



Git Introduction



Did you finish pre-class work?



Students, drag the icon!



Pear Deck Interactive Slide
Do not remove this bar



In case of fire



1. git commit



2. git push



3. leave building

Did you install Git?



Students, drag the icon!

REINVENT YOURSELF



Pear Deck Interactive Slide
Do not remove this bar

Did you create your Github account?



Students, drag the icon!



Pear Deck Interactive Slide
Do not remove this bar

Git Journey



- *Git introduction
- *Basic Git operation
- *Remote repository



- *Branches
- *Contribution to the remote repository



Hands-on



git



GitHub

Table of Contents



- ▶ What is version control?
- ▶ What is Git?
- ▶ Basic Git Operations
- ▶ Remote Repository

What do you know about Git? »

Let's discuss about Git



What is Git?



Git is a free, open source
distributed version control system

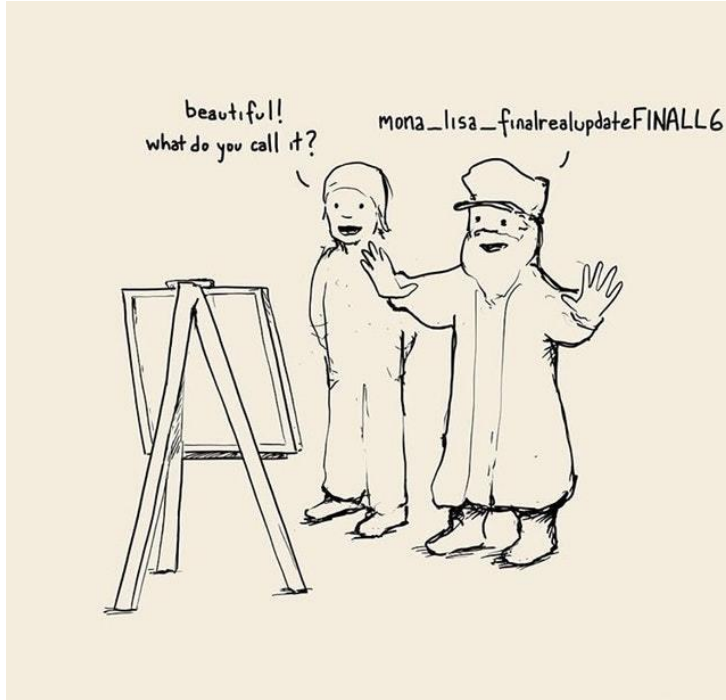




1

What's Version Control?

What's Version Control?



Students, write your response!



What's Version Control?

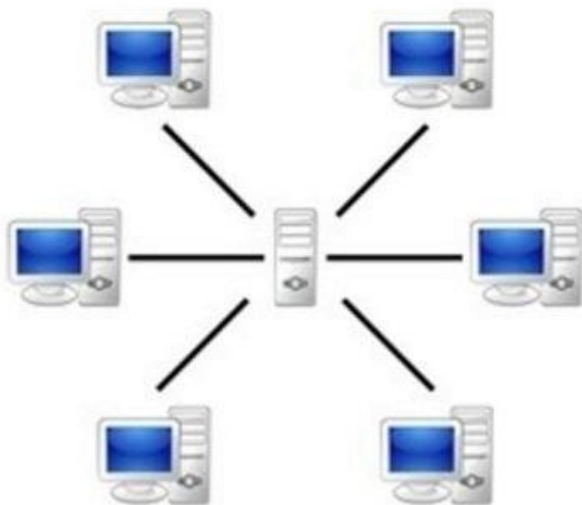
- Track changes
- Undo / Redo
- Time Travel
- Collaborative development
- Compare and Blame
 - ◆ What changed
 - ◆ When it changed
 - ◆ Why it changed
 - ◆ Who changed it



Version Control Systems

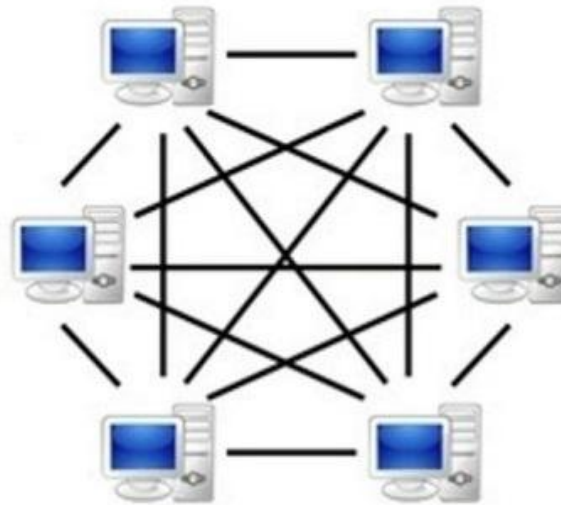
- **Centralized**

You need to be connected to the server



- **Distributed**

You can work while offline





What's Version Control?



Your Daily Tasks

- **Create** things
- **Save** things
- **Edit** things
- Save the things **again**

What's Version Control?



Time



Day - 1

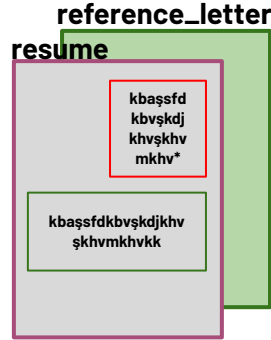
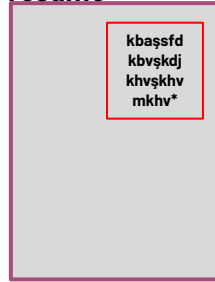
Day - 2

Day - 3



Your
folder

resume



VCS

Add "about"
section to resume

resume.doc



modified

Create "reference
letter" file

reference_letter.doc



created

Add photo and
change layout

resume.doc

modified

Add "education"
section to resume

resume.doc

modified



What's Version Control?



Version Control Systems (VCS)

- **Tracks** and **records** changes to files over time
- Can track any type of file, but most commonly used for code
- Contains extra information such as date, author, and a message explaining the change



What's Version Control?



Benefits of Version Control Systems (VCS)

- Can **retrieve** previous version of files at any time
- Retrieve files that were accidentally deleted
- Can be used **locally**, or **collaboratively** with others



2

What is Git?



What is Git?

- Content Tracker
- Distributed Version Control System (VCS)
- Linus Torvalds





► Why do we need Git?

