



git

Introduction to Version control

Git gud.
PSA: It's not that hard



GitHub

Who is this guy?

Hi, I'm Chuan Hao



- Year 1 DIT student (DSDA)
- I cannot do the hacking
- I can barely do the AI
- I can maybe do the coding
- I almost didn't do the talk
- Why is the school not teaching this
- I am running out of things to write
- This is a placeholder text

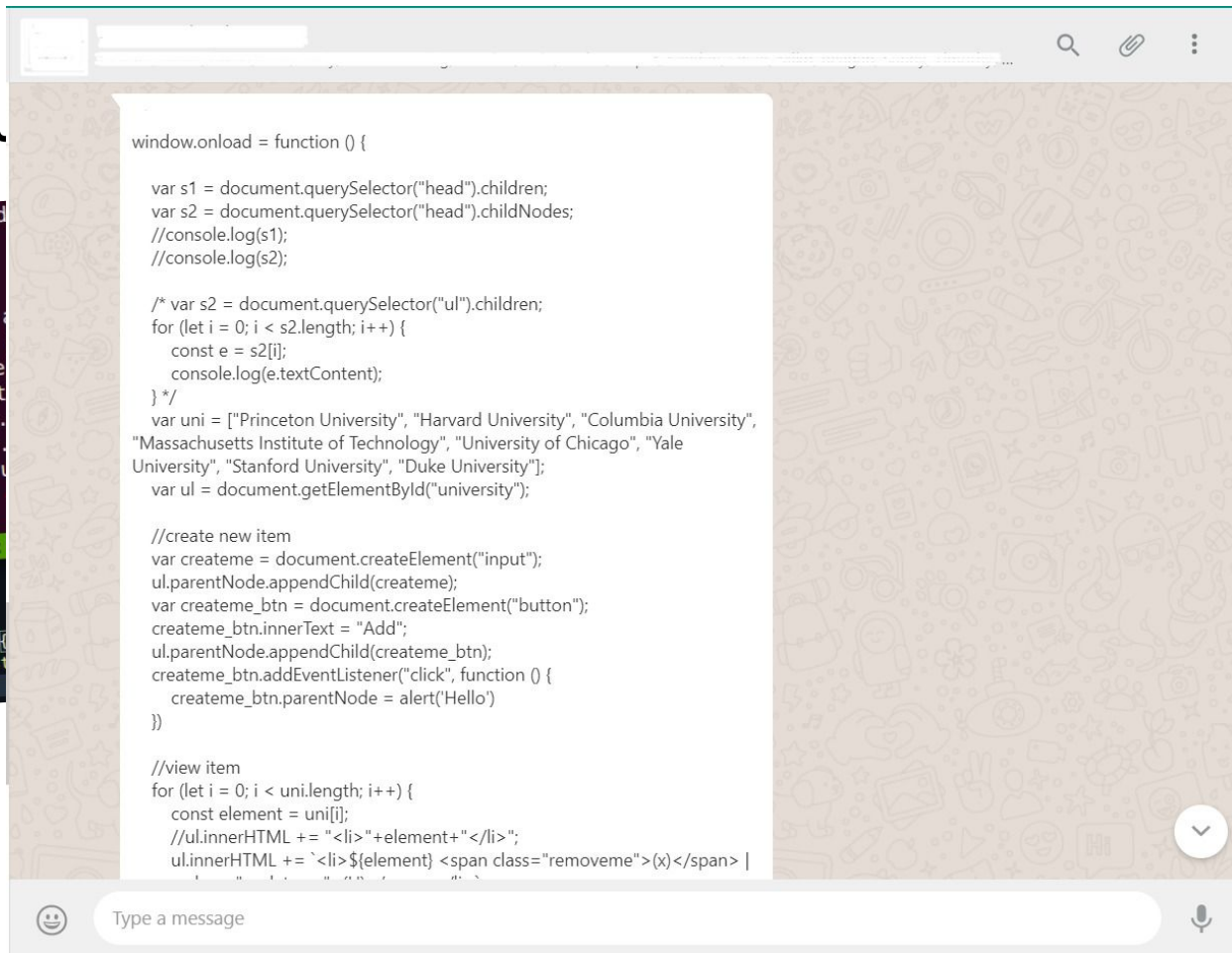
Also I am part of SEED



@chuanhao01

Have you

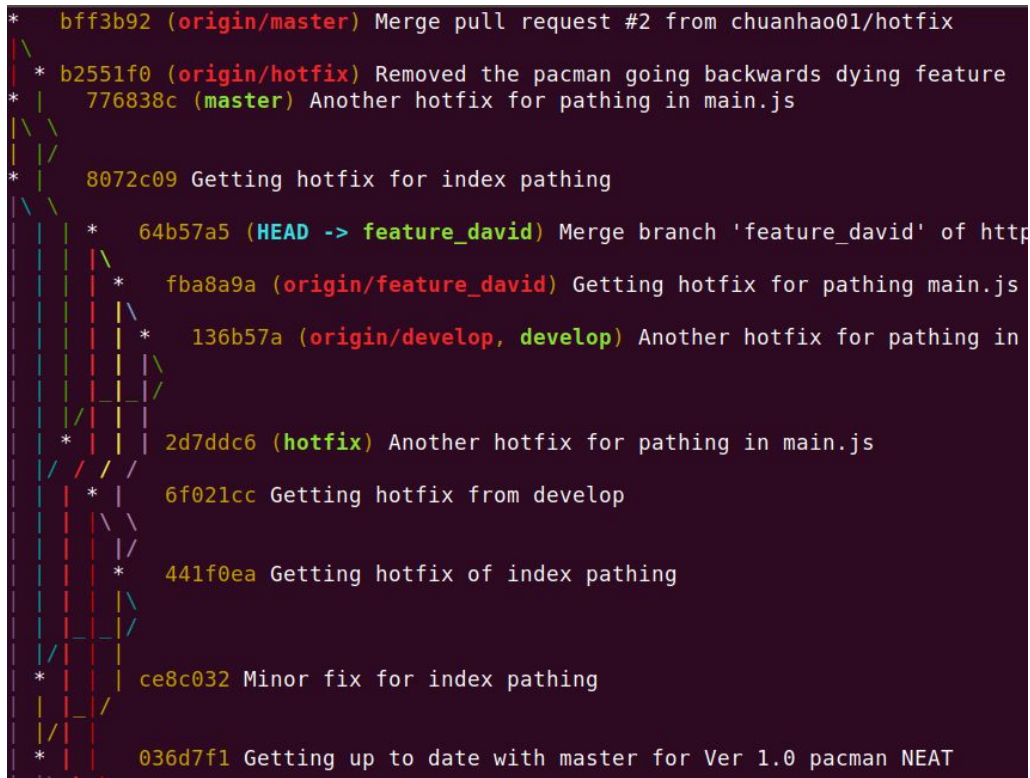
```
/home/chuanhao01/My_Fold
}
^
SyntaxError: missing )
    at Module._compile
    at Object.Module._e
    at Module.load (int
    at Function.Module.
    at Function.Module.
    at internal/main/r
[nodemon] app crashed
[0] 0: bash 1: bash 2:
// Setting up ports
const PORT = 3000;
app.listen(PORT, function() {
  console.log('Server list
```



the same

What if there was a way to solve this...

