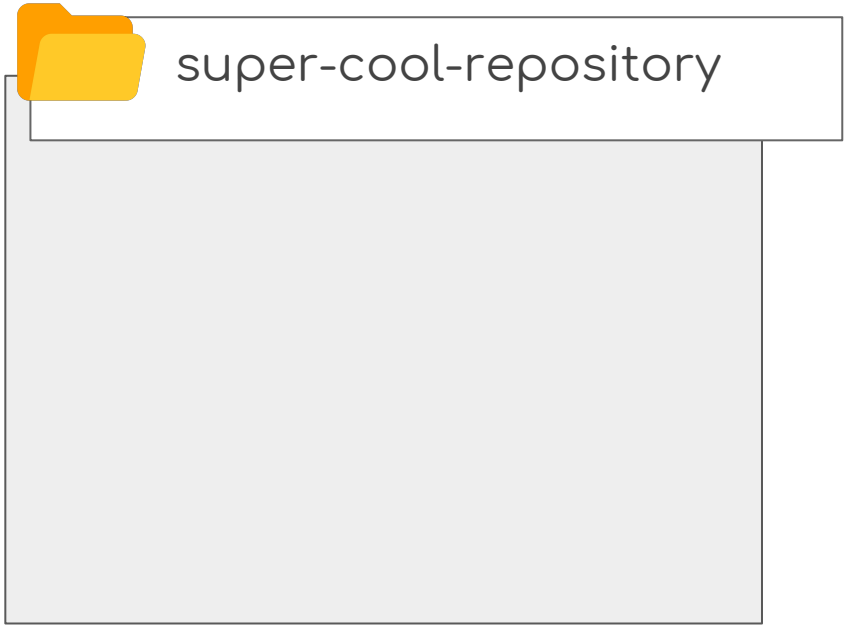
The Flow of
Git

Before push

MY COMPUTER



REMOTE REPO



The Flow of
Git

After push

MY COMPUTER

my_cool_website_project

HTML home.html

REMOTE REPO

super-cool-repository

HTML home.html

The Flow of
Git

Home.html is now in sync with remote repository !

Repo is up to date!

Let's

Review!

UNSTAGED

STAGED

UP-TO-DATE!

Let's

Review!

UNSTAGED

Add files to staged them!

Let's

Review!

STAGED

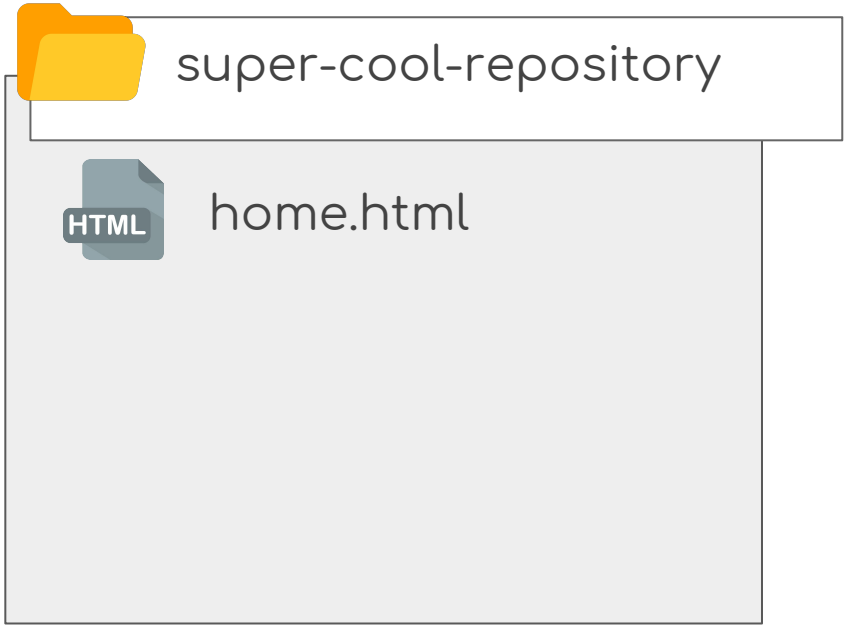
Commit staged files
Set the commit message


LET'S DO A
SHORT BREAK !

