Git!

Let’s Git Started!

ILOs

- Version Control Systems
- Git
- GitHub
- Use command-line tool
- Useful GitHub - VS Code integrations

Version control systems

Version control (or revision control, or source control) is all about managing multiple versions of documents, programs, web sites, etc.

Some well-known version control systems are CVS, Subversion, Mercurial, and **Git**

Why version control?

Download and install Git

http://git-scm.com/downloads

(Debian / Ubuntu):
$ sudo apt-get install git

Don’t memorize commands

Verify installation

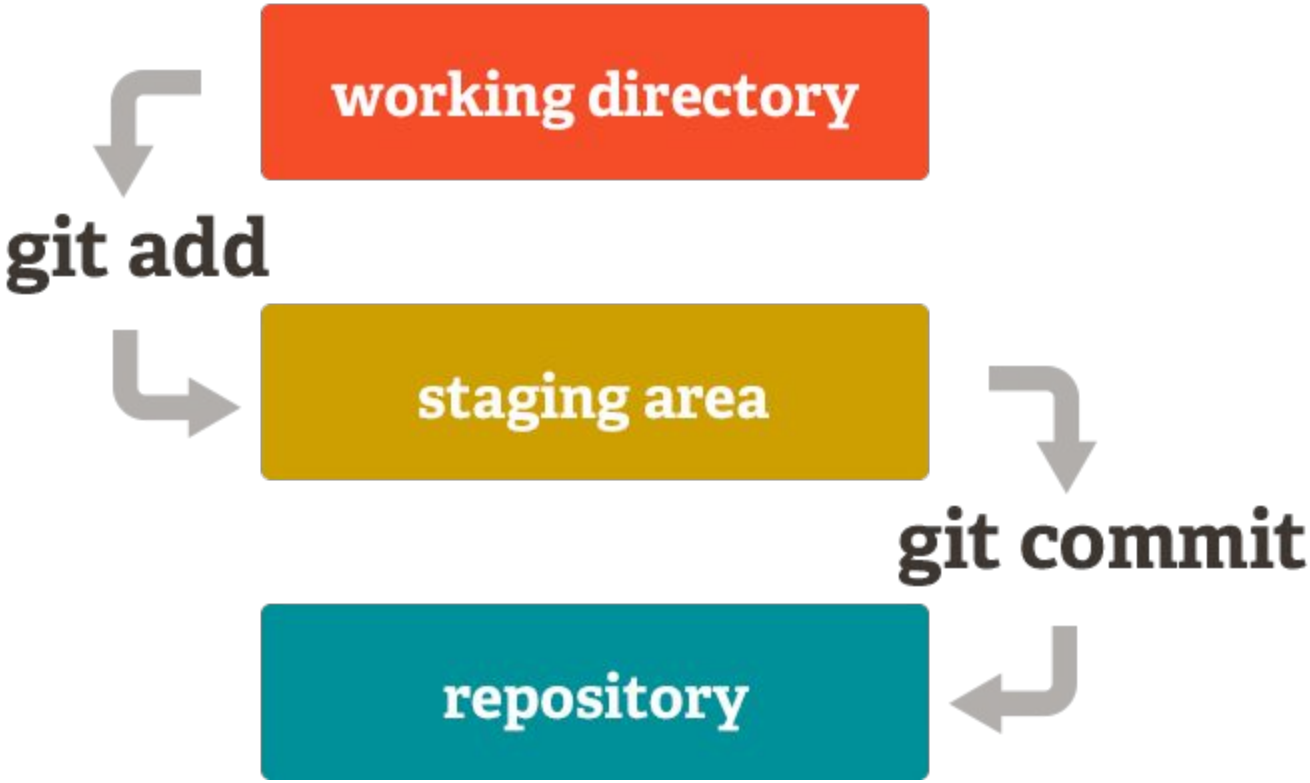
cmd:
git version

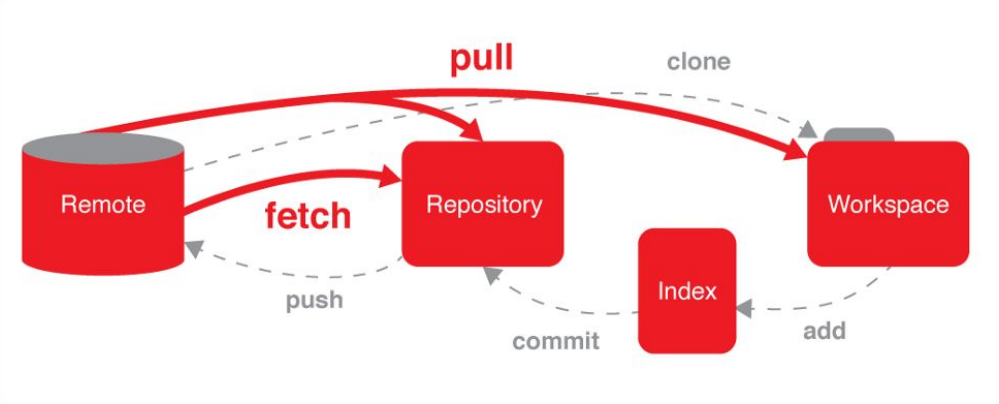
(Debian / Ubuntu):
$ git --version

Introduce yourself to Git

git config --global user.name "Your Name Here"
git config --global user.email youremail@email.com

You only need to do this once
Then use: “git init” to initialize the repository in your working directory

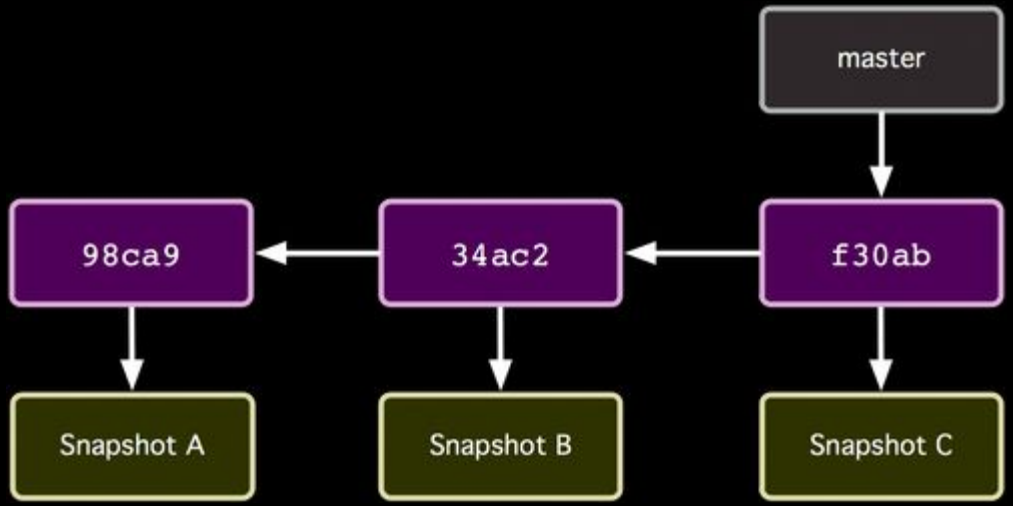

working directory
git add
staging area
git commit
repository



GitHub

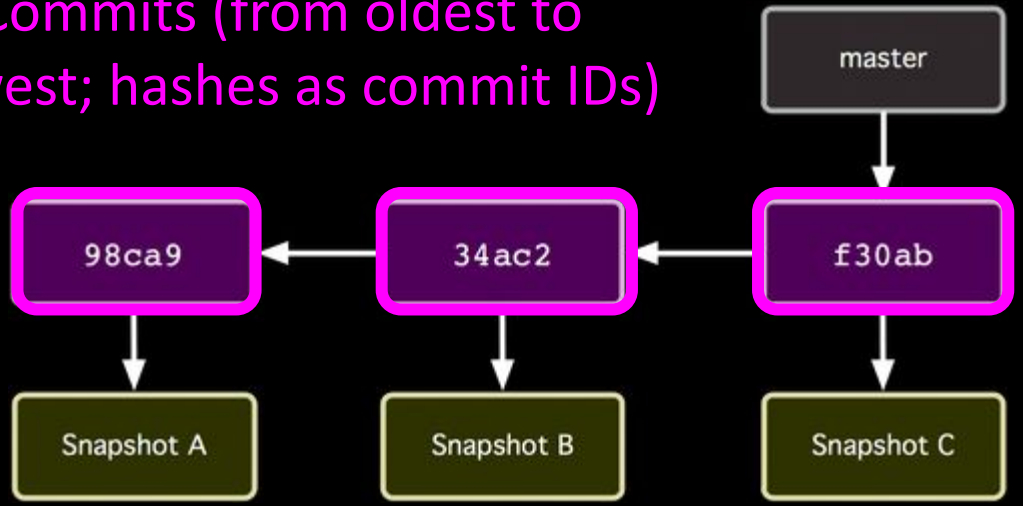
Remote repositories are versions of your project hosted on the internet or network.
Used for collaboration.