- What if you could track all changes on the project?
- What if you could properly collaborate on projects, without having to code on the same laptop?



```
* bff3b92 (origin/master) Merge pull request #2 from chuanhao01/hotfix
* b2551f0 (origin/hotfix) Removed the pacman going backwards dying feature
* 776838c (master) Another hotfix for pathing in main.js
* 8072c09 Getting hotfix for index pathing
* 64b57a5 (HEAD -> feature_david) Merge branch 'feature_david' of http
* fba8a9a (origin/feature_david) Getting hotfix for pathing main.js
* 136b57a (origin/develop, develop) Another hotfix for pathing in
* 2d7ddc6 (hotfix) Another hotfix for pathing in main.js
* 6f021cc Getting hotfix from develop
* 441f0ea Getting hotfix of index pathing
* ce8c032 Minor fix for index pathing
* 036d7f1 Getting up to date with master for Ver 1.0 pacman NEAT
```

But this looks complicated.....

What is this??

```
* bff3b92 (origin/master) Merge pull request #2 from chuanhao01/hotfix
* b2551f0 (origin/hotfix) Removed the pacman going backwards dying feature
* 776838c (master) Another hotfix for pathing in main.js

8072c09 Getting hotfix for index pathing

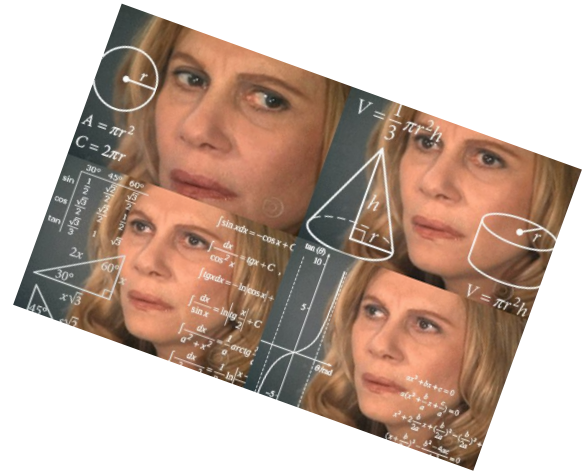
* 64b57a5 (HEAD -> feature_david) Merge branch 'feature_david' of http
* fba8a9a (origin/feature_david) Getting hotfix for pathing main.js
* 136b57a (origin/develop, develop) Another hotfix for pathing in

2d7ddc6 (hotfix) Another hotfix for pathing in main.js
6f021cc Getting hotfix from develop

441f0ea Getting hotfix of index pathing

ce8c032 Minor fix for index pathing

036d7f1 Getting up to date with master for Ver 1.0 pacman NEAT
```



PSA: It's rarely like that

Most of the time your version control would look something like this



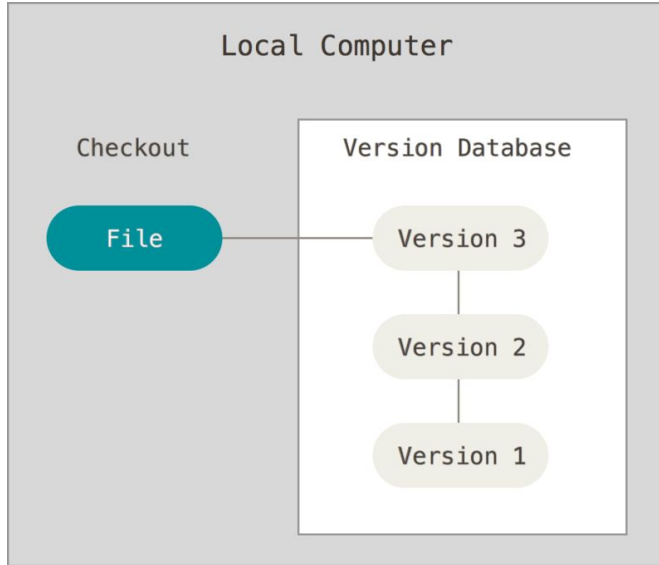
```
* 318576c Removing old code from previous
* 5911490 Fixed account creation and login so no duplicate username
* 7dabe43 Changed get queries to search only for non-deleted
* 623f204 Added view product button
* 413c7ef Deleting listing works
* f3cf0c4 Added semi-working view all listings
* 63fe81b BED Finished Q10
* 4cc5820 BED Finished Q9
* ade7396 BED Finished Q8
* 2f68340 BED Finished Q7
* e114451 BED finished Q5 and Q6
* f6a6358 Able to add a listing by a user
* 3989d36 Adding in page and request for adding a listing
* 24d58c1 Chaning parentDir format
* 709903e BED PUT /users/:id
* 4a02b97 BED changed files to new promise format
* ea9e86f BED Fixed POST /user db to create a user
* 38807f8 Final promise format
* 53d7b31 Added jwt expire, fixed format of usersDB
* 042e6b2 BED added GET /users/:id
* e1dcde3 Successfully added /post USERS
* 774fbf0 Fixed creation status error code from 200 to 201
```

So, what is version control?

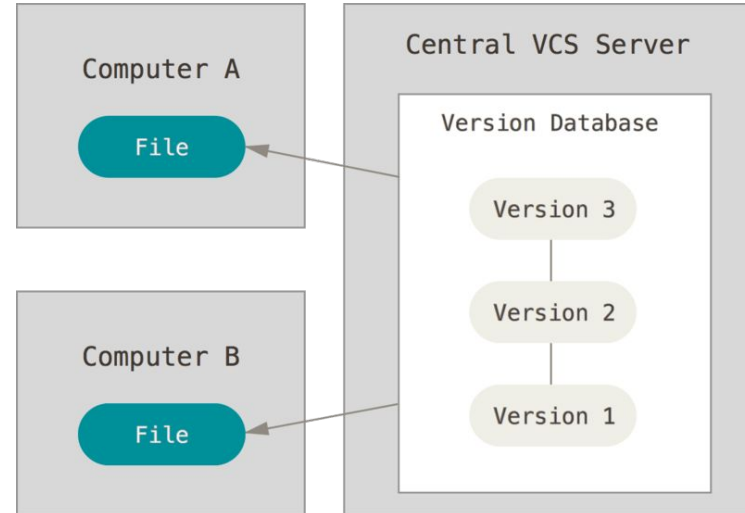
“Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.”