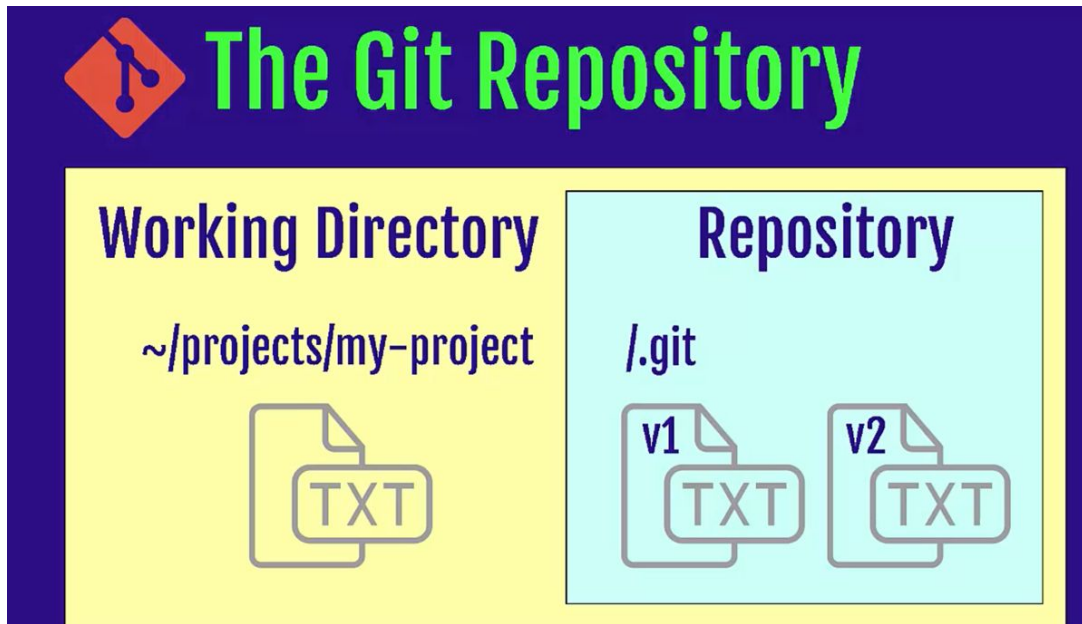
- Backup/Versioning/History
- Undo Changes
- Comparing
- Collaboration and Teamwork
- Code Review



Git Repository

What is a repo?

- A directory or storage space where your projects can live.
- Local / Remote





3

Basic Git Operations



Kahoot!





Git Repository

→ Check if you have git in your computer

```
git --version
```

→ Configuration

```
git config --global user.name "<name>"
```

```
git config --global user.email "<email>"
```

```
git config --global core.editor "<text editor>"
```

```
git config --list
```




Git Repository

→ Create a local repo

```
git init
```

→ Get the content of a remote repo

```
git clone <url>
```



Git Repository

→ See the commands

```
git help
```

→ See the status of your repo

```
git status
```



.gitignore

- Specifies intentionally **untracked files** that Git should ignore.
- Files already tracked by Git are not affected!
- Each line specifies a pattern.

file

folder/

***.log**

!important/*.log

Workflow



Working Directory

Where you work.
Create new files,
edit files delete
files etc.



Staging Area (Index)

Before taking a
snapshot, you're
taking the files to
a stage. Prepare
files to be
committed.



Repository

Committed
snapshots of your
project will be
stored here with
a full version
history.





▶ Track a new file

- Create a new file in our project folder

lemonade.txt

- Edit this file

- Check the status of our project

git status



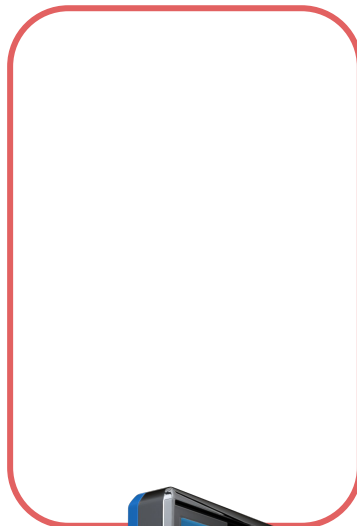
Create a new file



Working Directory



Staging Area (Index)



Repository



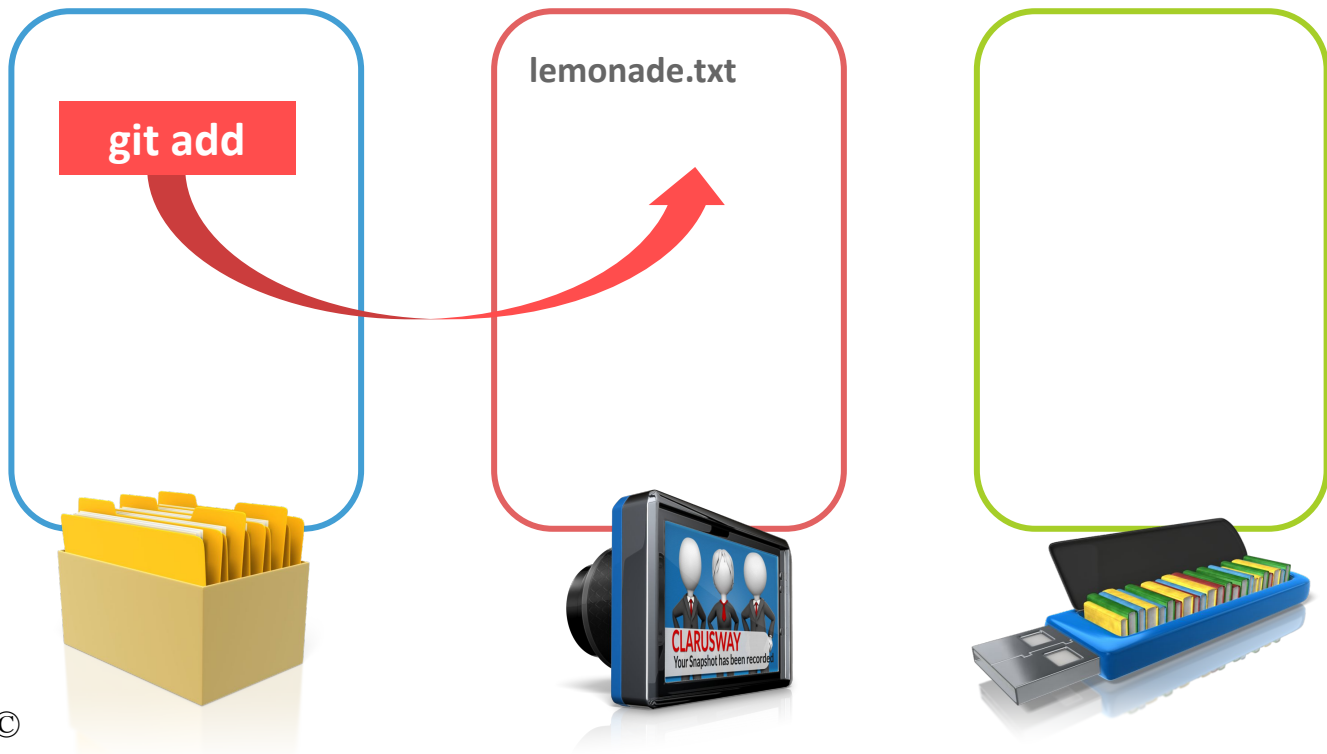


Track/stage a file

Working Directory

Staging Area (Index)