How the repos is organized



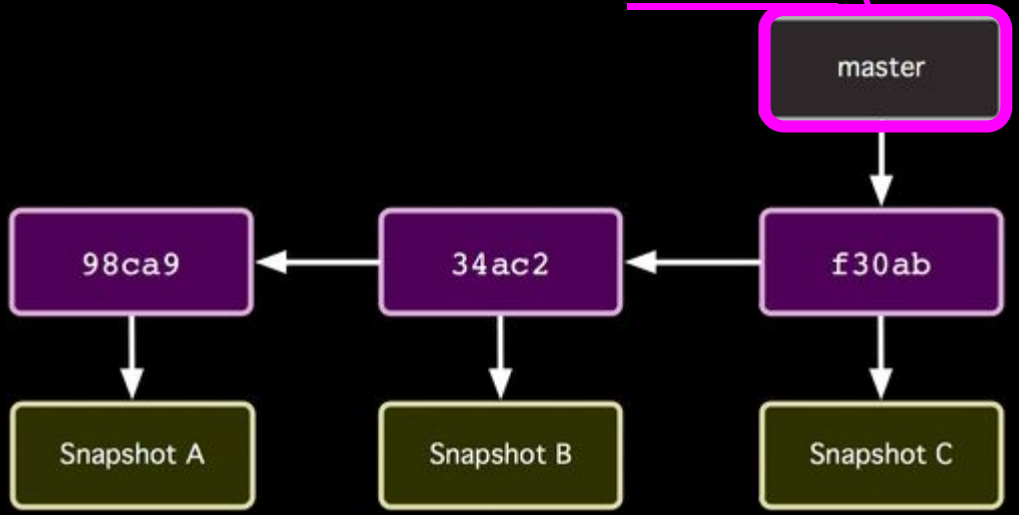
How the repos is organized

Commits (from oldest to newest; hashes as commit IDs)



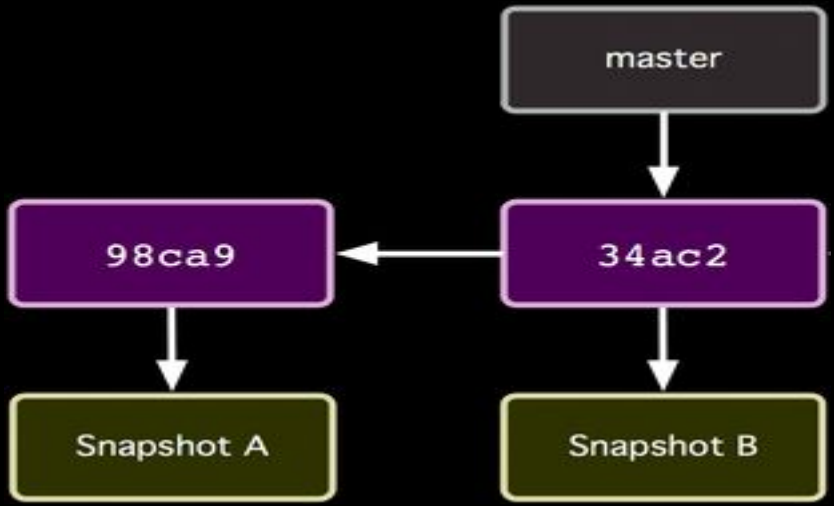
How the repos is organized

Branch (last commit)