The Flow of
Git

First Time in John's Computer!

REMOTE REPO



JOHN'S COMPUTER

The Flow of
Git

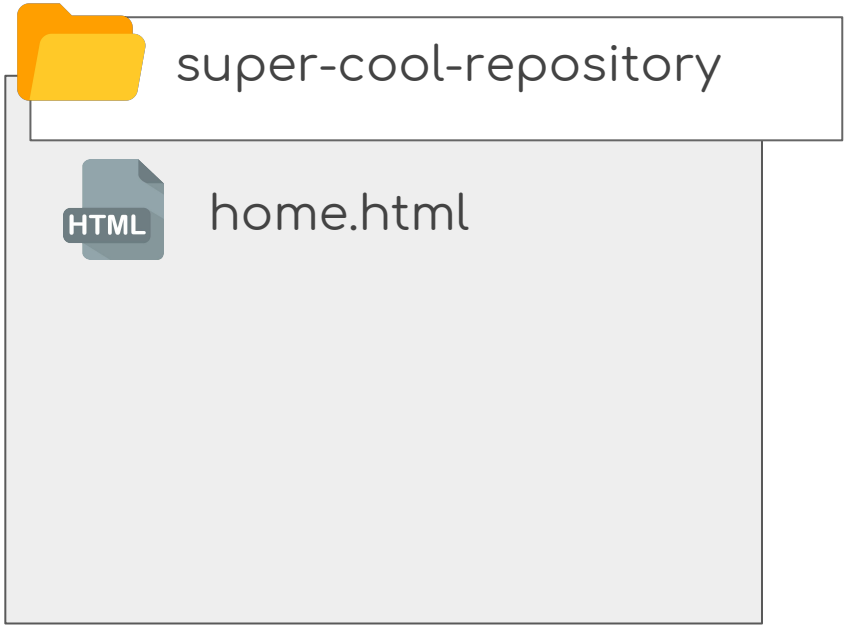
CLONING

"Copy" the existing project from remote repository