Repository





Stage file options

→ Stage one file

```
git add <filename>
```

→ Stage all files

```
git add .
```




Record the current state

- Stash the changes in a dirty working directory away

```
git stash
```

- Bring stashed changes back to the working directory

```
git stash apply <stash>
```

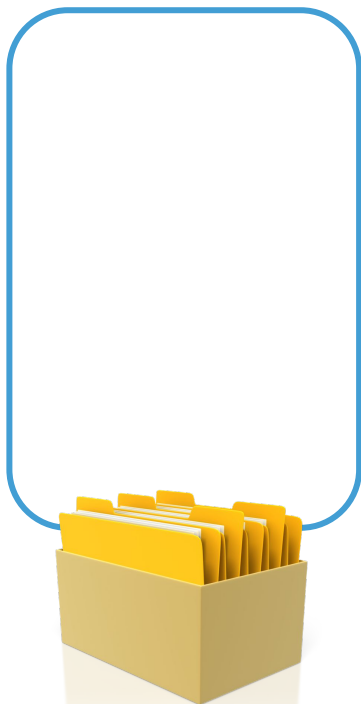
- Remove all the stash entries

```
git stash clear
```

Commit



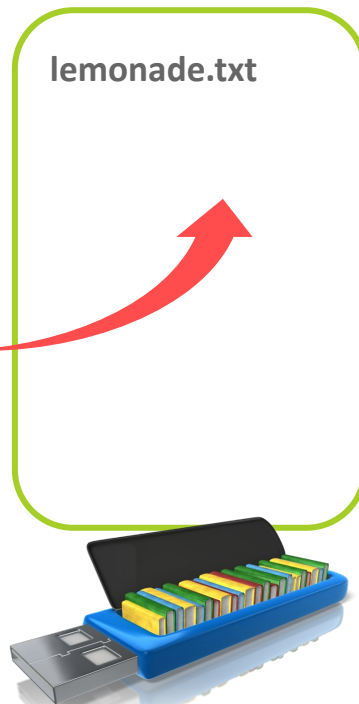
Working Directory



Staging Area (Index)



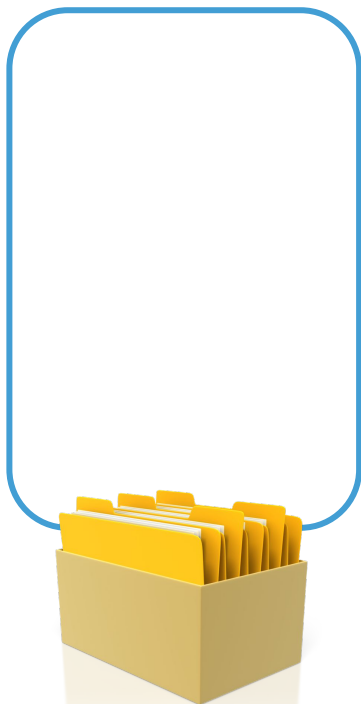
Repository



Commit



Working Directory



Staging Area (Index)



Repository





Remove from stage

Working Directory

Staging Area (Index)

Repository





Commit

- Commit the files on the stage

```
git commit -m "message"
```

- Add and commit all -tracked- files

```
git commit -am "message"
```

- Modify the most recent commit

```
git commit --amend
```

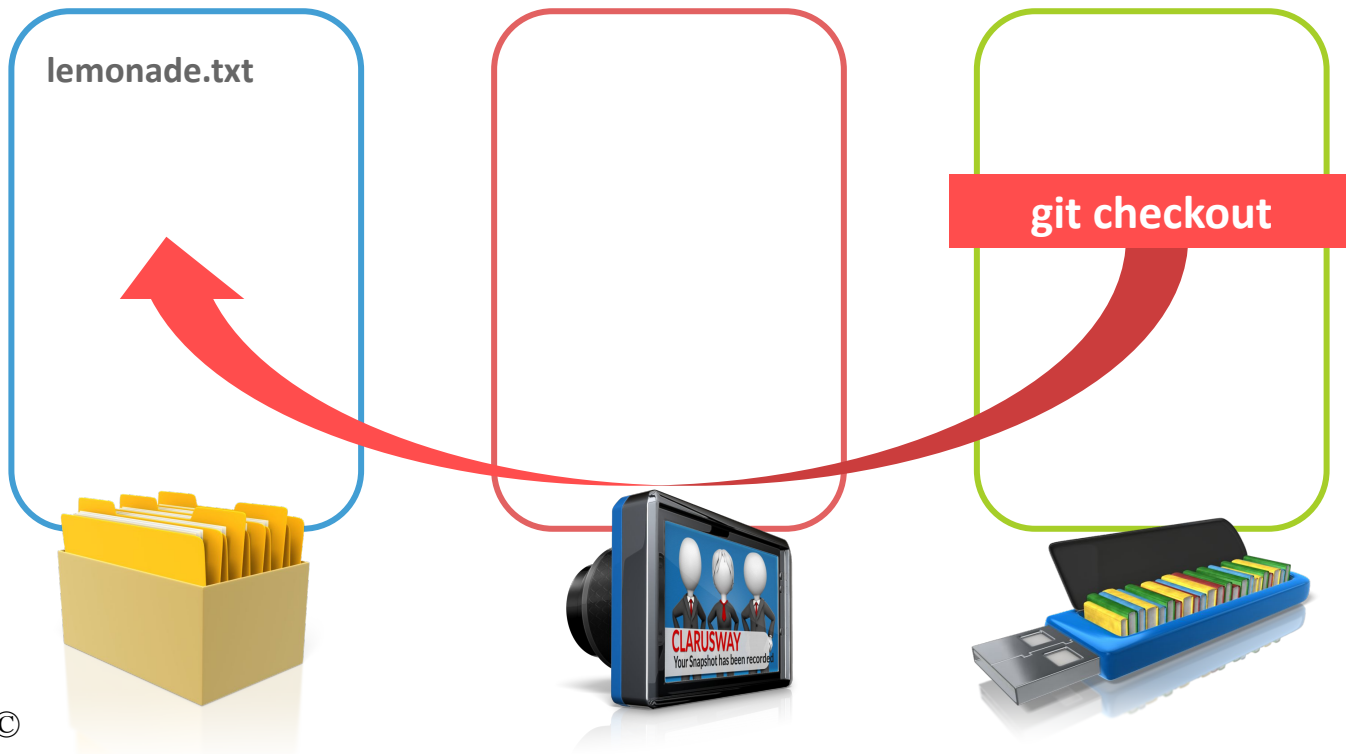


Checkout from repo

Working Directory

Staging Area (Index)