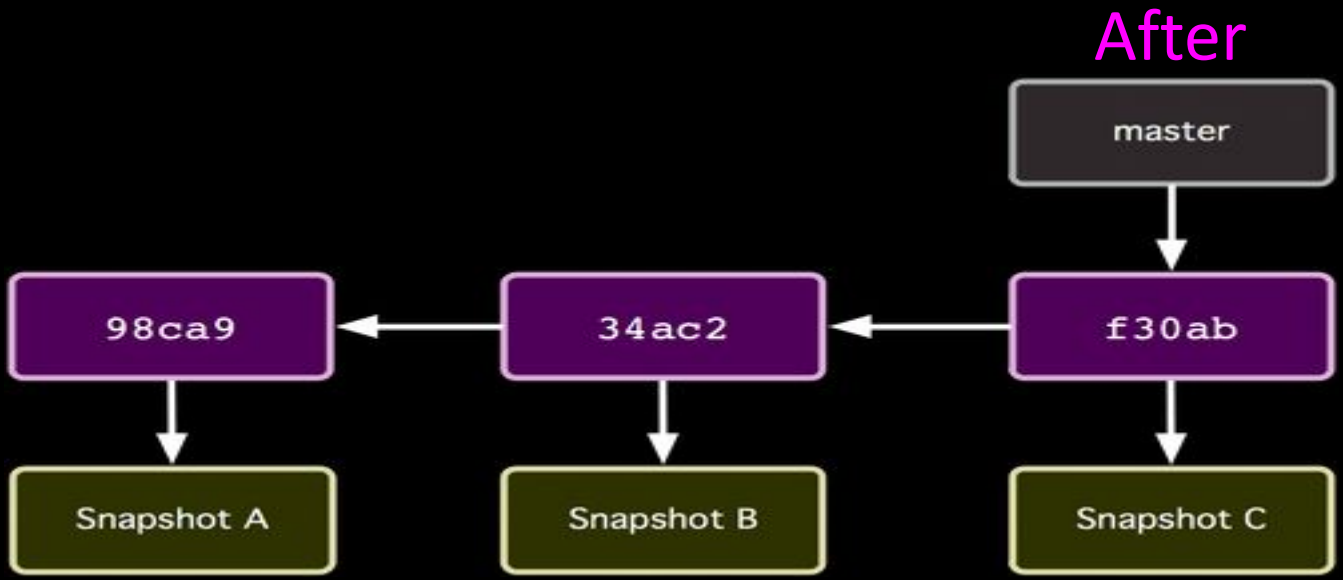


How commit works

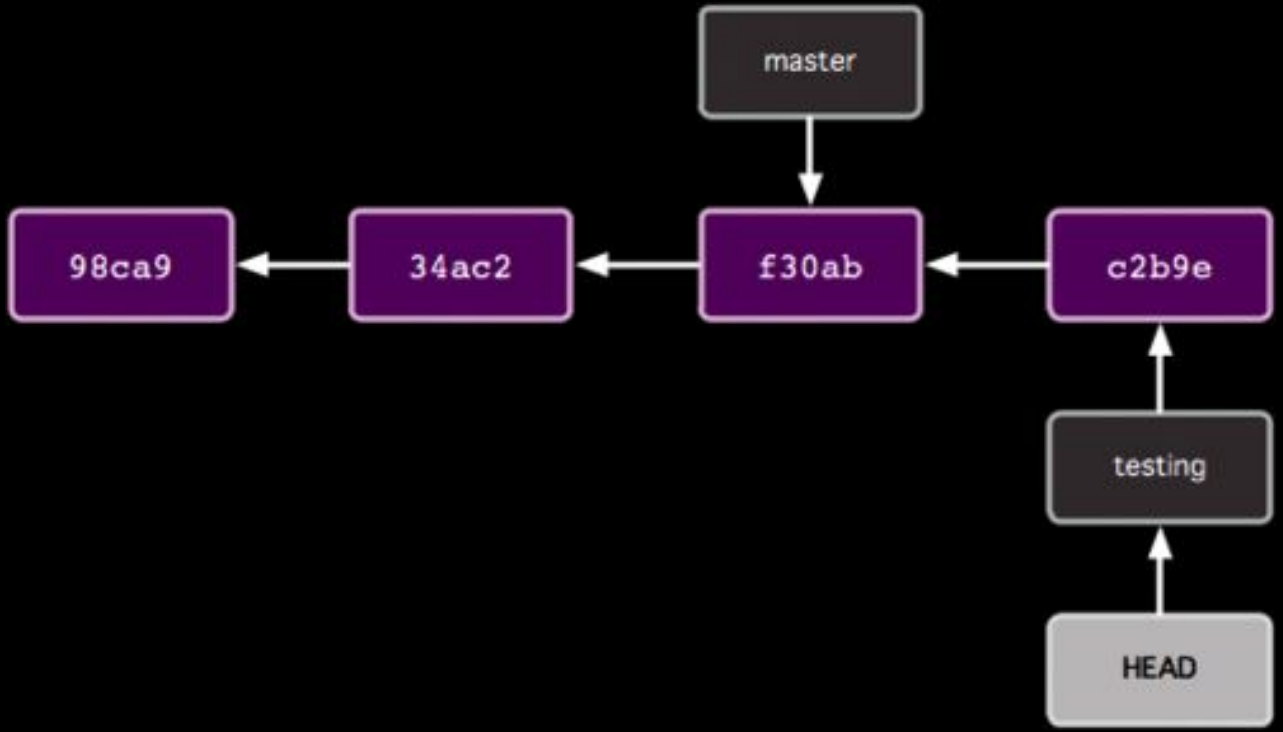
Before



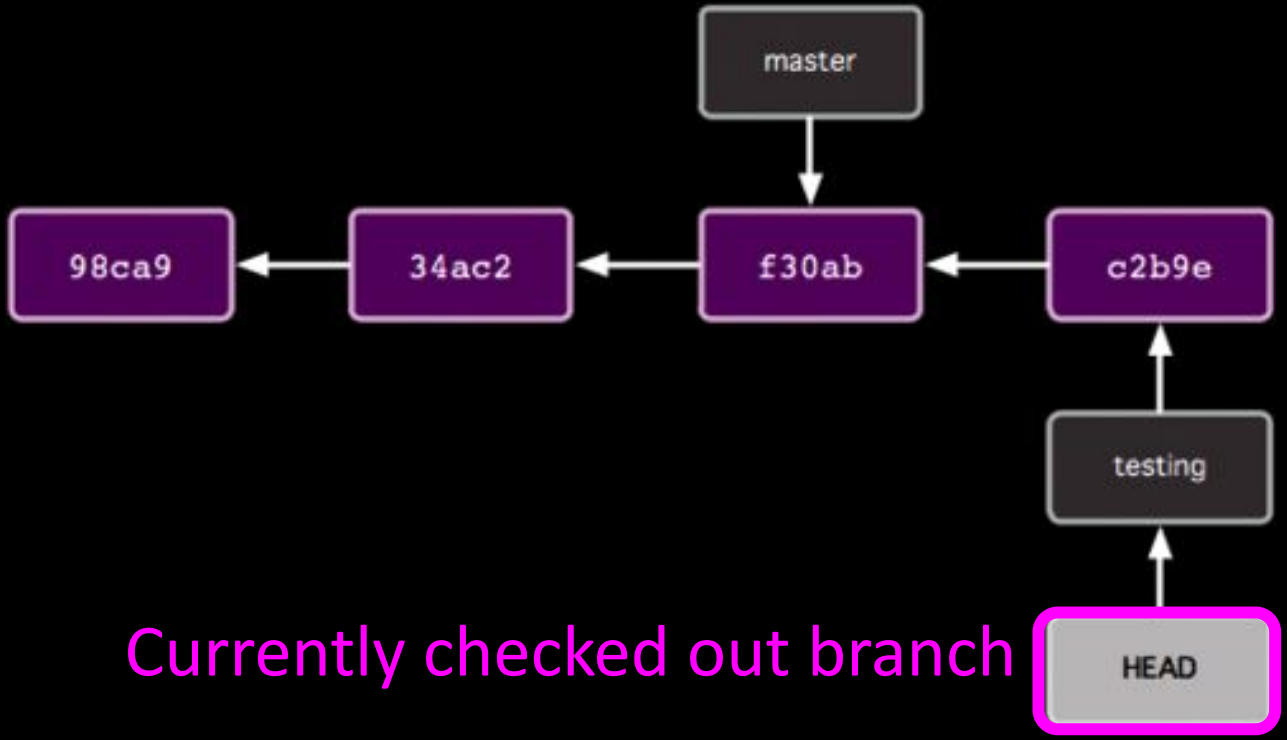
How commit works



Organization with two branches



Organization with two branches

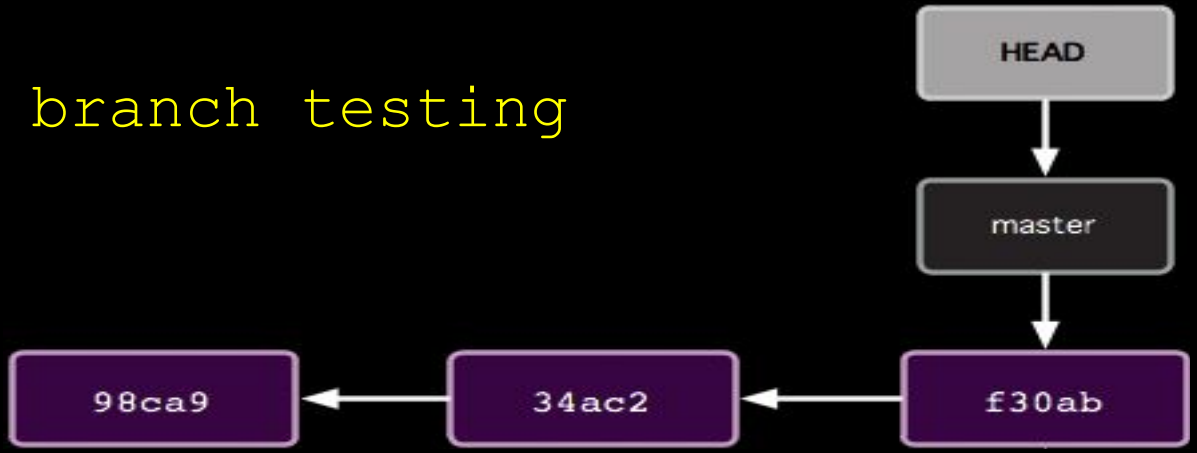


Common Workflow

1. Create temp local branch
2. Checkout temp branch
3. Edit/Add/Commit on temp branch
4. Checkout master branch
5. Pull to update master branch
6. Merge temp branch with updated master
7. Delete temp branch
8. Push to update server repos

How git branch works

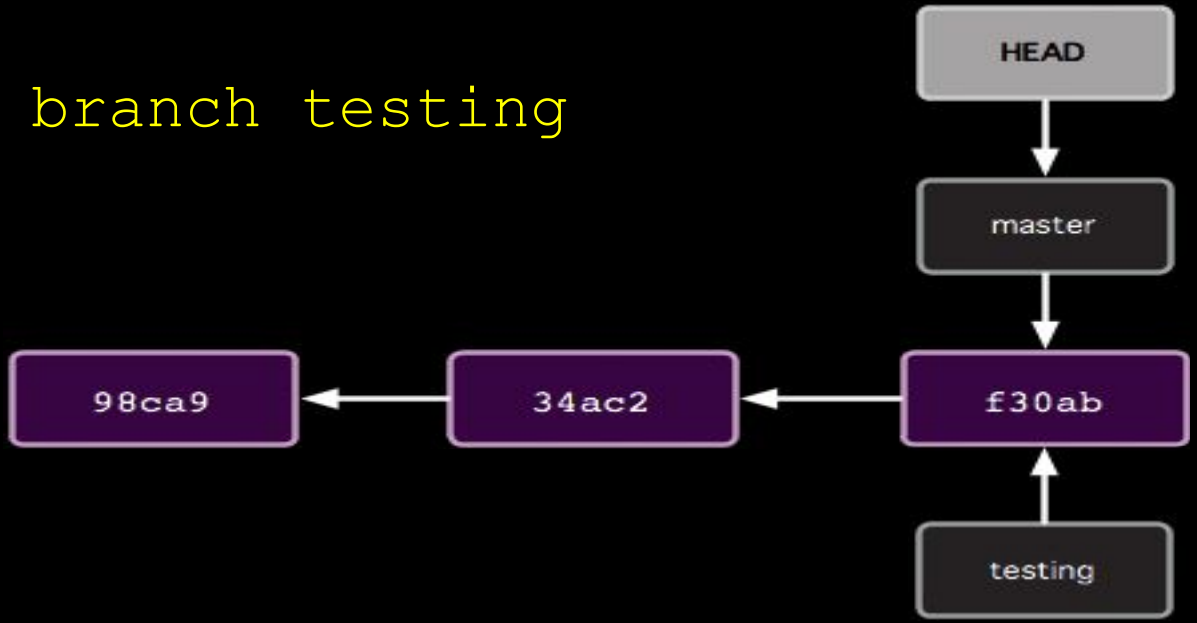
```
$ git branch testing
```



Before

How git branch works

```
$ git branch testing
```



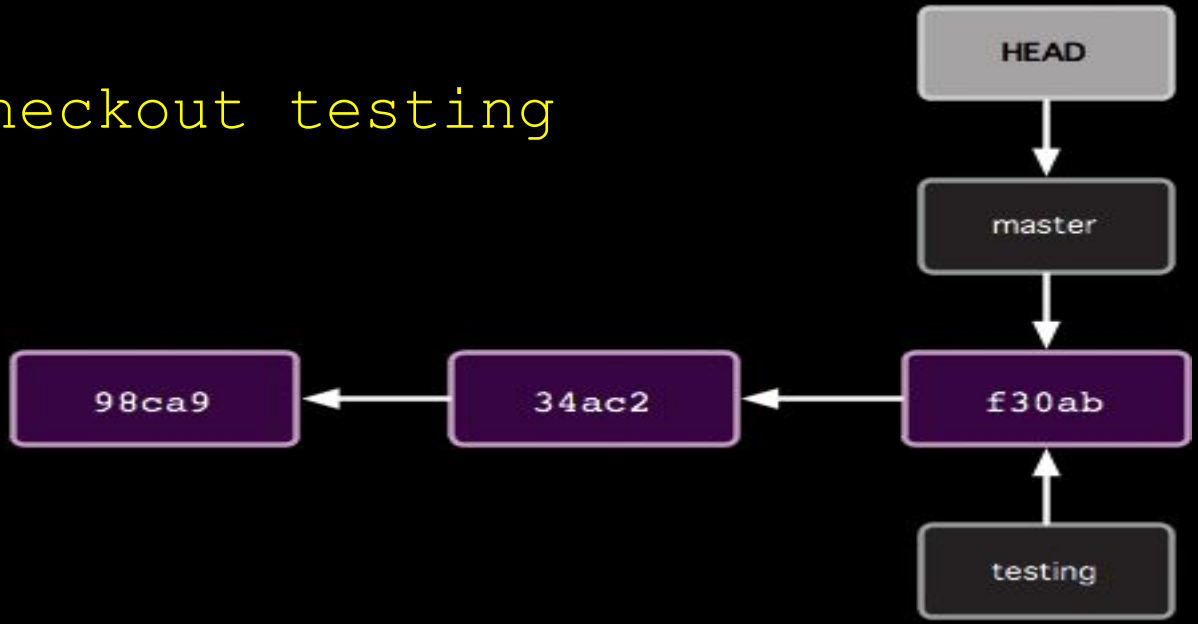
After

Common Workflow

1. Create temp local branch
2. Checkout temp branch
3. Edit/Add/Commit on temp branch
4. Checkout master branch
5. Pull to update master branch
6. Merge temp branch with updated master
7. Delete temp branch
8. Push to update server repos