— *Git*

For single users/personal projects



For multiple people. (Not covered today)



So it's time to **Git gud**

Me: Oh, no, I accidentally `git commit --amend` over a big commit!

My Source Control:



Now how do I `--amend` this?!

Git

- This is the version control software that we are going to use today
- Benefits
 - Free
 - Works on most Operating Systems(OS)
 - Integration with other cloud based git services (Github, Gitlab, Bitbucket, etc.)
 - A lot of people use it

PS: We will get to github later on



git

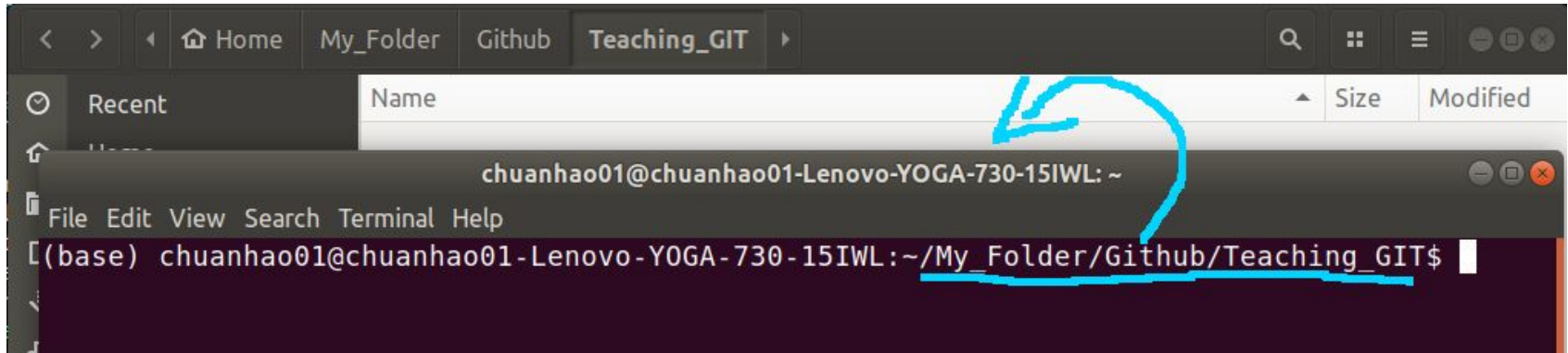
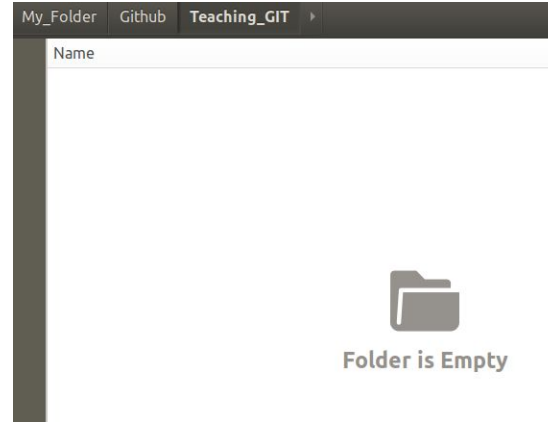
Installation process

1. Go to
<https://git-scm.com/downloads>
2. Download the corresponding git version for your OS
3. Install Git.



Setup

1. Now make a new folder where you will be playing with git.
2. Make sure you open a git bash terminal in the folder.

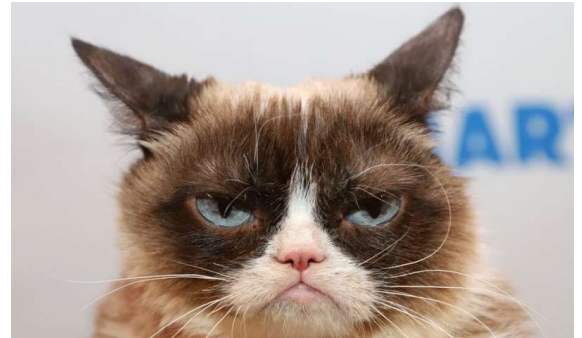


The command to watch them all

git status

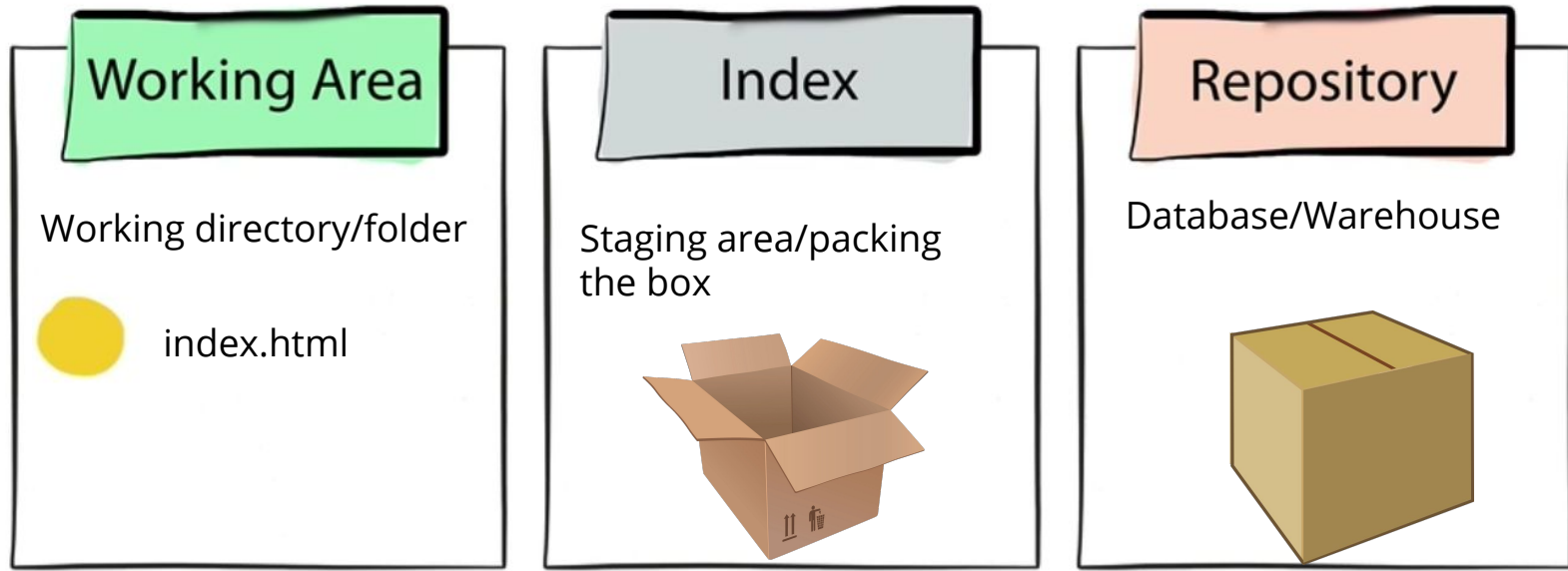
Like getting the status on anything, it tells you what is currently going on in git.