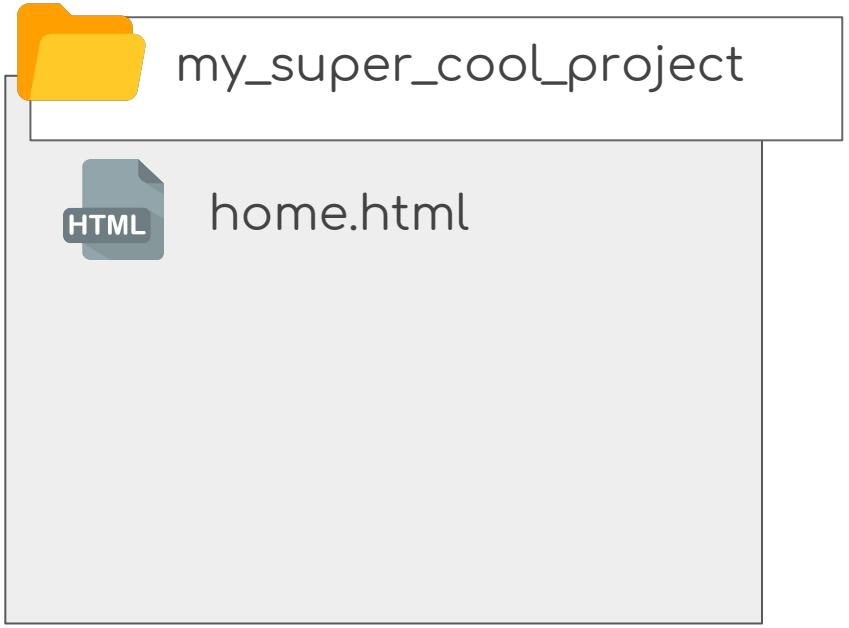
The Flow of
Git

After Clone

REMOTE REPO



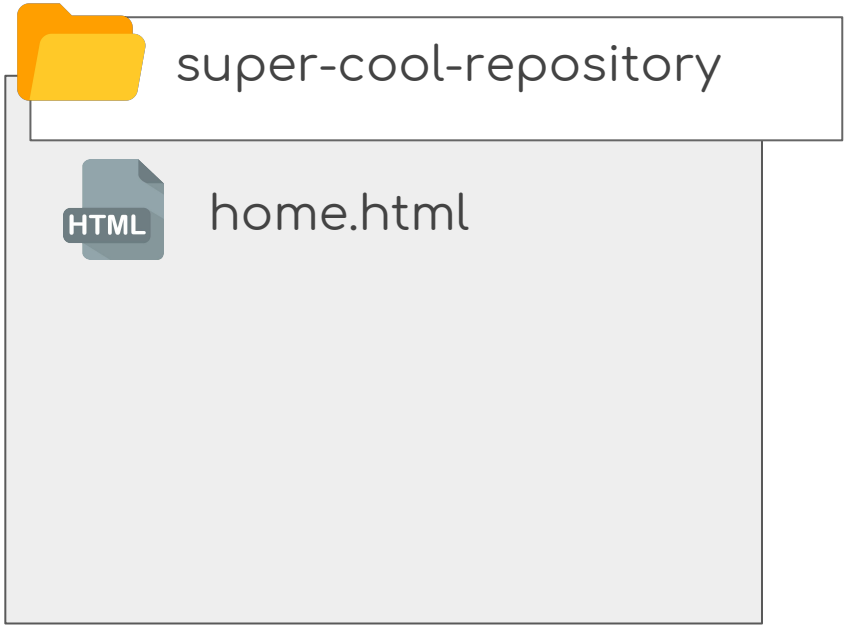
JOHN'S COMPUTER



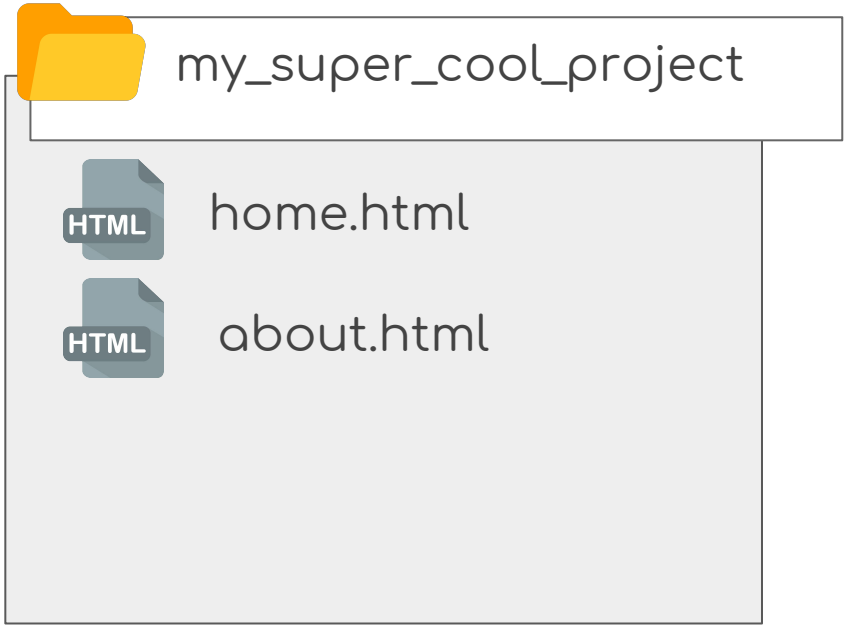
The Flow of
Git

John add about.html !