How git checkout works

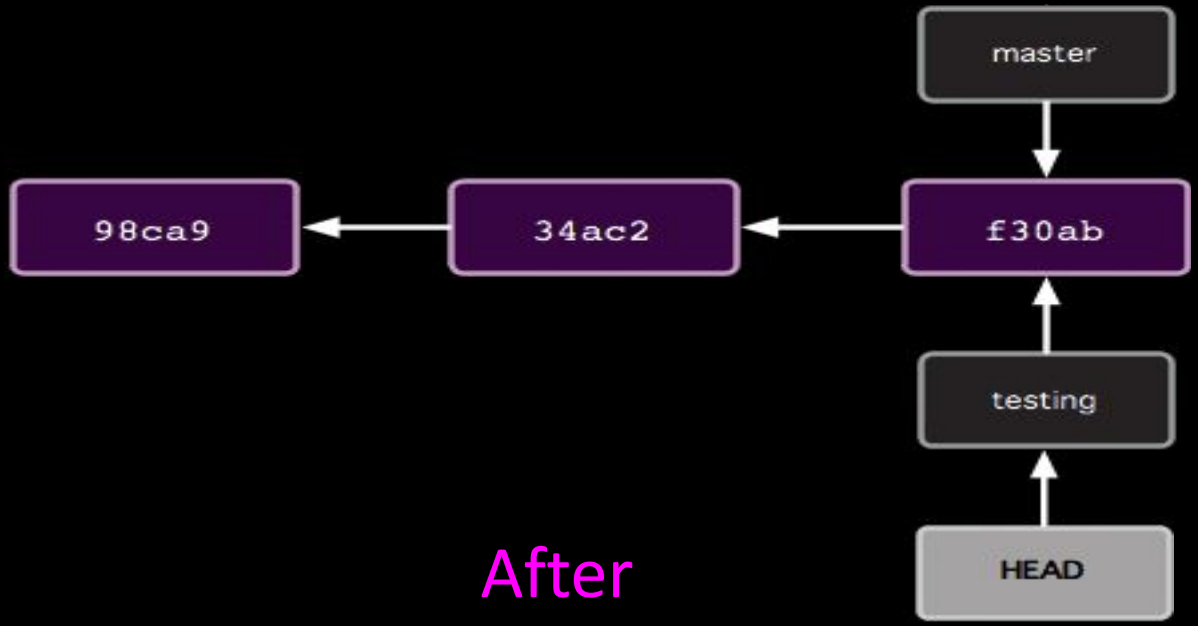
```
$ git checkout testing
```



Before

How git checkout works

```
$ git checkout testing
```



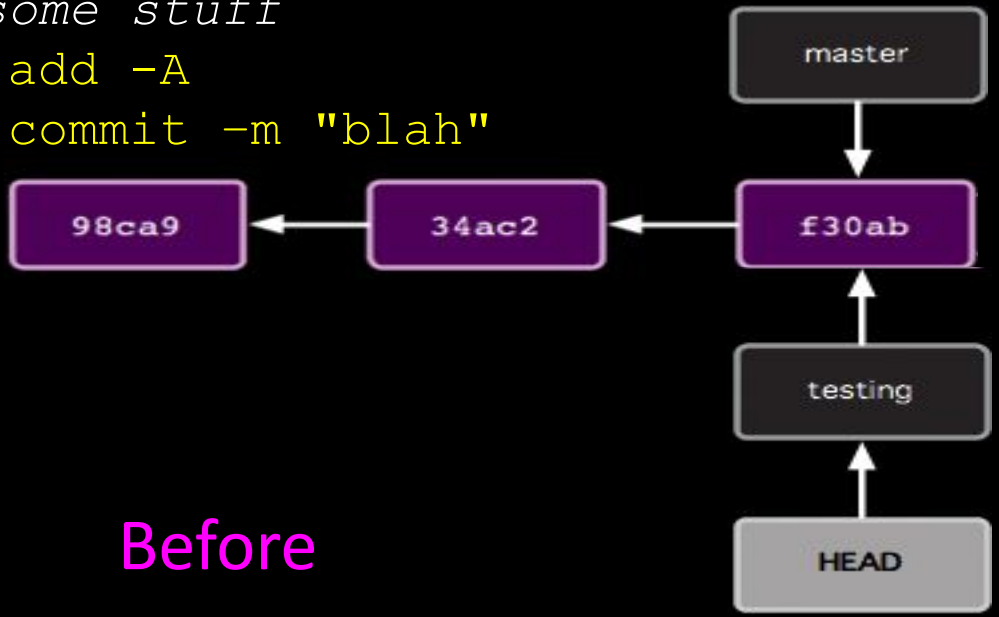
After

Common Workflow

1. Create temp local branch
2. Checkout temp branch
3. Edit/Add/Commit on temp branch
4. Checkout master branch
5. Pull to update master branch
6. Merge temp branch with updated master
7. Delete temp branch
8. Push to update server repos

How git commit works with multiple branches

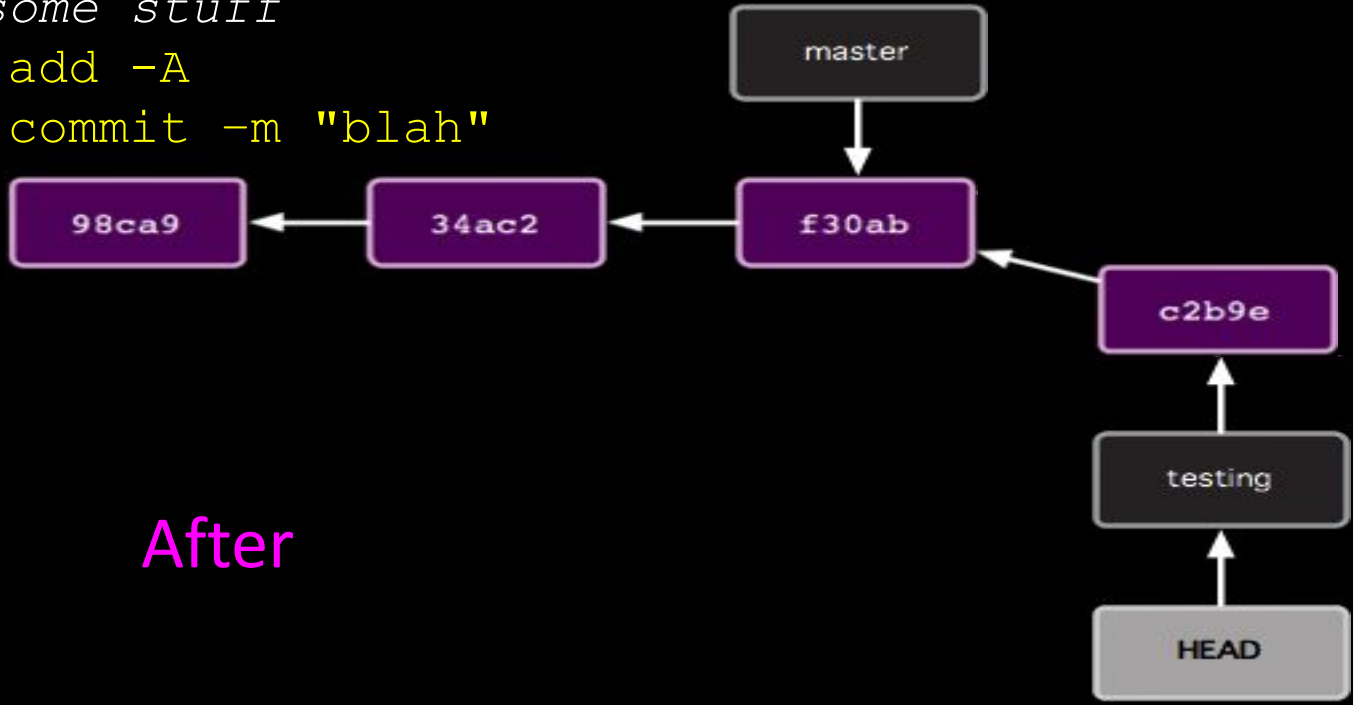
```
Edit some stuff
$ git add -A
$ git commit -m "blah"
```



Before

How git commit works with multiple branches

```
Edit some stuff
$ git add -A
$ git commit -m "blah"
```