I don't know how to make this funny or really more entertaining so here are some cats

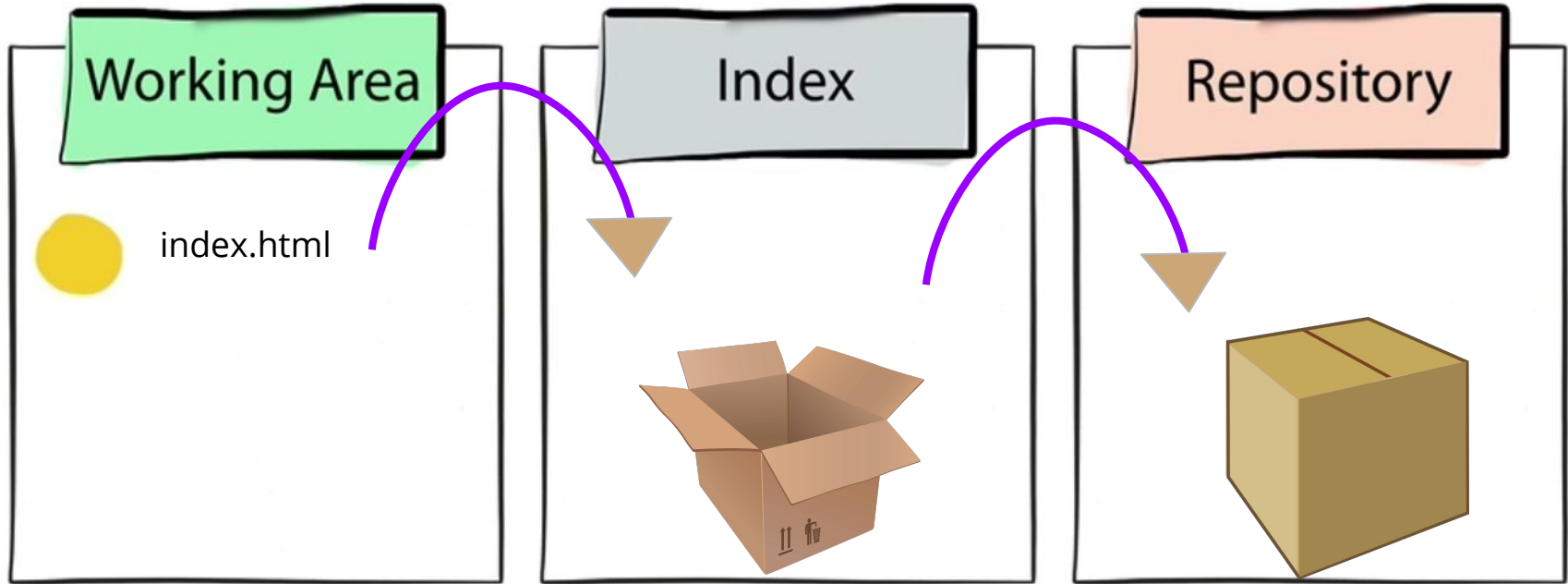


But wait, before we go

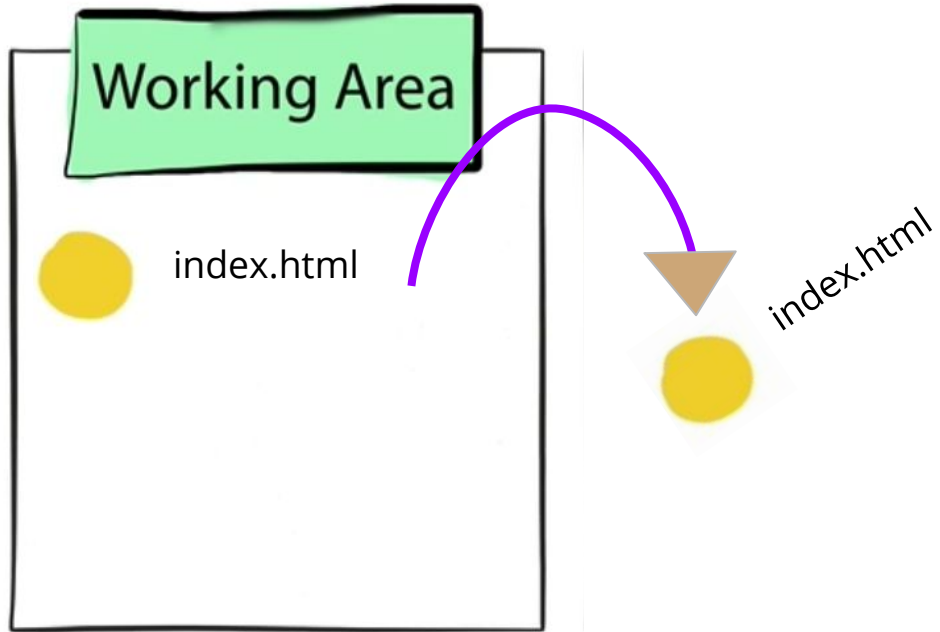
Some fundamental theory is needed:



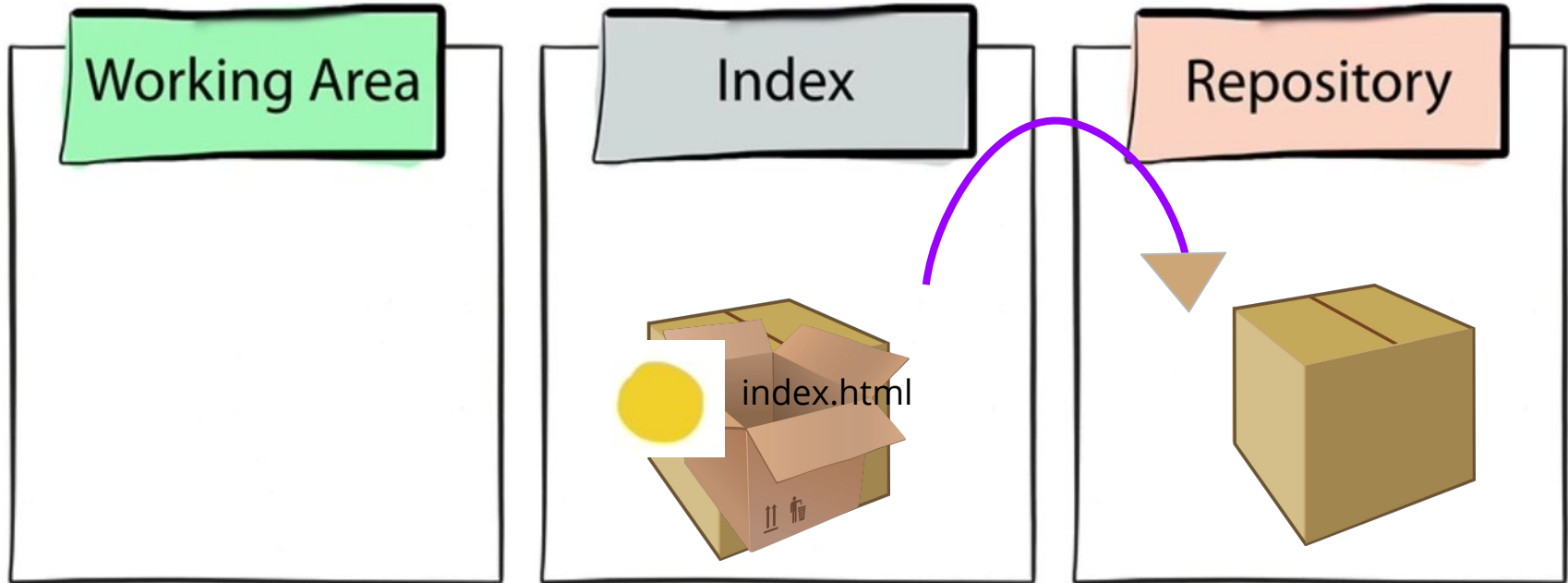
Fundamental theory (adding code)



