



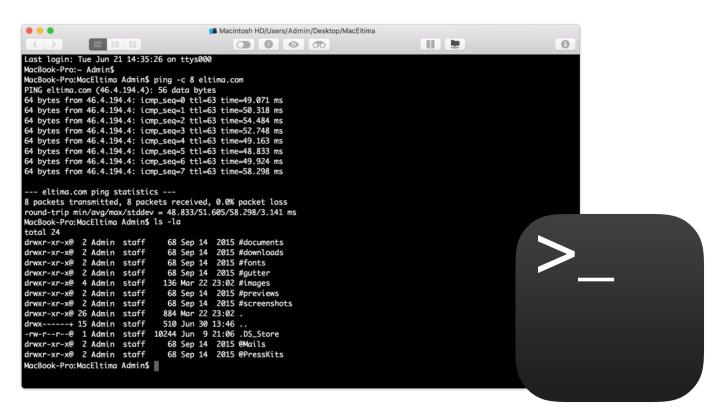
#### What's under the hood

## **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







#### **ISSUES**

- Conflicts and overwritten files
- Revision issues

#### **SOLUTION?**



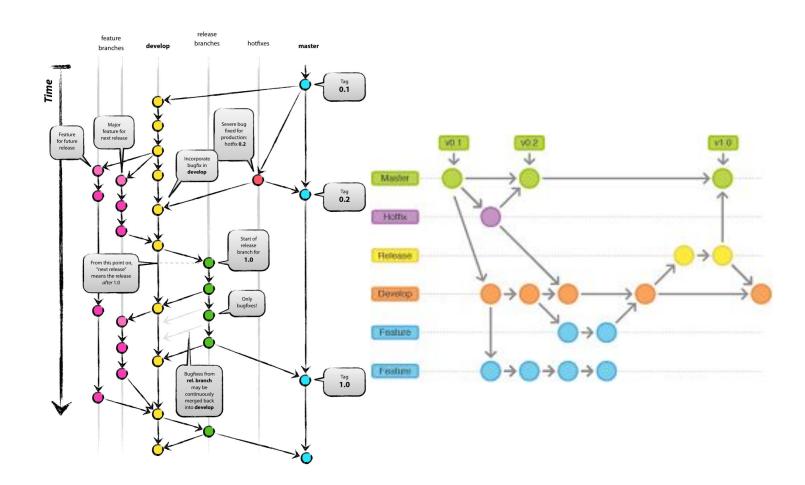




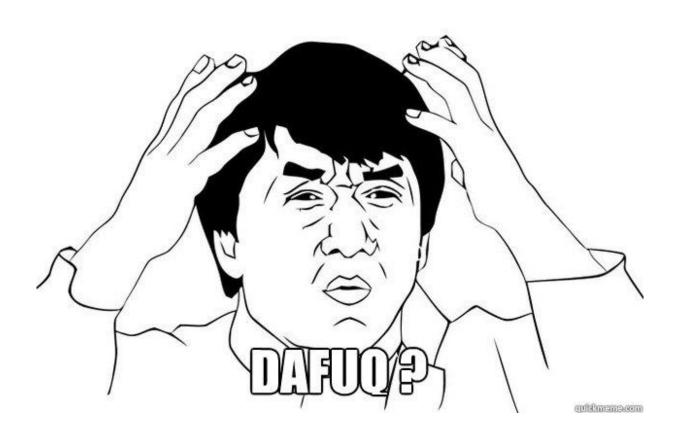
Git = version control system

#### What is

## Git



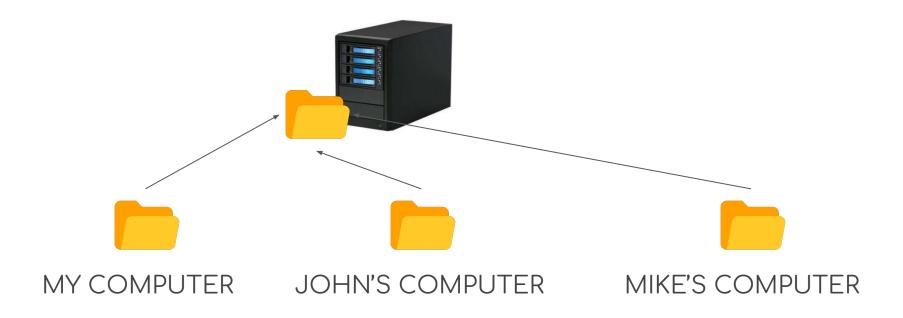
## What is





#### EVERYONE SHOULD SYNC TO THE REMOTE SERVER!

#### REMOTE COMPUTER / SERVER









Folders and Files



History





Git tracks every file changes.