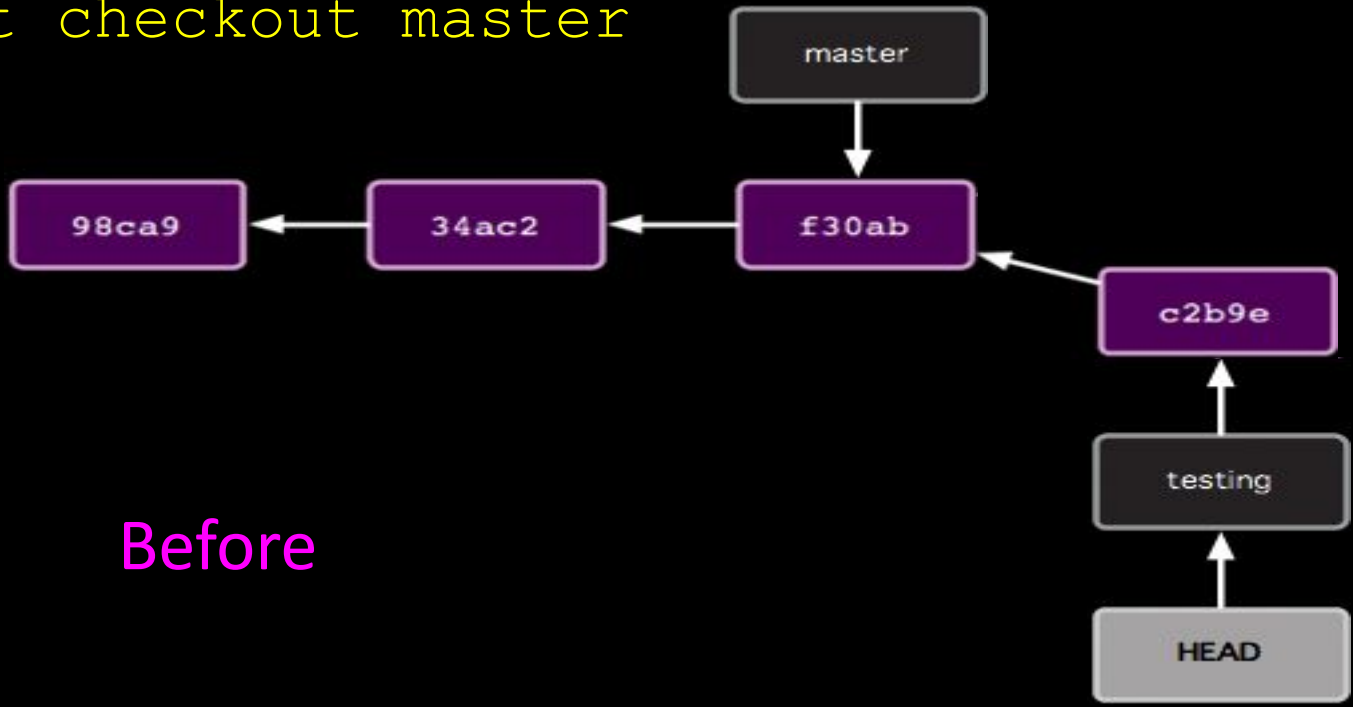
After

Common Workflow

1. Create temp local branch
2. Checkout temp branch
3. Edit/Add/Commit on temp branch
4. Checkout master branch
5. Pull to update master branch
6. Merge temp branch with updated master
7. Delete temp branch
8. Push to update server repos

How git checkout works

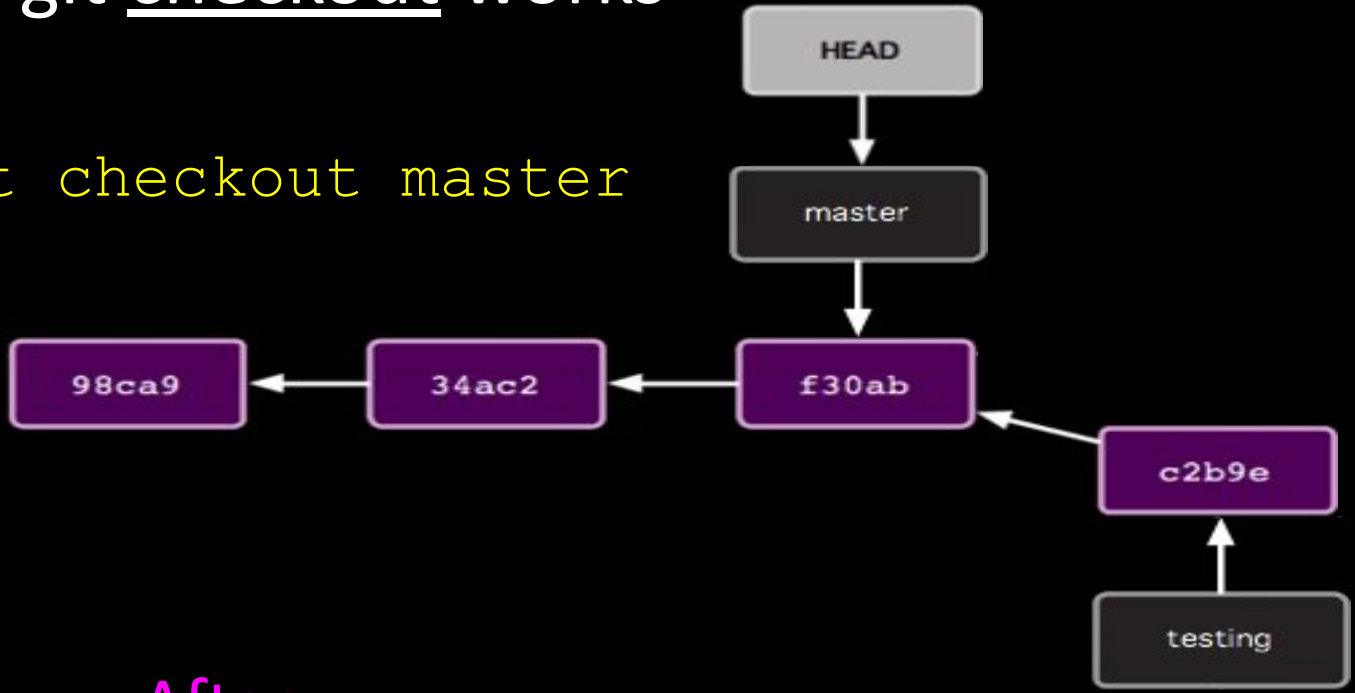
```
$ git checkout master
```



Before

How git checkout works

```
$ git checkout master
```



After

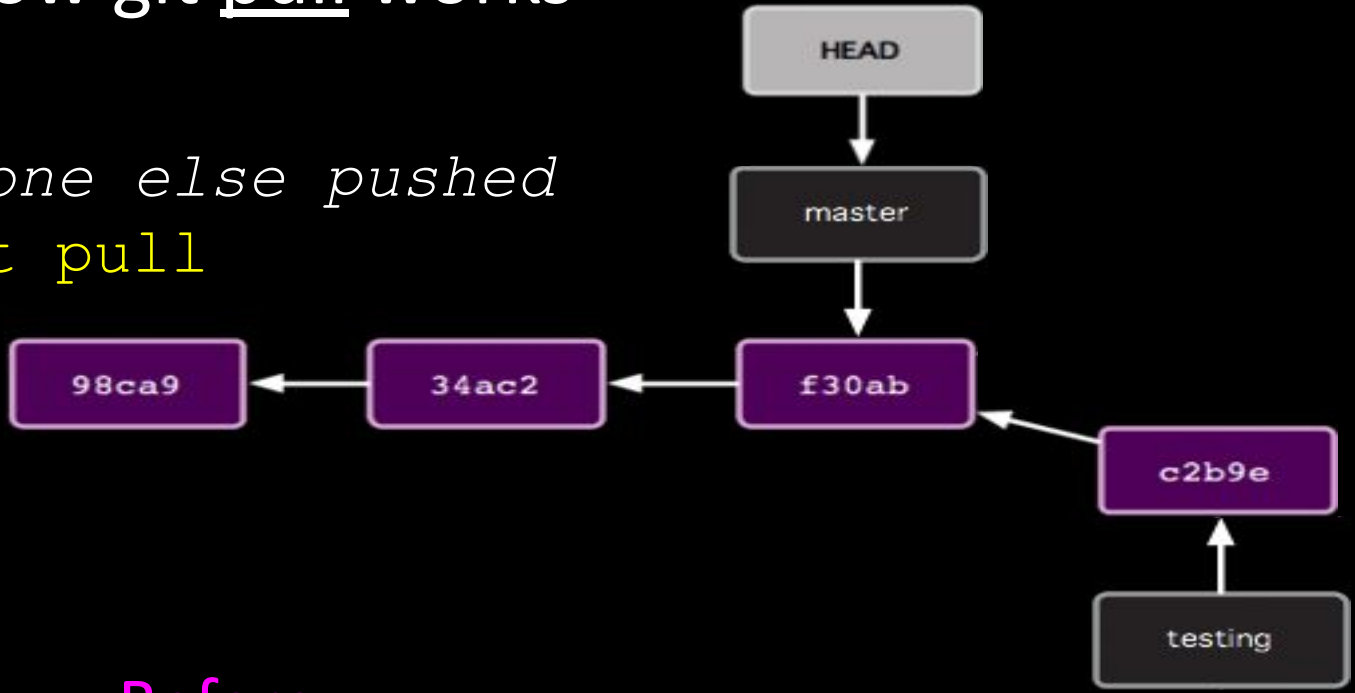
Common Workflow

1. Create temp local branch
2. Checkout temp branch
3. Edit/Add/Commit on temp branch
4. Checkout master branch
5. Pull to update master branch
6. Merge temp branch with updated master
7. Delete temp branch
8. Push to update server repos