REMOTE REPO

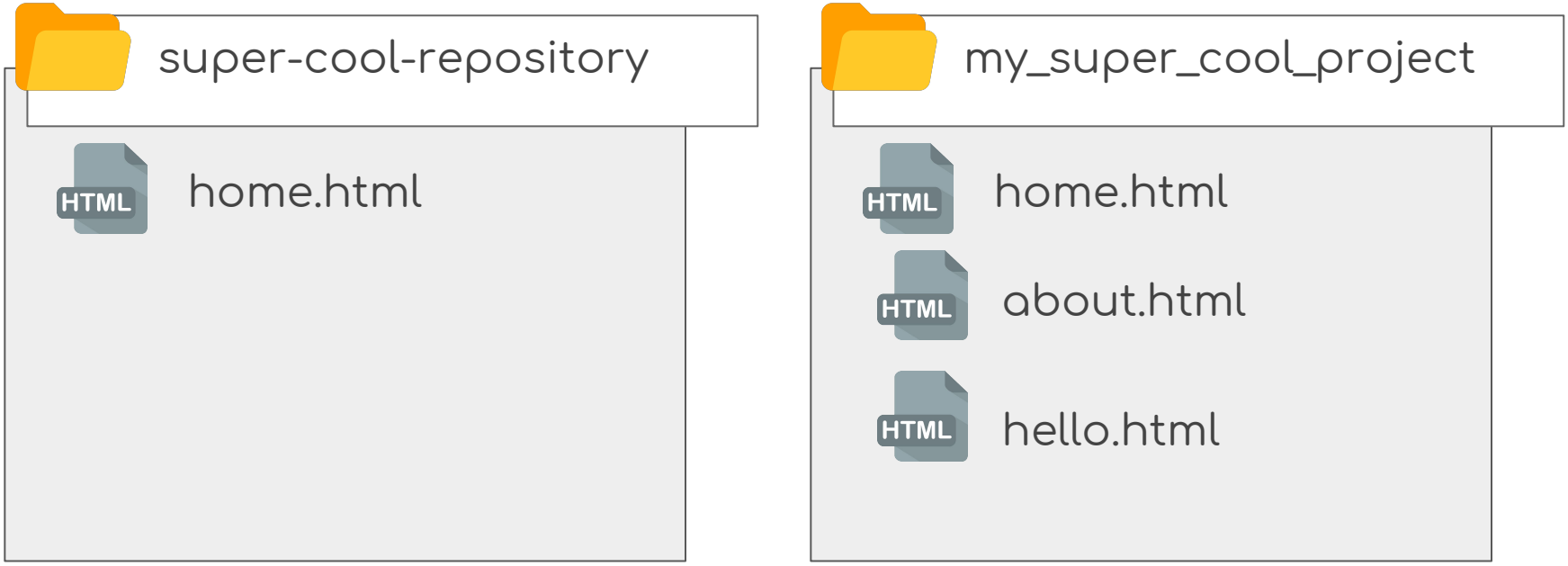


JOHN'S COMPUTER



The Flow of
Git

John add hello.html !