Repository



Summary



git init

git status

git add

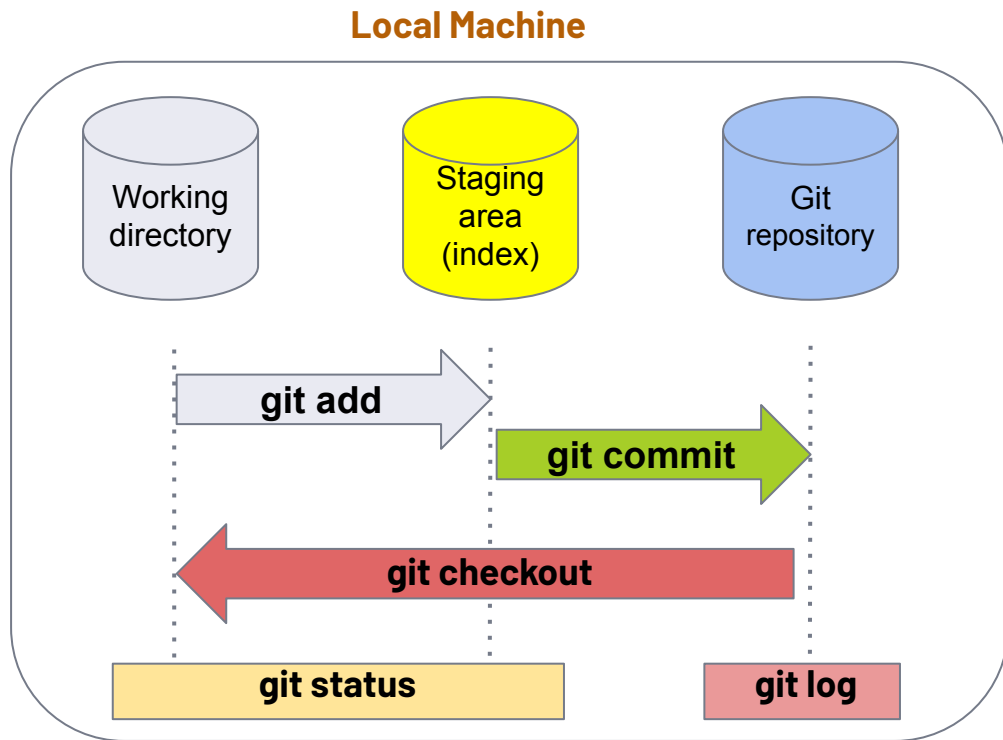
git commit

git log

git diff

git rm

git checkout





4

Remote Repository



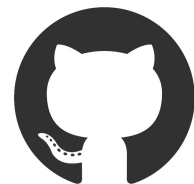
Github - Remote Repository



Git

&

GitHub



Version-control system

Repository hosting service



Github - Remote Repository



Bitbucket

+ Follow

+ I use this

Stacks	Followers	Votes
25.8K	19.2K	2.8K



GitHub

+ Follow

+ I use this

Stacks	Followers	Votes
132.1K	99.8K	10.1K



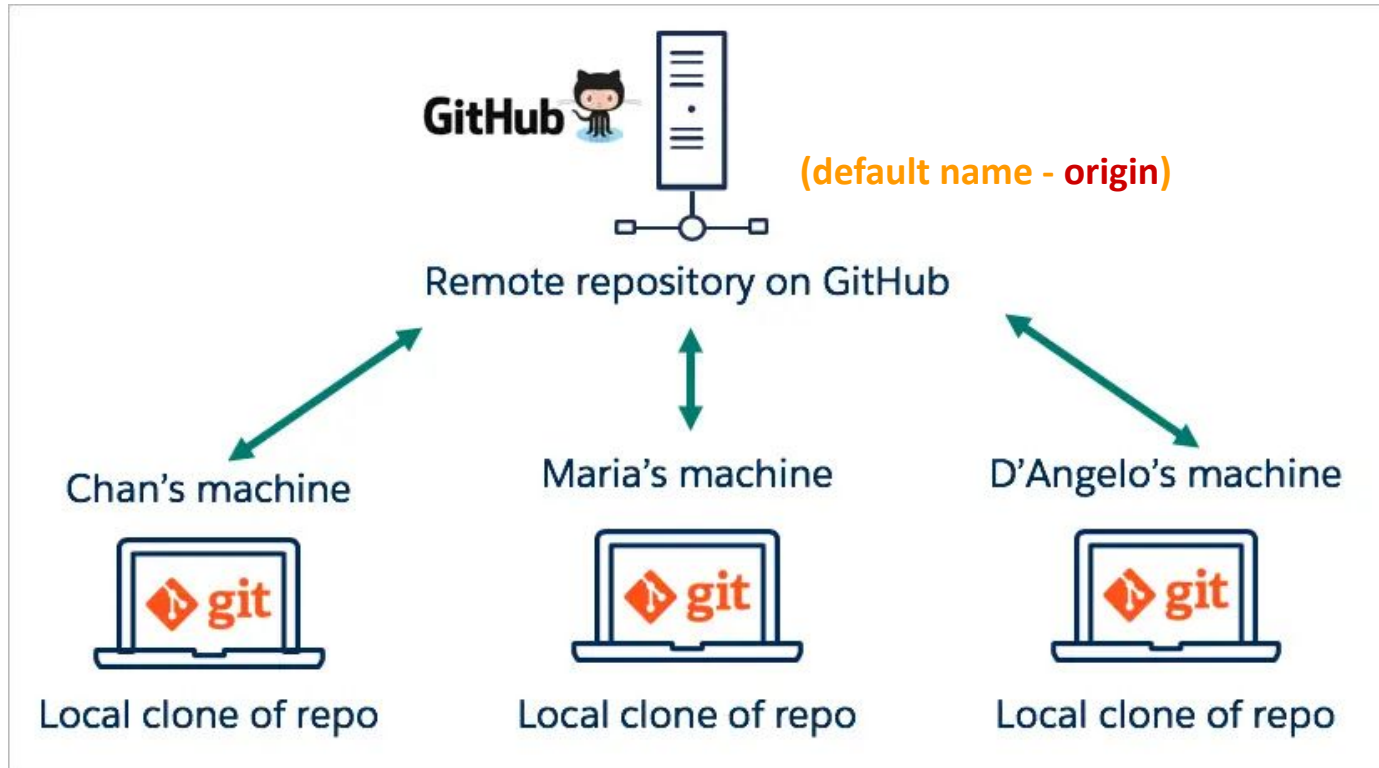
GitLab

+ Follow

+ I use this

Stacks	Followers	Votes
30.5K	23.4K	2.3K

Github - Remote Repository





Github - Remote Repository

→ Copy a repo from remote to local

```
git clone <url>
```

→ Get the latest version

```
git pull
```

→ Upload your commit

```
git push
```



Github - Remote Repository

→ Connect to remote repo

```
git remote add origin <url>
```

→ First push

```
git push -u origin master
```

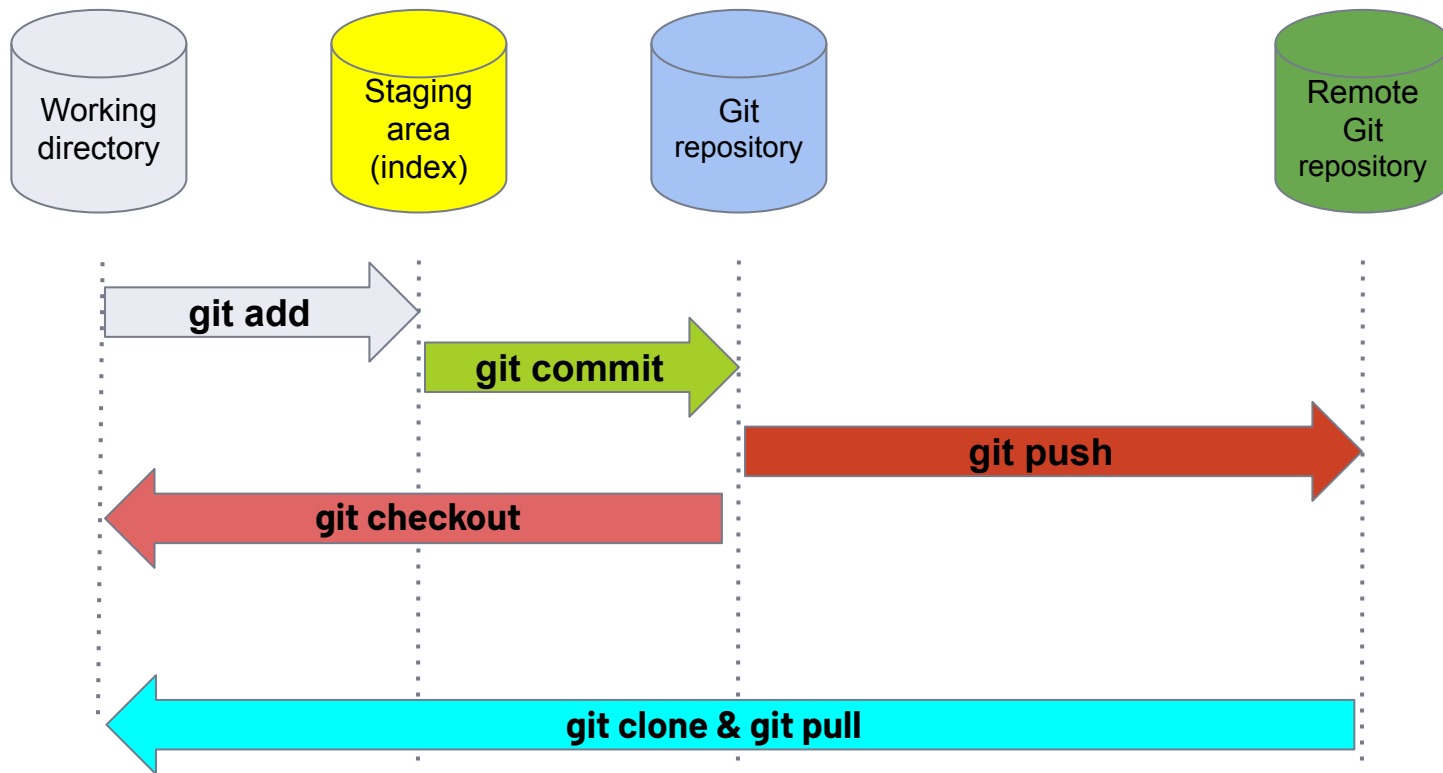
→ After first push continue with:

```
git push
```





Github - Remote Repository