How git pull works

Someone else pushed

```
$ git pull
```

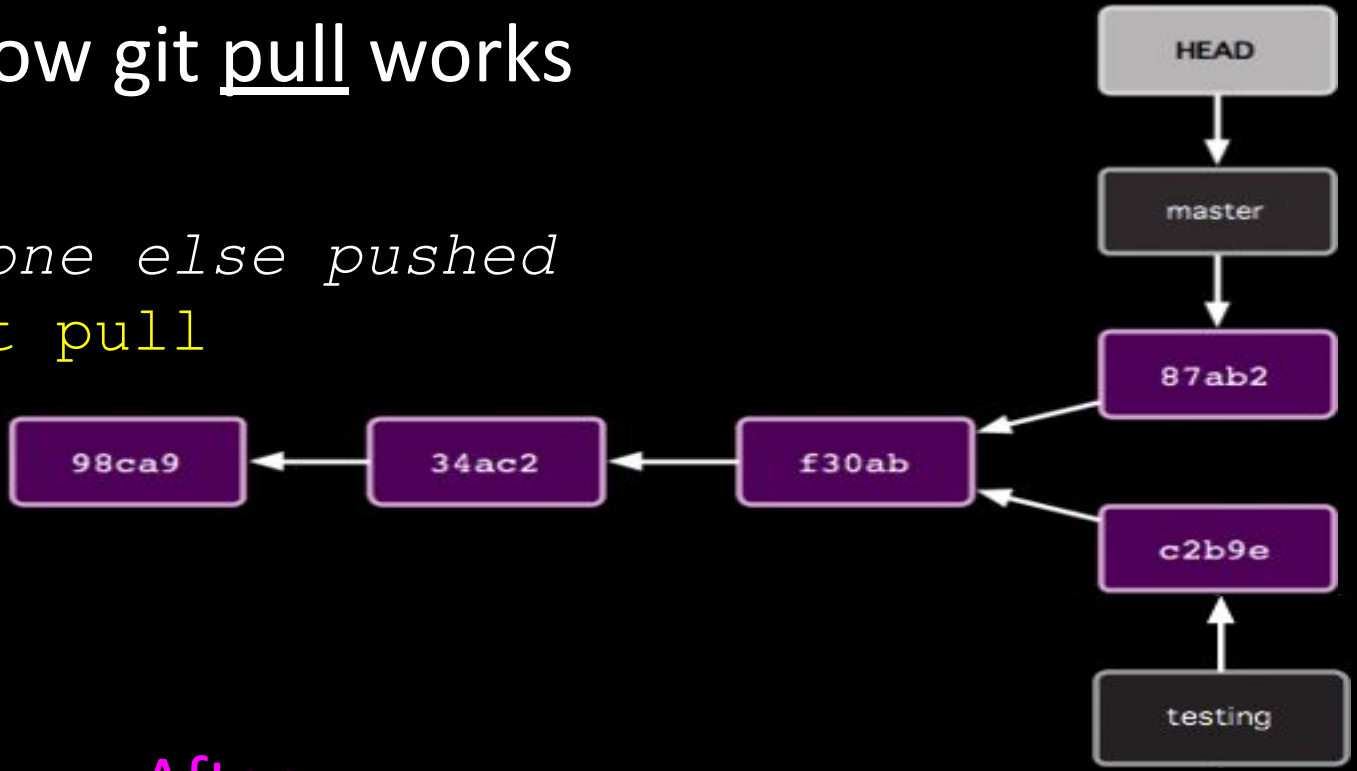


Before

How git pull works

Someone else pushed

```
$ git pull
```



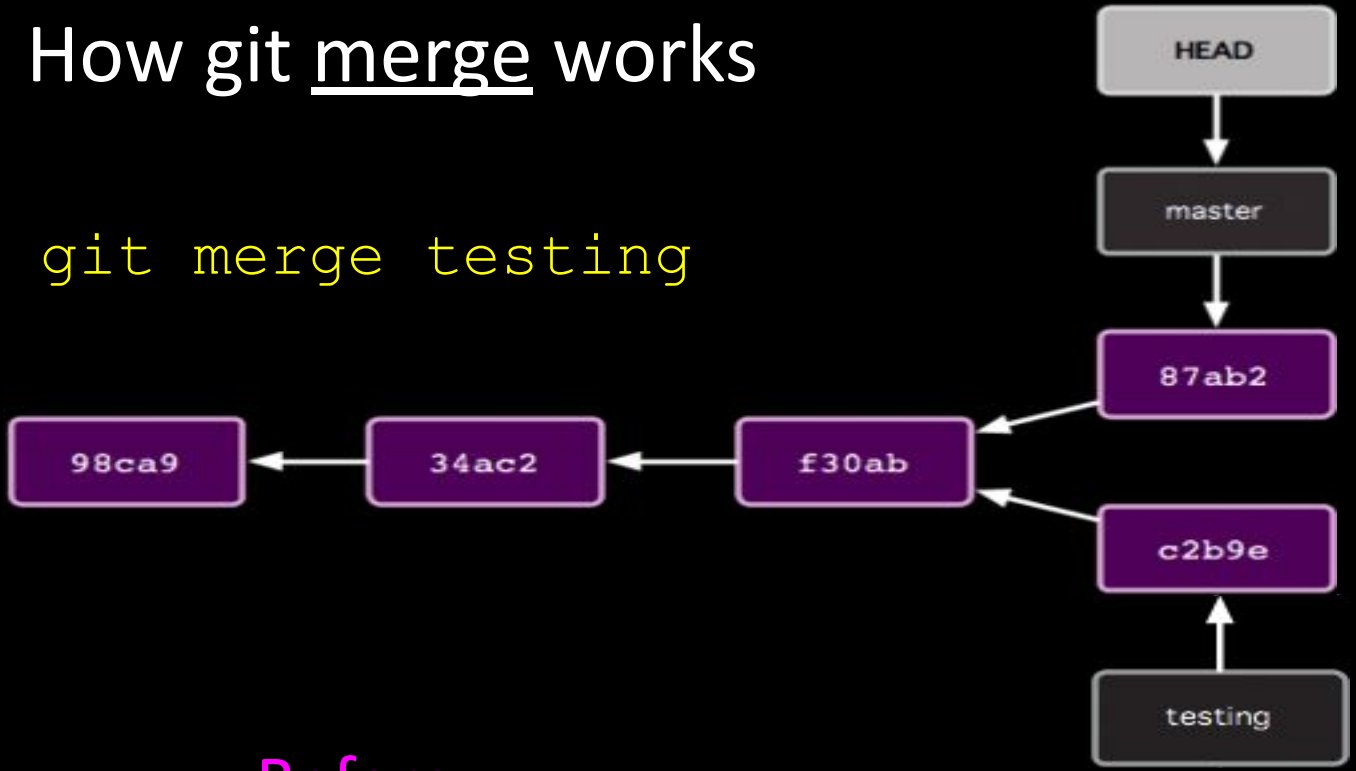
After

Common Workflow

1. Create temp local branch
2. Checkout temp branch
3. Edit/Add/Commit on temp branch
4. Checkout master branch
5. Pull to update master branch
6. Merge temp branch with updated master
7. Delete temp branch
8. Push to update server repos