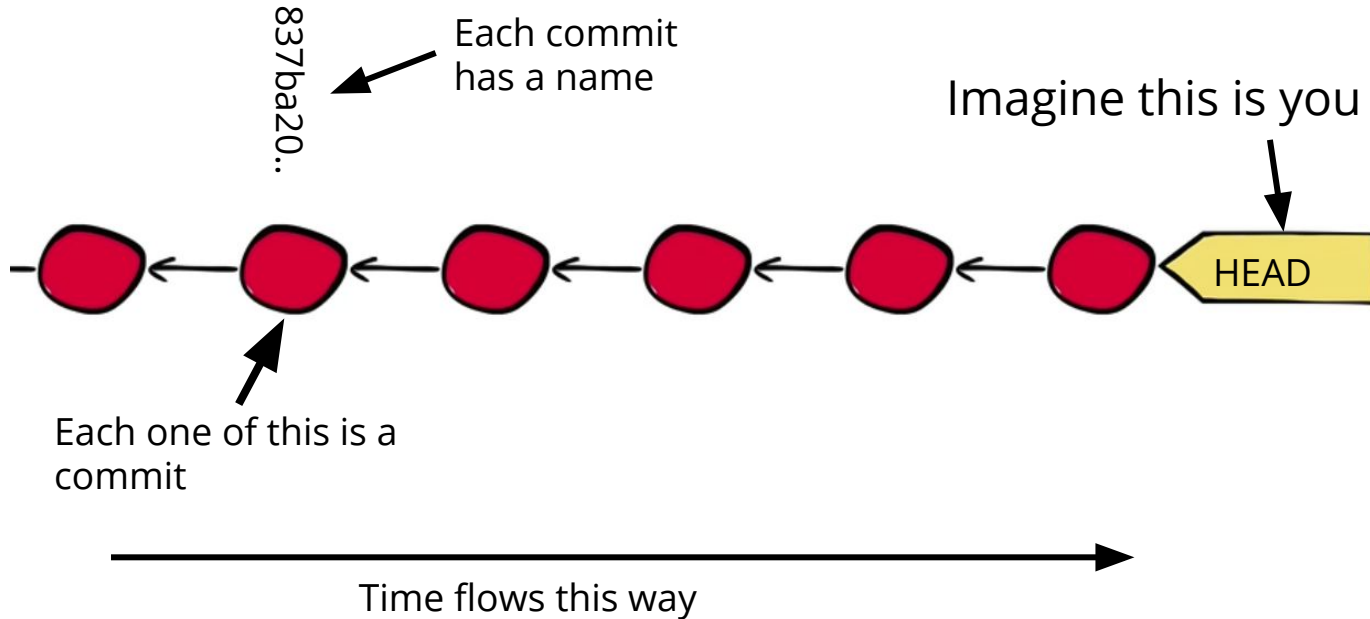
Git Add



Git commit

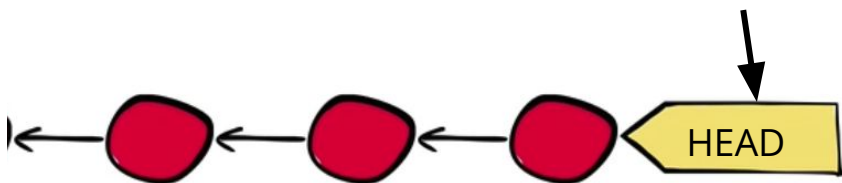


Fundamental theory (Looking at commits)



Fundamental theory (Looking at commits)

Imagine this is you



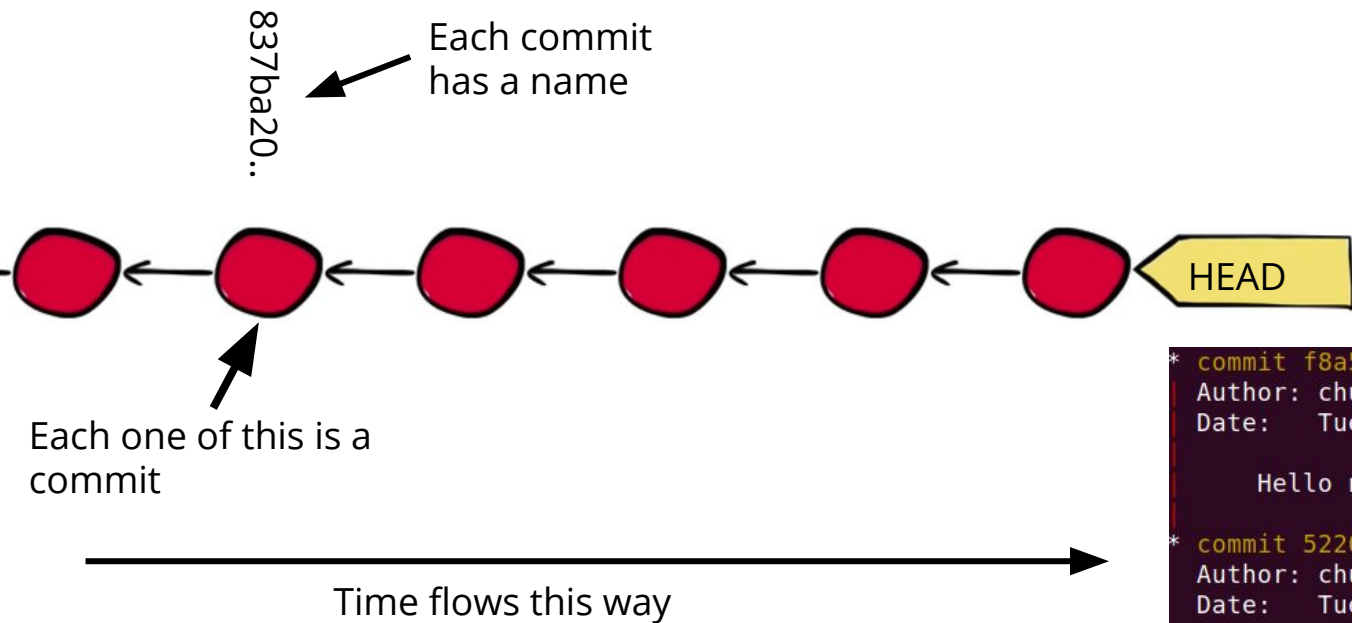
```
* commit f8a5ef7d23c5c6ab28353a82c80b2b4cdaefab64 (HEAD -> master)
Author: chuanhao01 <chuanhaolim@gmail.com>
Date:   Tue Jan 14 14:49:22 2020 +0800

    Hello number 2

* commit 522074ab52d1de4f4733fb99dccaafd8f6e0b331c
Author: chuanhao01 <chuanhaolim@gmail.com>
Date:   Tue Jan 14 14:49:05 2020 +0800

    Adding first file
```