Your changes, John's changes, everyone!





Git can undo to some "points"

We call it as Commit

Commit = the record of changes





#### IT IS QUITE COMPLICATED TO SETUP A GIT SERVER!



REMOTE COMPUTER / SERVER



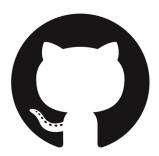
#### WE NEED A SERVICE TO BE THE SERVER!



REMOTE COMPUTER / SERVER

What is Git

#### **HELLO GITHUB!**



Github = git hosting service



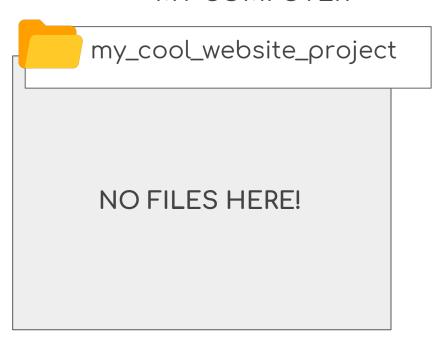
### Create empty repository in github



Go to
github.com
and create a new repository

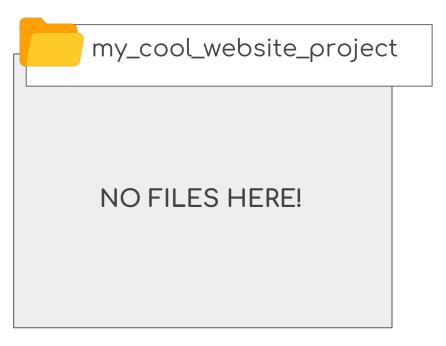


#### MY COMPUTER



### Prepare the Project Folder

#### MY COMPUTER



## Initialize the git using

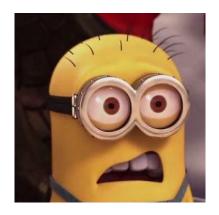
TERMINAL: \$ git init

## What is Git



Your Folder is now a Git Repository!





BUT still no connection to the remote repository!

#### MY COMPUTER



NO FILES HERE!

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

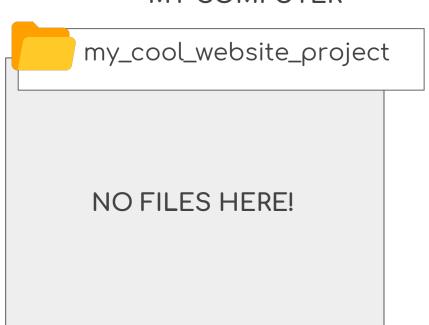


https://github.com/super -cool-repository What is **Git** 



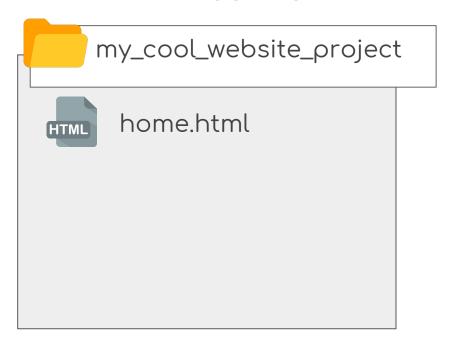
Yay... your local git is now connected to remote repository!

Now let's focus on your local computer..

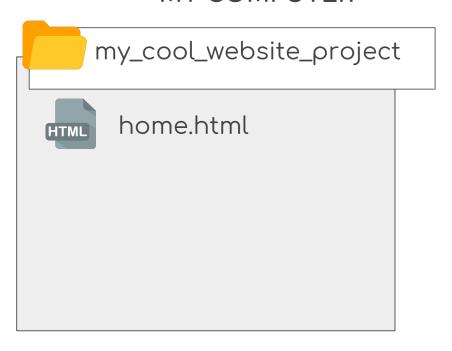


#### Add new file called home.html





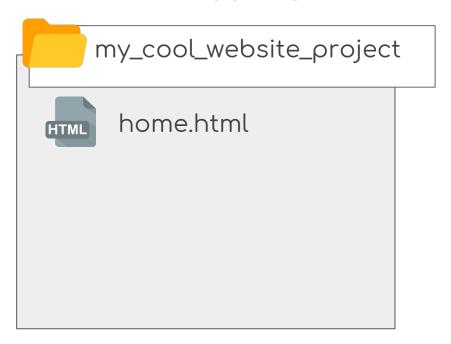
### At this point, home.html is UNSTAGED