How git merge works

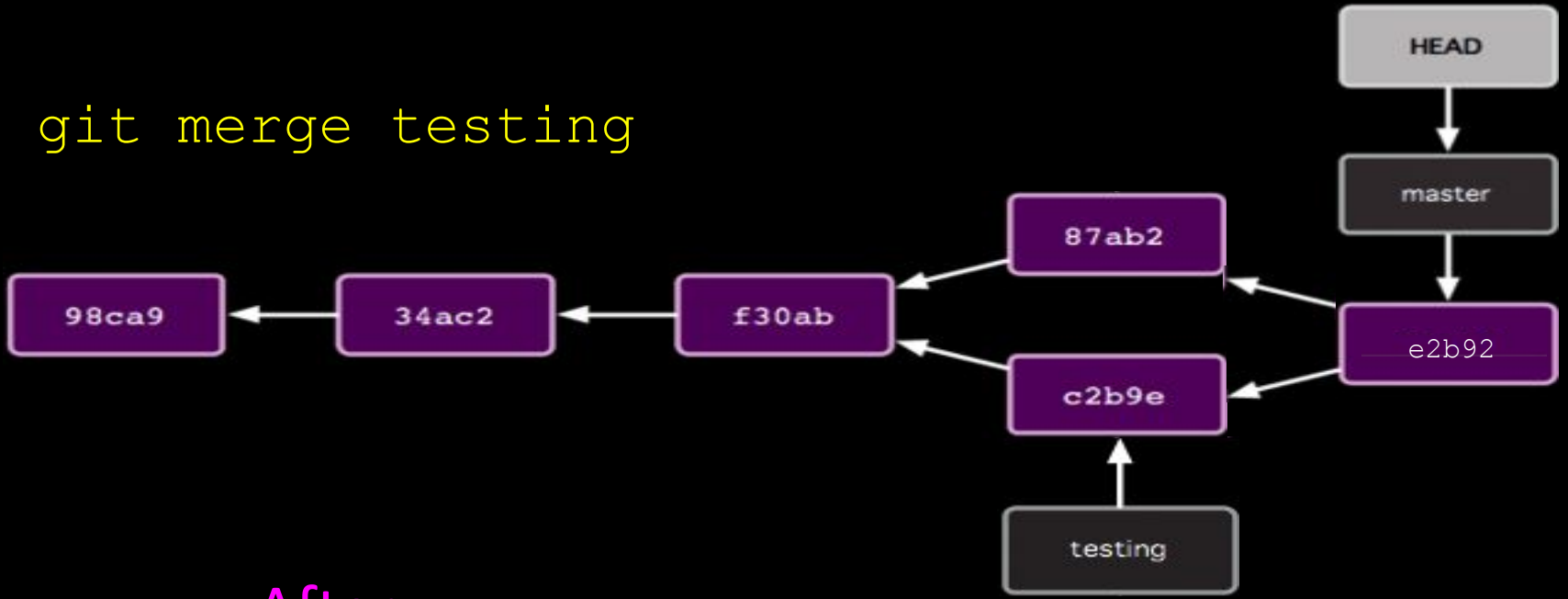
```
$ git merge testing
```



Before

How git merge works

```
$ git merge testing
```



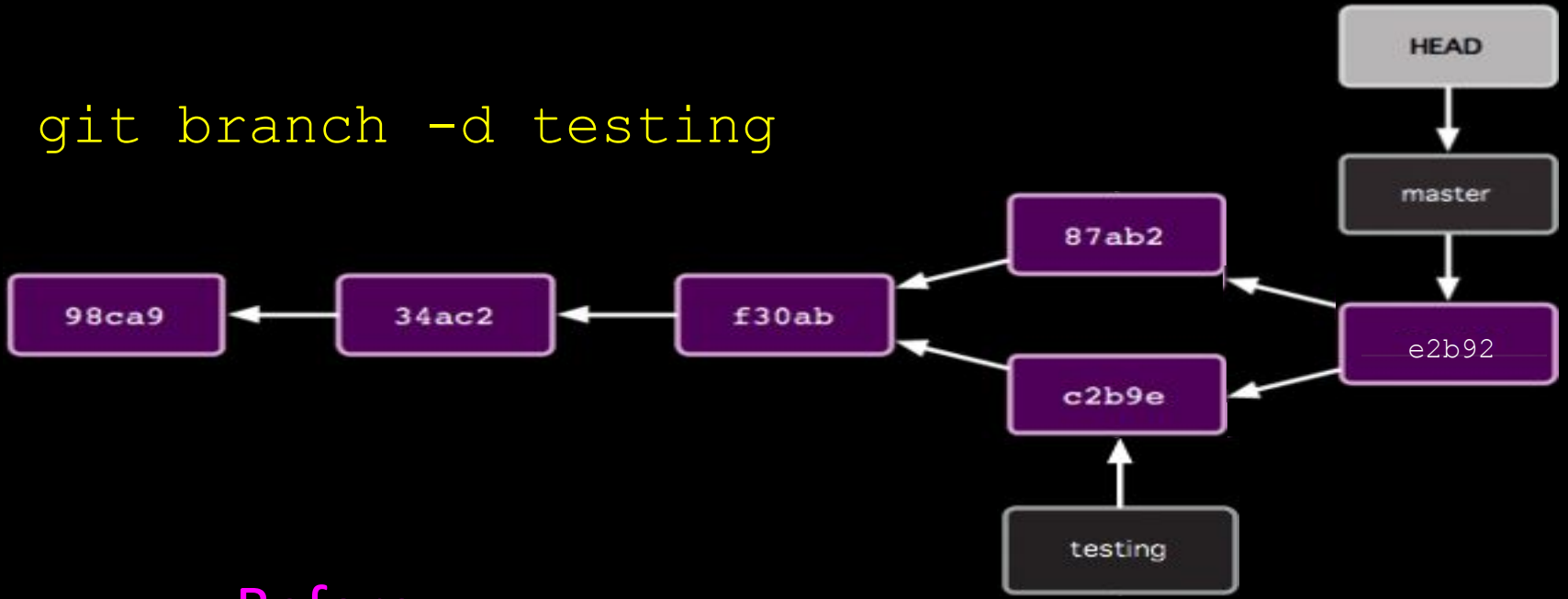
After

Common Workflow

1. Create temp local branch
2. Checkout temp branch
3. Edit/Add/Commit on temp branch
4. Checkout master branch
5. Pull to update master branch
6. Merge temp branch with updated master
7. Delete temp branch
8. Push to update server repos

How to delete branches

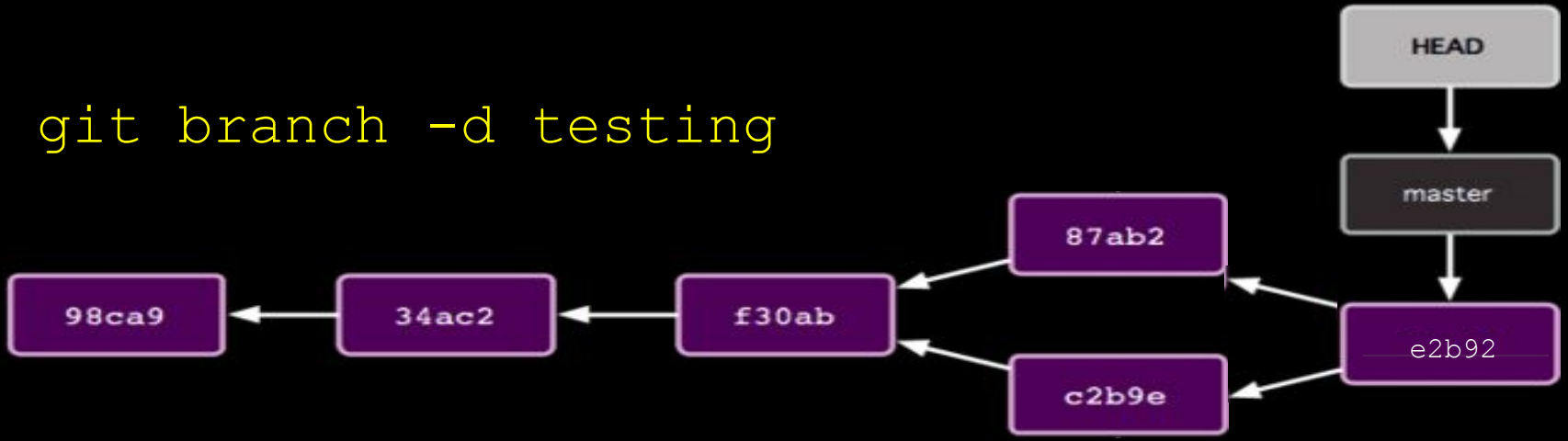
```
$ git branch -d testing
```



Before

How to delete branches

```
$ git branch -d testing
```



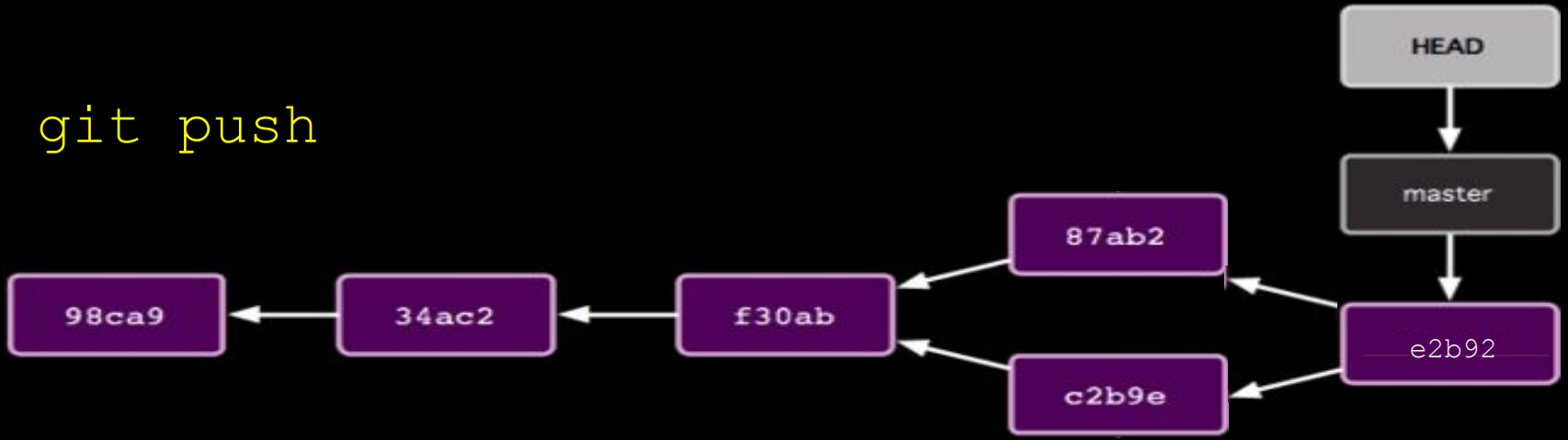
After

Common Workflow

1. Create temp local branch
2. Checkout temp branch
3. Edit/Add/Commit on temp branch
4. Checkout master branch
5. Pull to update master branch
6. Merge temp branch with updated master
7. Delete temp branch
8. Push to update server repos