Github - Remote Repository

111 contributions in the last year

Contribution settings ▾



Contribution activity

Jump to ▾

2018

September 2018

2017

How well did you like this lesson?



Students, drag the icon!



Pear Deck Interactive Slide
Do not remove this bar



THANKS!

Any questions?

You can find me at:

- ▶ @rafe
- ▶ rafe@clarusway.com



Did you finish pre-class work?



Students, drag the icon!



Pear Deck Interactive Slide
Do not remove this bar

Did you install Git?



Students, drag the icon!

REINVENT YOURSELF



Pear Deck Interactive Slide
Do not remove this bar

Did you create your Github account?



Students, drag the icon!



Pear Deck Interactive Slide
Do not remove this bar

Git Journey



- *Git introduction
- *Basic Git operation
- *Remote repository



- *Branches
- *Contribution to the remote repository



Hands-on



git



GitHub



5 Git Branches





Git Branches

- Branches
- Merges
- Merge Conflicts





Repo, Commit, Branch, Head

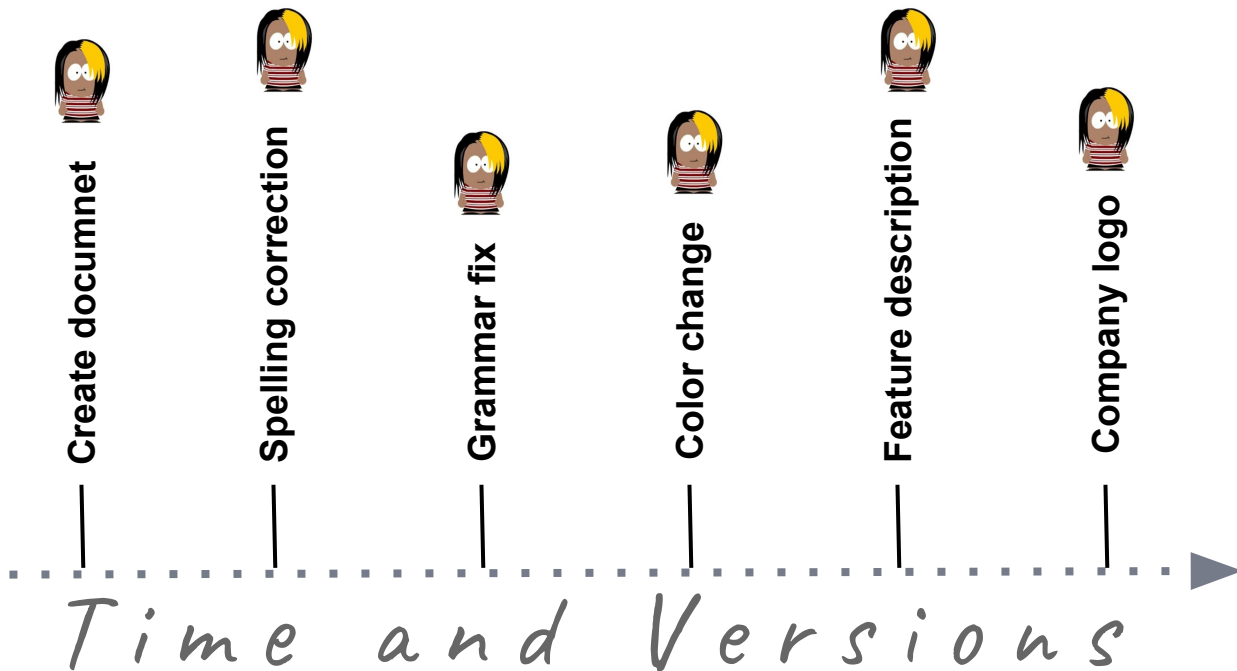
What comes to you your mind when you hear this?