Fundamental theory (Looking at commits)



```
* commit f8a5ef7d23c5c6ab28353a82c80b2b4cdaefab6
Author: chuanhao01 <chuanhaolim@gmail.com>
Date: Tue Jan 14 14:49:22 2020 +0800

    Hello number 2

* commit 522074ab52d1de4f4733fb99dccaafd8f6e0b331
Author: chuanhao01 <chuanhaolim@gmail.com>
Date: Tue Jan 14 14:49:05 2020 +0800

    Adding first file
```

The magical command `git log --graph --oneline`

```
* f8a5ef7 (HEAD -> master) Hello number 2
* 522074a Adding first file
```



Commit's
name

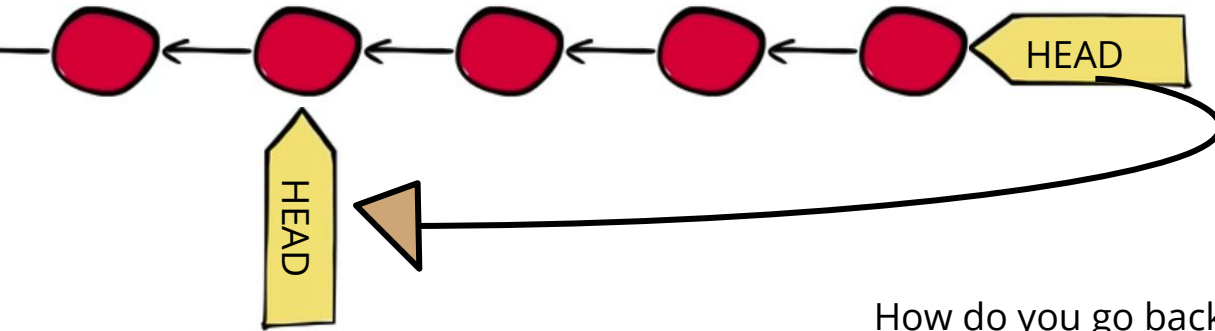


HEAD (You)



Your commit's
message

Now how do we go back in time?



How do you go back and see how the code was back then?

Checking out those commits `git checkout <commit name>`

Now you can go back in time to look at those commits.

You can check them out!

But don't forget to go back to your master

```
* f8a5ef7 (HEAD -> master) Hello number 2
* 522074a Adding first file
(base) chuanhao01@chuanhao01-Lenovo-YOGA-730-15IWL:~/My_Folder/Github/Teaching_GIT$ git checkout 522074a
Note: checking out '522074a'.
```

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using `-b` with the checkout command again. Example:

```
git checkout -b <new-branch-name>
```

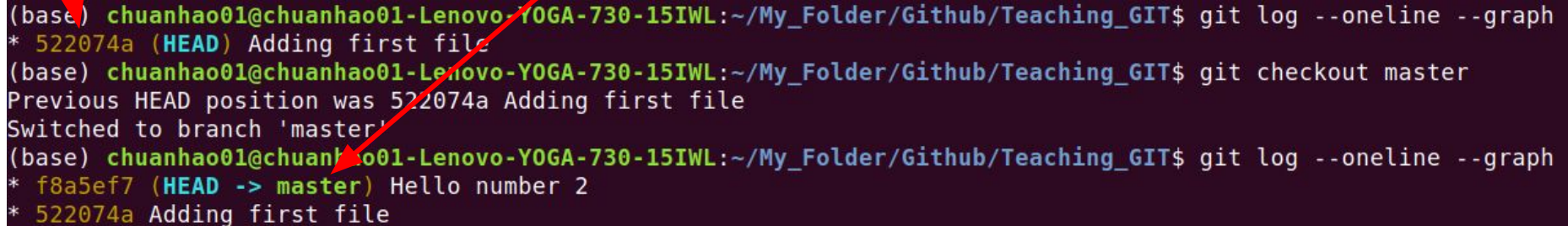
```
HEAD is now at 522074a Adding first file
```


Going back to the present `git checkout master`

Just like how you wanted a blast from the past, you still have to go back to your master.

Was checking out another commit

Back to master

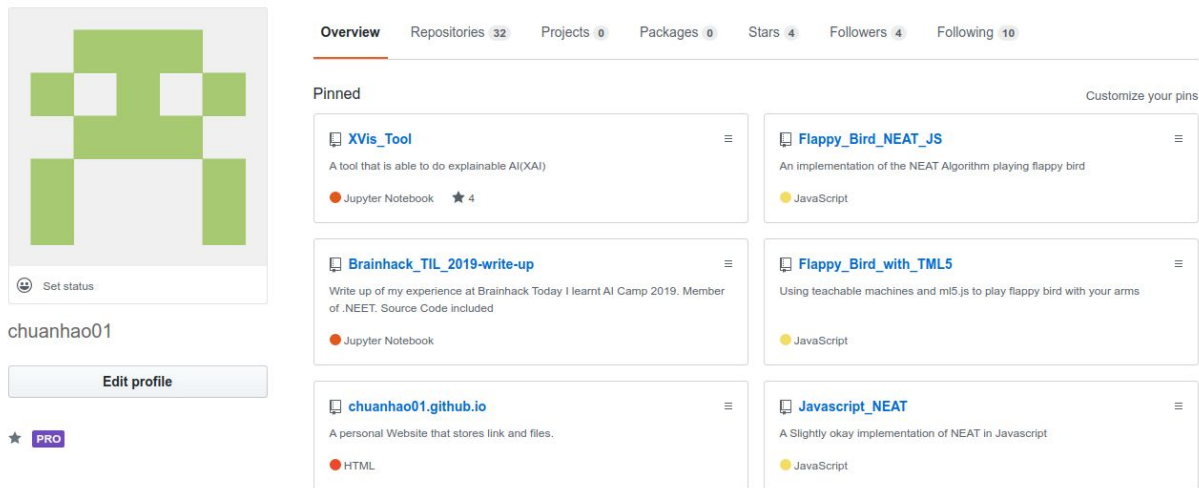


```
(base) chuanhao01@chuanhao01-Lenovo-YOGA-730-15IWL:~/My_Folder/Github/Teaching_GIT$ git log --oneline --graph
* 522074a (HEAD) Adding first file
(base) chuanhao01@chuanhao01-Lenovo-YOGA-730-15IWL:~/My_Folder/Github/Teaching_GIT$ git checkout master
Previous HEAD position was 522074a Adding first file
Switched to branch 'master'
(base) chuanhao01@chuanhao01-Lenovo-YOGA-730-15IWL:~/My_Folder/Github/Teaching_GIT$ git log --oneline --graph
* f8a5ef7 (HEAD -> master) Hello number 2
* 522074a Adding first file
```