What is **Git** 



All changes in MY COMPUTER
But not yet recorded in Git



## What is **Git**

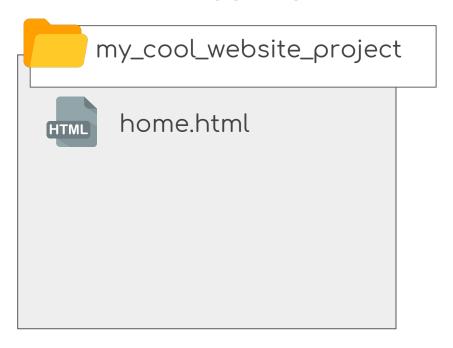


Home.html is now staged!

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

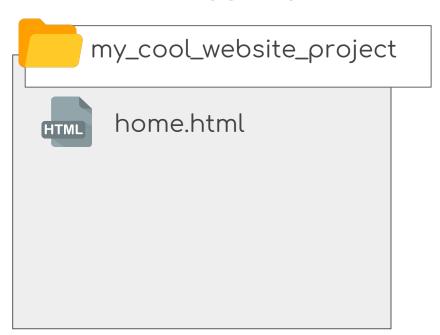
The Flow of

### MY COMPUTER



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

#### MY COMPUTER





Record some changes in MY COMPUTER to Git

### **COMMIT MESSAGE?**

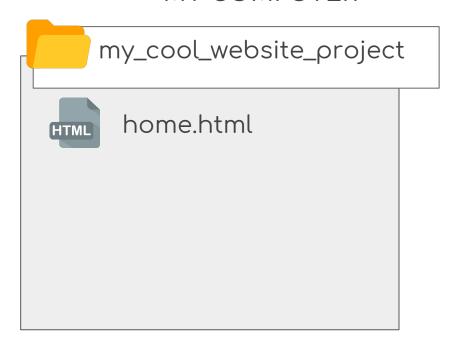
When I record something, I need to name / label them!

Example: "Add Home Page"

### The Flow of

### TERMINAL: \$ git commit -m "Add home page"

#### MY COMPUTER



## What is **Git**

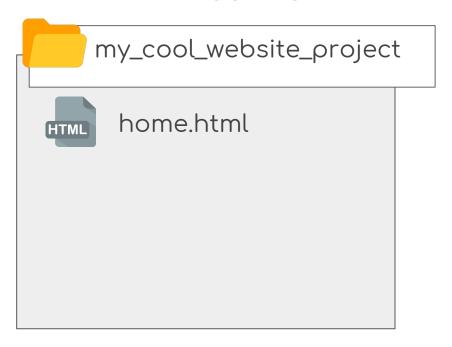


Home.html is now committed!

### home.html is now staged and committed



#### MY COMPUTER





## Recorded changes of files in MY COMPUTER'S GIT

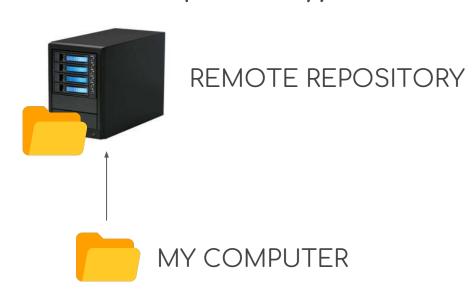


## Changes committed and I guess my work's done (for now).

So what now?



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

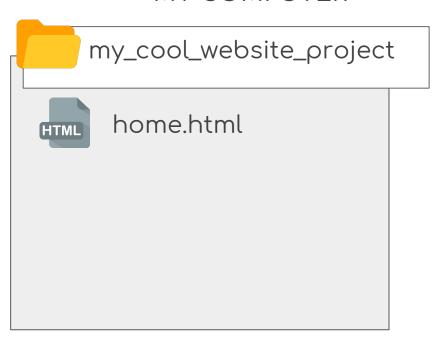


### **PUSH**

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



#### MY COMPUTER



### **REMOTE REPO**

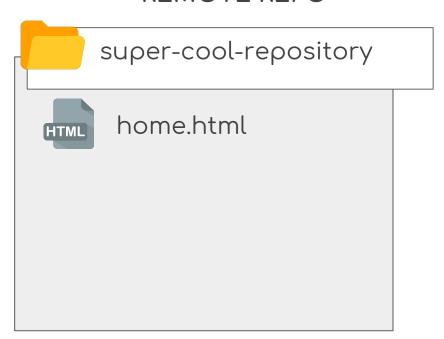




#### MY COMPUTER



### **REMOTE REPO**





Home.html is now in sync with remote repository!