The Flow of
Git

John's Terminal

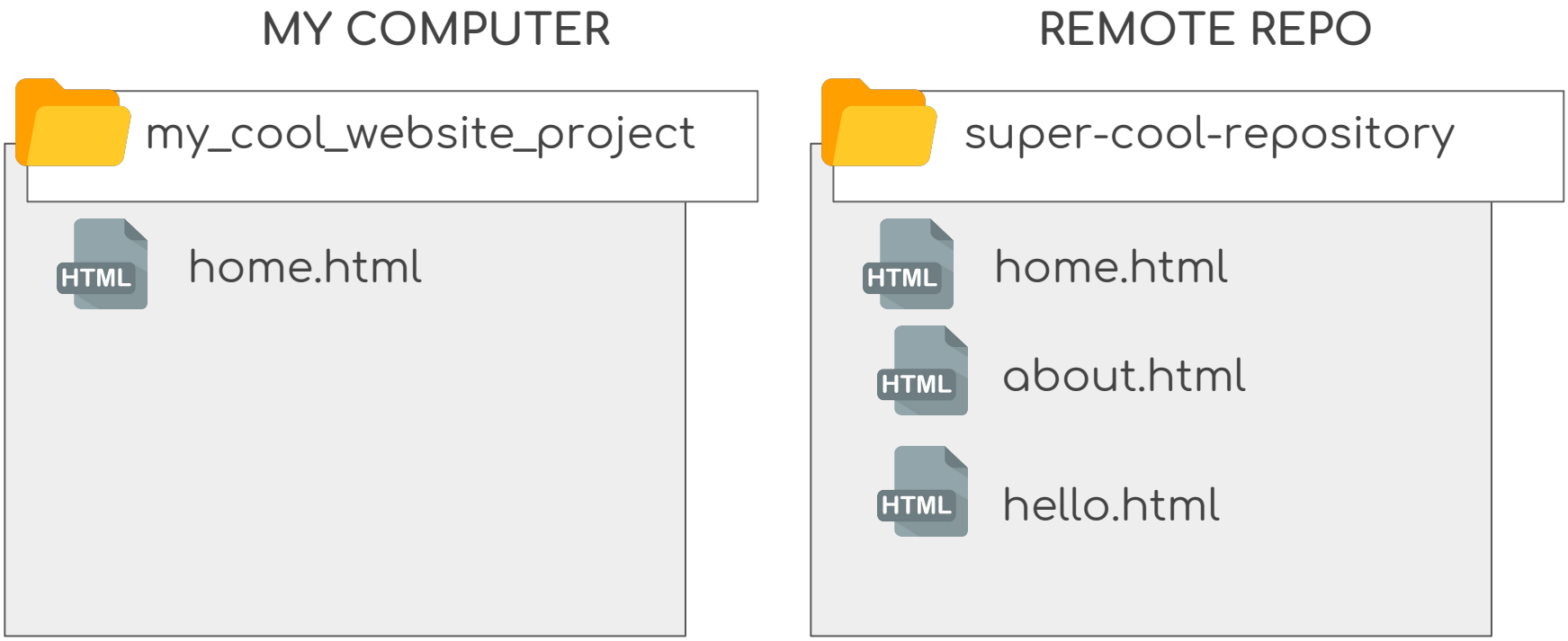
```
> git add about.html

> git add hello.html

> git commit -m "add hello and about page"

> git push origin master
```