The cloud, git cloud, wait I mean github

Now that we can add, commit and checkout stuff in git, how do we share and save code on the cloud?

The answer: **Github** (Told you we would get here)



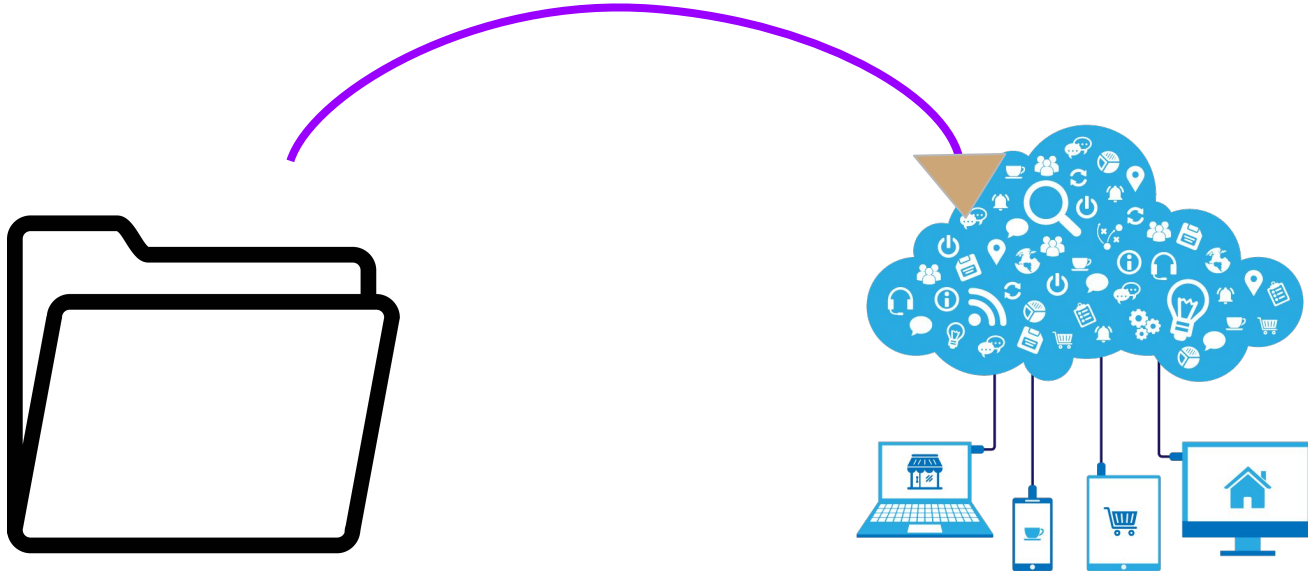
The screenshot shows a GitHub profile for the user 'chuanhao01'. On the left is a green pixelated avatar. Below it is a 'Set status' button and the username 'chuanhao01'. Further down is an 'Edit profile' button and a 'PRO' badge. The main content area has a navigation bar with 'Overview' selected, and links for 'Repositories 32', 'Projects 0', 'Packages 0', 'Stars 4', 'Followers 4', and 'Following 10'. Below the navigation bar is a 'Pinned' section with a 'Customize your pins' link. It contains six pinned items arranged in two columns:

- XVis_Tool**: A tool that is able to do explainable AI(XAI). Jupyter Notebook. 4 stars.
- Brainhack_TIL_2019-write-up**: Write up of my experience at Brainhack Today I learnt AI Camp 2019. Member of .NEET. Source Code included. Jupyter Notebook.
- chuanhao01.github.io**: A personal Website that stores link and files. HTML.
- Flappy_Bird_NEAT_JS**: An implementation of the NEAT Algorithm playing flappy bird. JavaScript.
- Flappy_Bird_with_TML5**: Using teachable machines and ml5.js to play flappy bird with your arms. JavaScript.
- Javascript_NEAT**: A Slightly okay implementation of NEAT in Javascript. JavaScript.

Github and its magical cloud storage

You can think of github as a cloud version of your projects.

Like google drive, but cooler



The github repository

Making your first repo



Overview **Repositories 32** Projects 0 Packages 0 Stars 4 Followers 4 Following 10

Find a repository... Type: All Language: All **New**

Flappy_Bird_with_TML5 ★ Star

Using teachable machines and ml5.js to play flappy bird with your arms


JavaScript Updated 7 days ago

The github 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:  chuanhao01 ▾ /

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

Description (optional)

☒  **Public**
Anyone can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

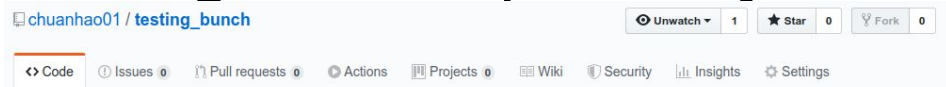
☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer.

Add .gitignore: **None** ▾ | Add a license: **None** ▾ ⓘ

Pick a name for your first repository.
This can be different from your file name.

Then create the repository

The github repository



Quick setup — if you've done this kind of thing before

or **HTTPS** **SSH** `https://github.com/chuanhao01/testing_bunch.git`

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# testing_bunch" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/chuanhao01/testing_bunch.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/chuanhao01/testing_bunch.git
git push -u origin master
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

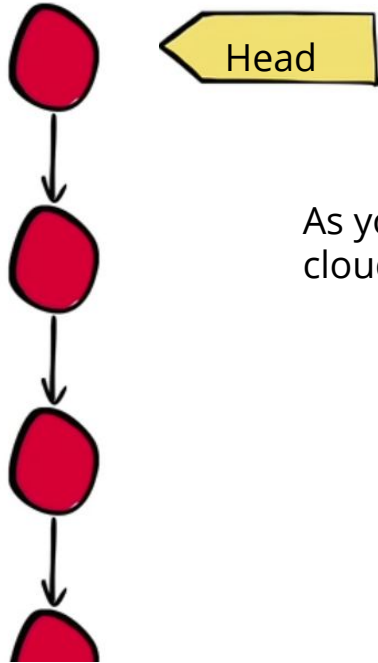
[Import code](#)

ProTip! Use the URL for this page when adding GitHub as a remote.