Git Branches

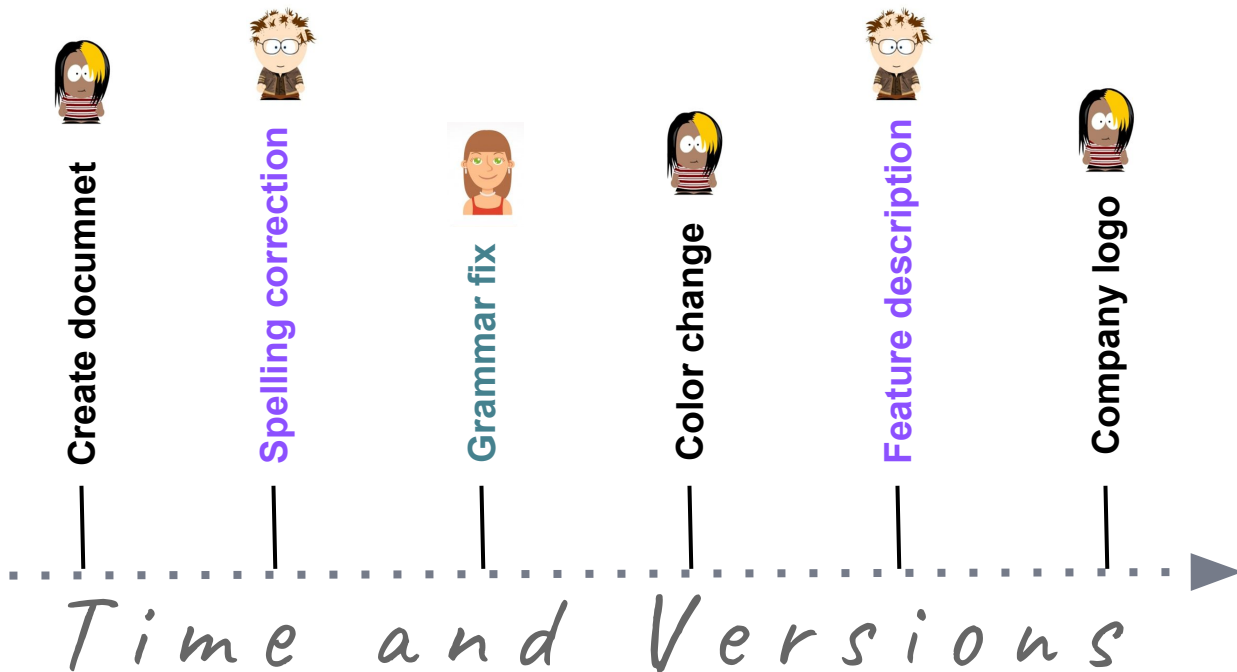
History Tracking





Git Branches

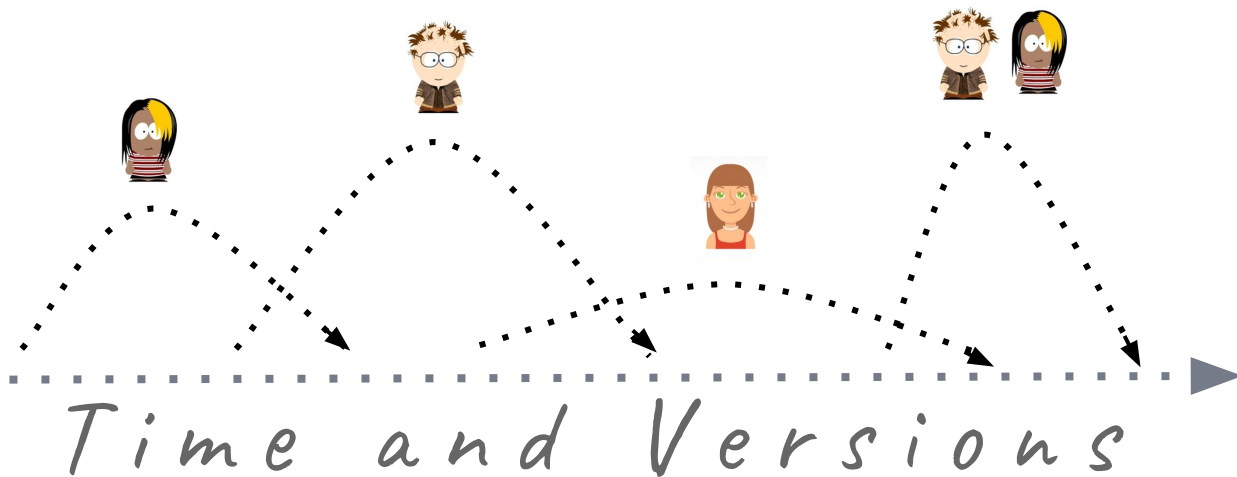
Collaborative History Tracking





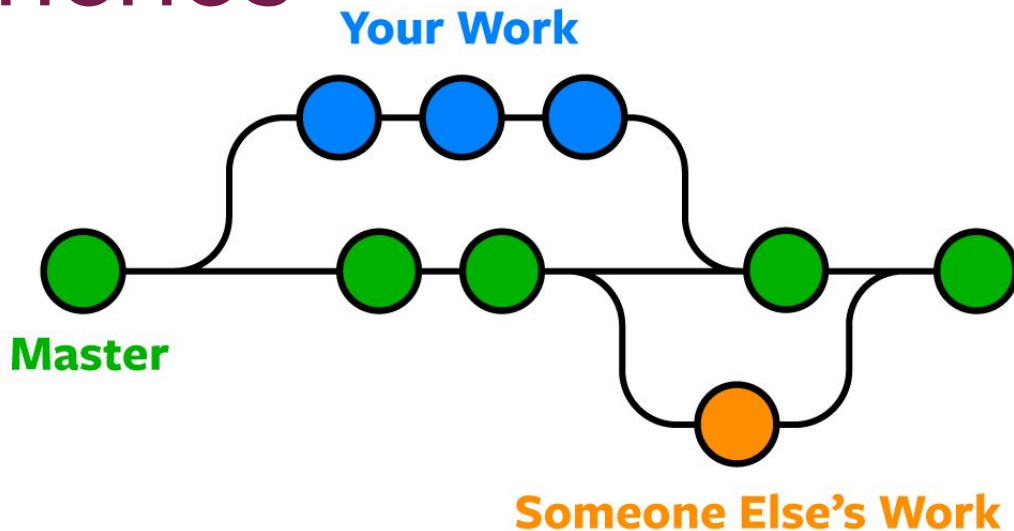
Git Branches

Collaborative History Tracking





Git Branches



- Production of the project lives on master/main branch
- Branches are reference to a commit

```
Eric's-Mac:project eric$ git branch  
* master
```



Git Branches

→ See local branches

```
git branch
```

→ See remote branches

```
git branch -r
```

→ See all branches

```
git branch -a
```



Creating/switching branches

- Create a new branch

```
git branch <branch name>
```

- Switch to a branch

```
git checkout <branch name>
```

- Create and switch to a branch

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



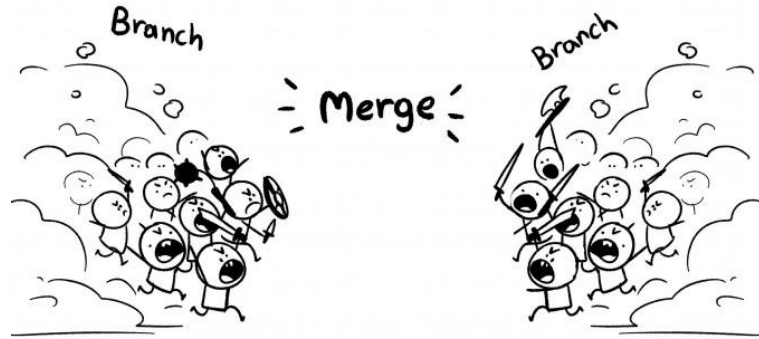
Deleting branches

→ Delete a local branch

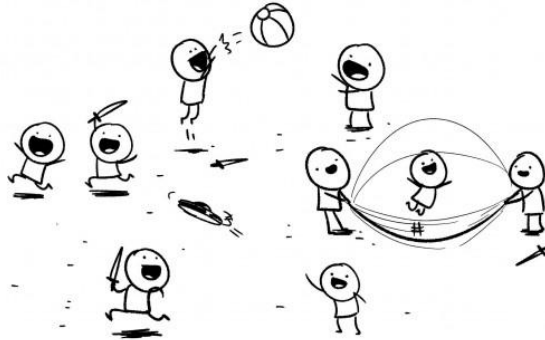
```
git branch -d <branch name>
```

```
git branch -D <branch name>
```