How git push works

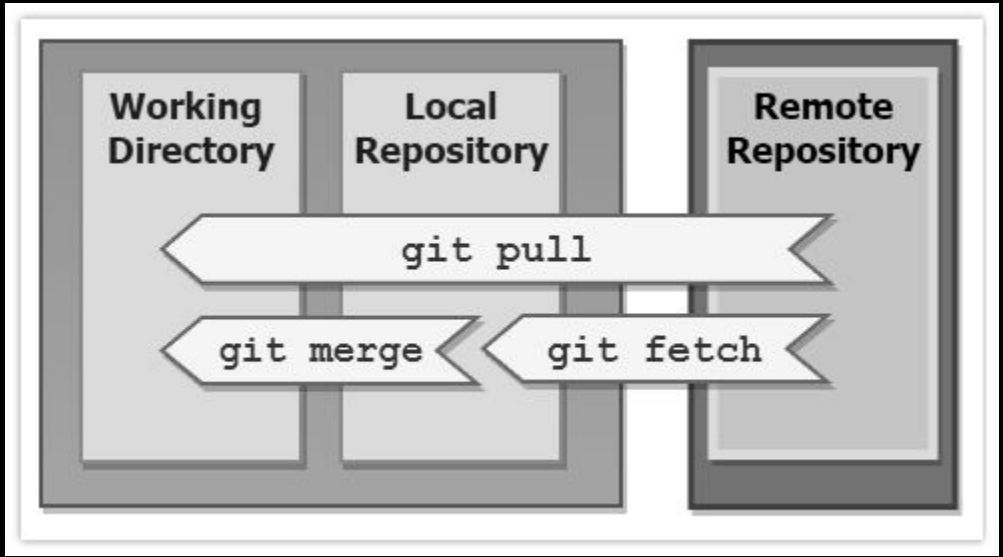
```
$ git push
```



Should update server repos

(if no one else has pushed commits to master branch since last pull)

Pull vs Fetch



Tips

- git output contains lots of hints
  - git status is your friend!
- Merging may not be as easy as I showed
  - E.g.: Multiple collabs updated same parts of file
  - Resolve Conflicts then commit
- Pull before starting temp branch
- Team communication important!

Origin vs Upstream

Fork around and find out!

Meet Alex 👩‍💻

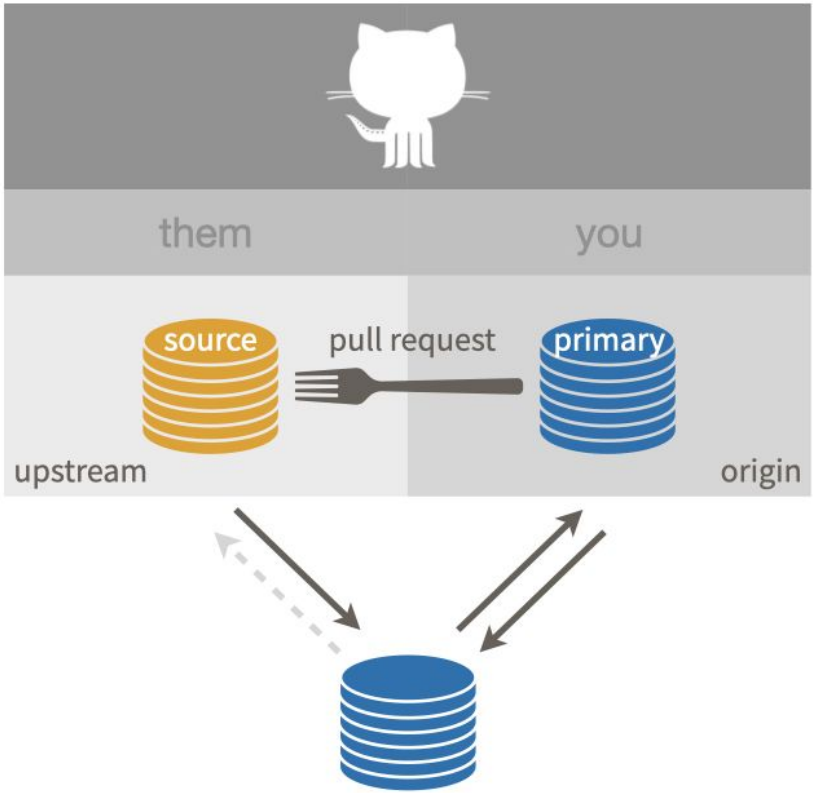
Alex finds a cool open-source project on GitHub called "ITI App".
She wants to improve it by adding a new feature.

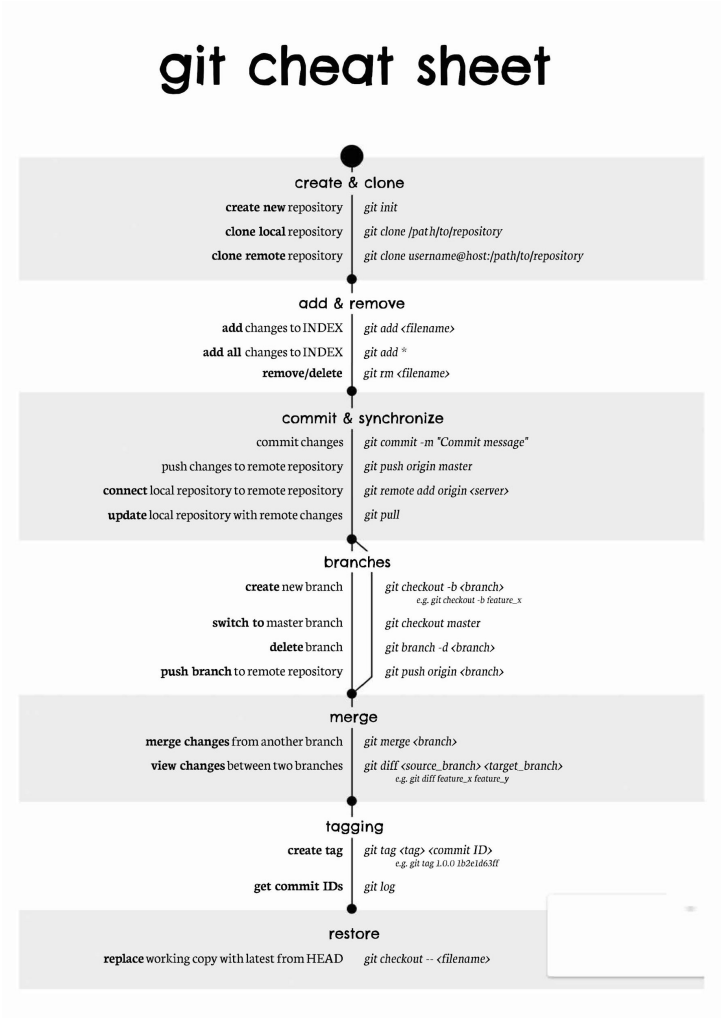
But there's a catch...
She **can't push** directly to the project, she's not a collaborator.

So, what does Alex do?

✅ She **forks** the project, now she has her own copy (on her GitHub).
🛠️ She works on her copy, pushes changes to her **origin**.
🔁 Later, she creates a **pull request** to the original project.

That original project? That's her **upstream**.



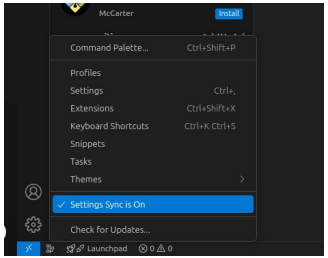


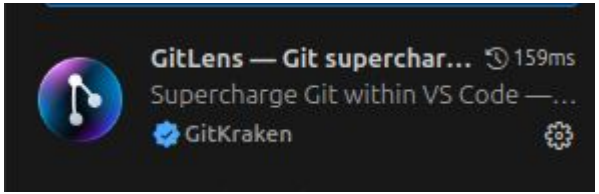
git cheat sheet

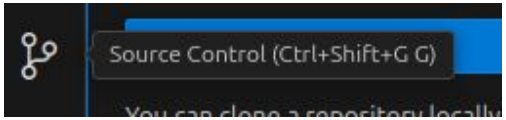
https://teacher.benpaddlejones.com/files/git_cheat_sheet.pdf

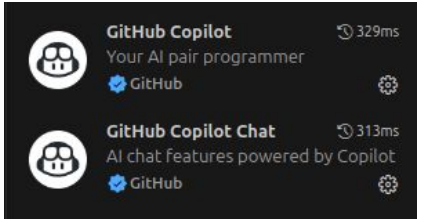
Git & VS Code

- VS Code Source Control tab (already there)
- GitHub Copilot (now free for all) [optional]
- To set-up & screenshot
  - Sync Editor Settings
  - GitLens (we also call it blame)
  - The following game levels









GAME TIME!!!!

https://ohmygit.org

Do one level of each category.

Note: must do the merging conflict level though

Developer Student Clubs

Thank you.
Very much!