Now this is all you have to follow to connect the repositories

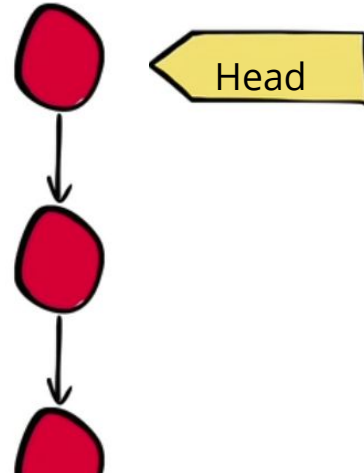
How the cloud works

On your computer



As you make more commits, the cloud may not be updated.

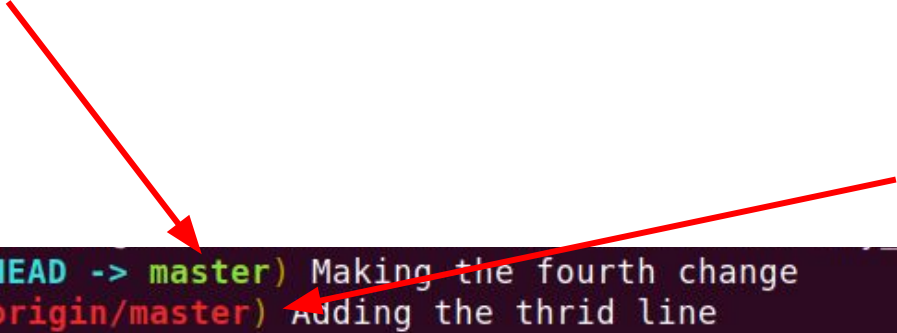
In the cloud



How it looks like in the code

Your local repo is ahead

You can see here that the github repo is behind



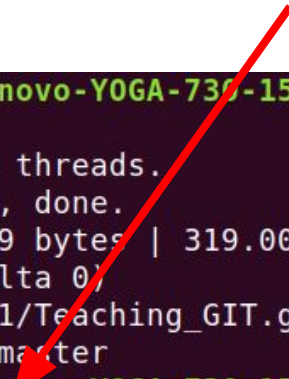
```
* 868170e (HEAD -> master) Making the fourth change
* b6f900a (origin/master) Adding the thrid line
* f8a5ef7 Hello number 2
* 522074a Adding first file
```

Pushing to the cloud **git push**

You have to push the code up to the cloud to update.

Keep on pushing

Now its updated



```
(base) chuanhao01@chuanhao01-Lenovo-YOGA-730-15IWL:~/My_Folder/Github/Teaching_GIT$ git push
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 319 bytes | 319.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/chuanhao01/Teaching_GIT.git
   b6f900a..868170e  master -> master
(base) chuanhao01@chuanhao01-Lenovo-YOGA-730-15IWL:~/My_Folder/Github/Teaching_GIT$ gl
* 868170e (HEAD -> master, origin/master) Making the fourth change
* b6f900a Adding the thrid line
* f8a5ef7 Hello number 2
* 522074a Adding first file
```

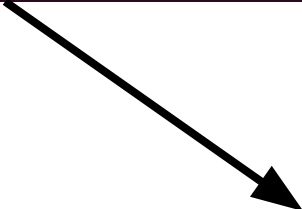
When you push, you can also pull **git pull**

If there are changes on github and you want to **pull** them down.

```
* 7a6b28d (origin/master) Adding changes from github
* 868170e (HEAD -> master) Making the fourth change
* b6f900a Adding the thrid line
* f8a5ef7 Hello number 2
* 522074a Adding first file
```

Github

You



```
(base) chuanhao01@chuanhao01-Lenovo-YOGA-730-15IWL:~/My_Folder/Github/Teaching_GIT$ git pull
Updating 868170e..7a6b28d
Fast-forward
 hi | 2 ++
 1 file changed, 2 insertions(+)
```

Kage Bunshin no Jutsu (Cloning technique)

Now say your friend wants to **clone** the repo on github to play around with the code. You can clone it!

git clone <link>



```
(base) chuanhao01@chuanhao01-Lenovo-YOGA-730-15IWL:~/My_Folder/Github$ git clone https://github.com/chuanhao01/Teaching_GIT.git
Cloning into 'Teaching_GIT'...
remote: Enumerating objects: 15, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 15 (delta 0), reused 12 (delta 0), pack-reused 0
Unpacking objects: 100% (15/15), done.
```

Some extra cool stuff

- Creating from git clone
- Basic branching
- Basic collaboration
- Going back in time

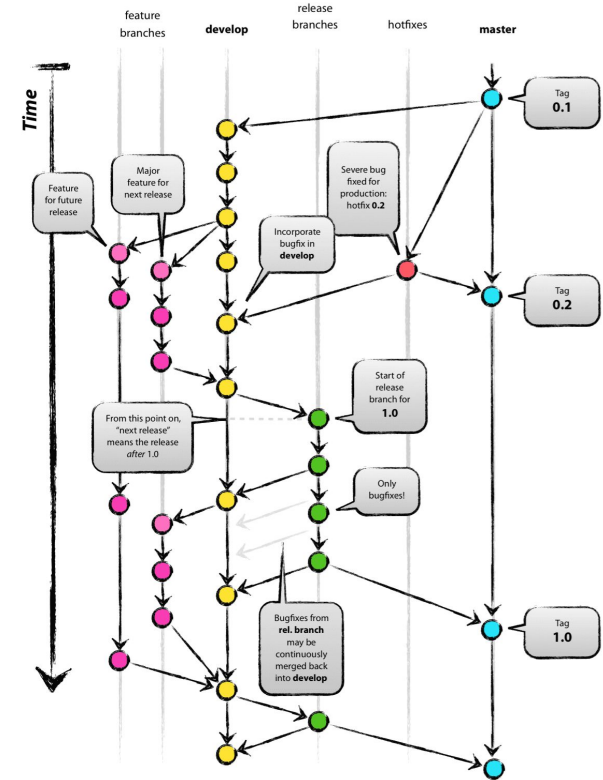
da future

- Git a friend (Advanced git and github workshop)
 - Git branches
 - Collaboration
 - Git flow (How to set up workflows and some best practices)
 - Github forks and remotes

Watch 35

Star 45

Fork 37



How do you end a presentation again? Do I click the button over here?