Repo is up to date!

## Review!



**UNSTAGED** 

**STAGED** 

**UP-TO-DATE!** 

## Review!



Add files to staged them!

#### Let's

### Review!



Commit staged files Set the commit message



The Flow of

### First Time in John's Computer!

#### **REMOTE REPO**







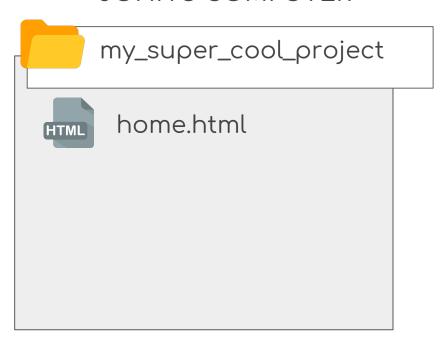
"Copy" the existing project from remote repository



### REMOTE REPO



### JOHN'S COMPUTER



The Flow of

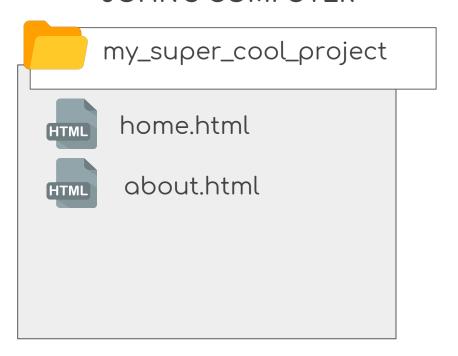
### John add about.html!

### Git

#### REMOTE REPO



### JOHN'S COMPUTER



### John add hello.html!



#### REMOTE REPO



#### JOHN'S COMPUTER



- > git add about.html
- > git add hello.html
- > git commit -m "add hello and about page"
- > git push origin master

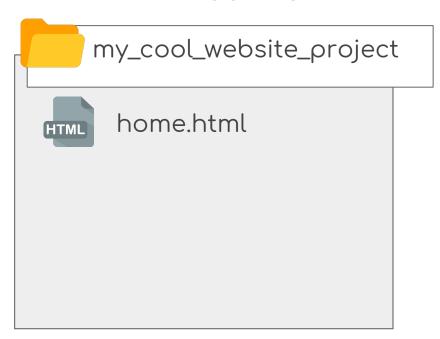
## What is Git



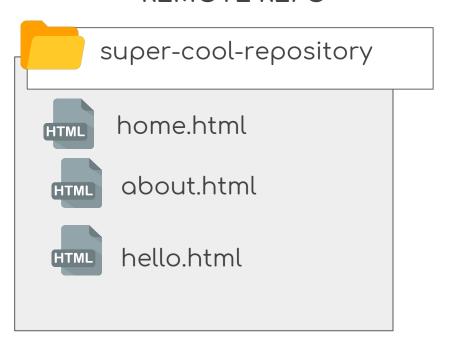
John's now in sync with remote repository



#### MY COMPUTER



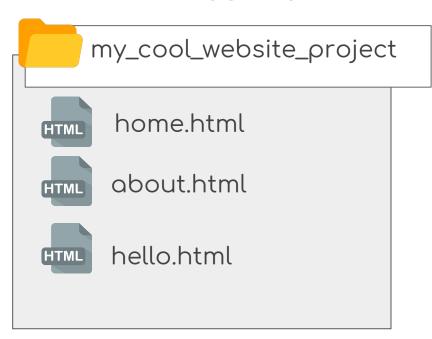
### **REMOTE REPO**



### TERMINAL: git pull origin master



#### MY COMPUTER



#### REMOTE REPO



## What is **Git**



Yay! All sync!



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

# In case of fire





1. git commit



2. git push



→ 3. leave building



### Understanding

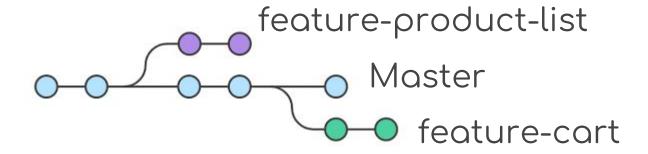
### Branch



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!