What is
Git

John’s now in sync with remote repository

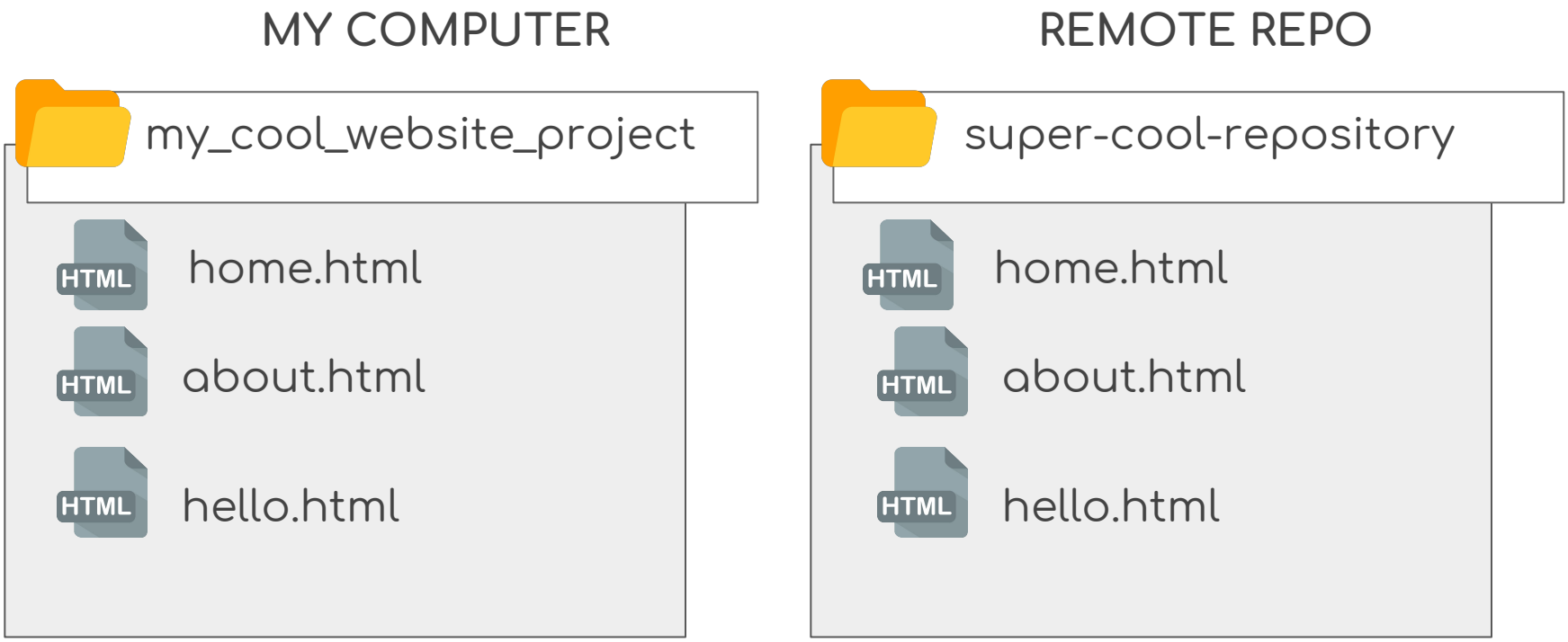
The Flow of
Git

Back to ME !



The Flow of
Git

TERMINAL: git pull origin master



What is Git

Yay! All sync!

The Flow of
Git

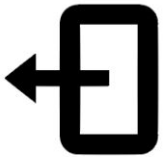
CONFLICTS

When changes on the same files
on the same line collides!

The Flow of
Git

1. `git commit`

2. `git push`



3. `leave building`

UNDERSTANDING
BRANCH

Understanding

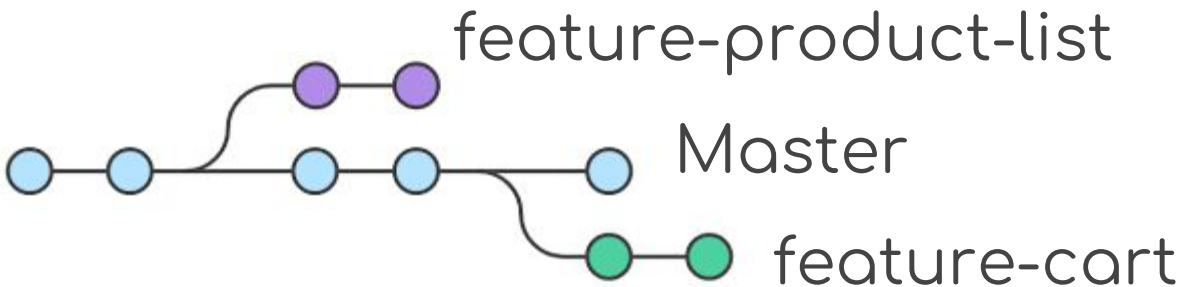
Branch

Let's build an e-commerce!

I need Product List Feature and
Add to Cart Feature

Understanding

Branch



Understanding

Branch

DO YOUR OWN EXPLORATION

> git branch
> git merge

Understanding

Branch

Demo Time!


GOT SOME
QUESTIONS?


THANK YOU AND
SEE YOU !