Merges



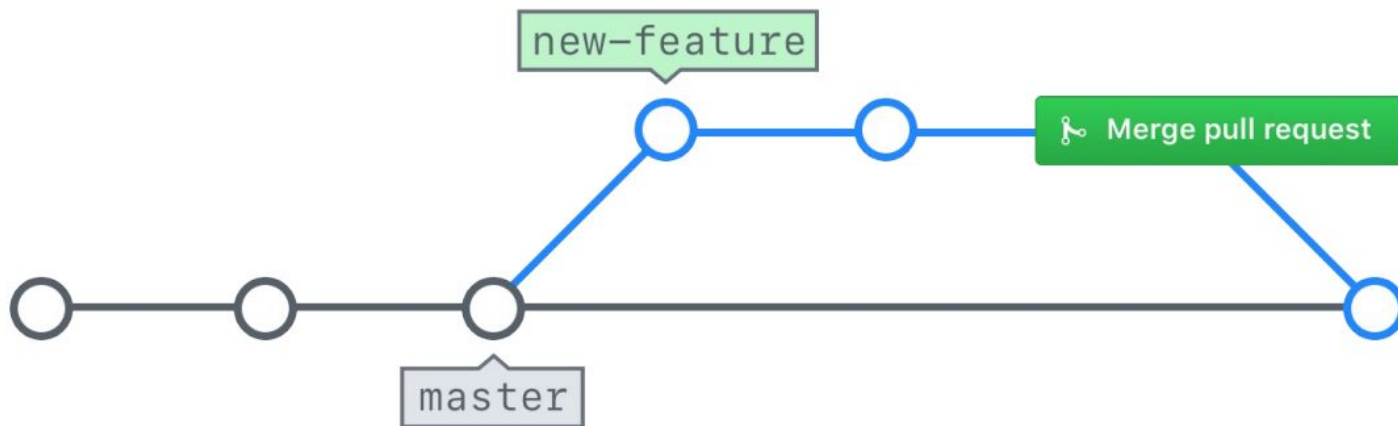
0 conflicts ♥
merge successful



 nasser_junior



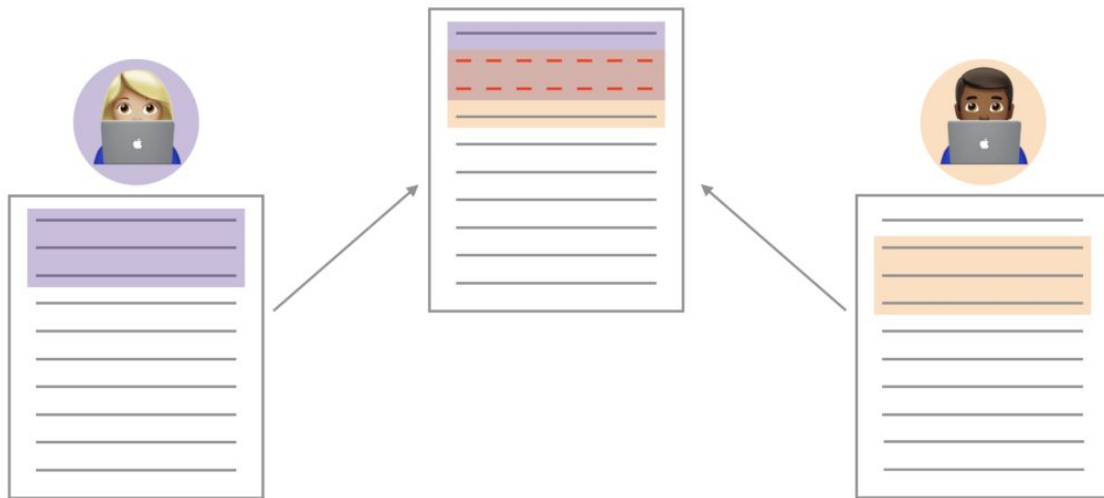
Merges





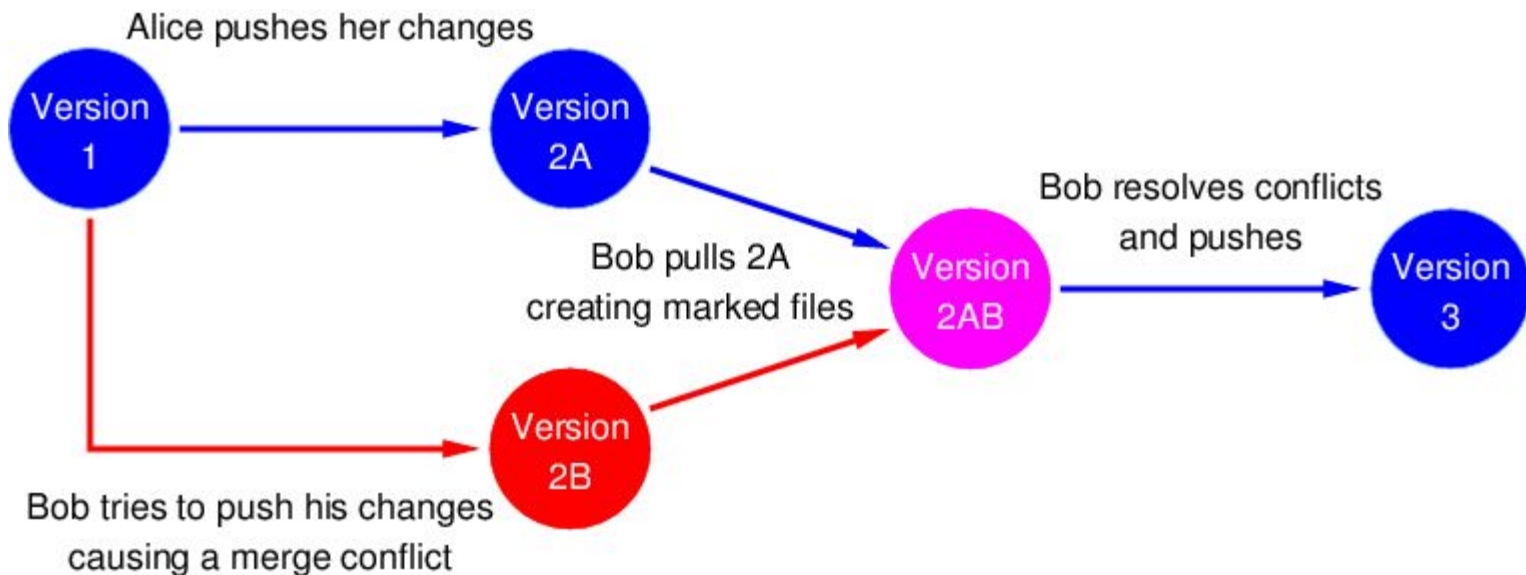
Merge Conflicts

- **Merge conflicts** happen when you merge branches that have competing commits, and Git needs your help to decide which changes to incorporate in the final merge.





Merge Conflicts





Kahoot!





6

Contribution to the Remote Repository



Objectives



- ▶ Pull Request
- ▶ Fork



Github - Pull Request

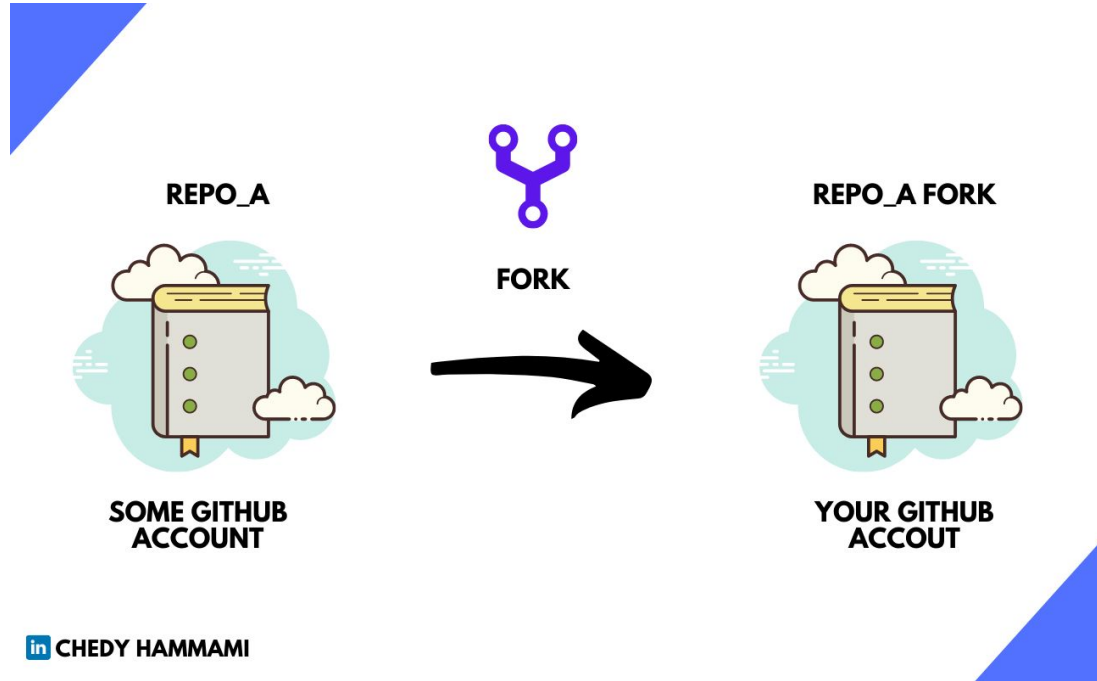
- Github's feature not Git's feature
- It allows you to contribute to other projects



Github - Pull Request

- **Pull Requests (PR)** let you tell others about changes you've pushed to a branch in a repository on GitHub
- You create a pull request to propose and collaborate on changes to a repository. These changes are proposed in a branch, which ensures that the master branch only contains finished and approved work.

Github - Fork



 CHEDY HAMMAMI

A fork is a copy of a repository.



In case of fire



1. git commit



2. git push



3. leave building

How well did you like this lesson?



Students, drag the icon!

REINVENT YOURSELF

Pear Deck Interactive Slide
Do not remove this bar



THANKS!

Any questions?

You can find me at:

- ▶ @rafe
- ▶ rafe@clarusway.com

