Git

CLARUSWAY©
WAY TO REINVENT YOURSELF

Git Journey

- Day1

Git introduction
Git workflow
Local repo operations

- Day2

Github repo
GUI
Branches
Merge & Conflicts

- Day3

Distribution models
Branching Strategies

- Day4

- Day5

CLARUSWAY©
WAY TO REINVENT YOURSELF

Git

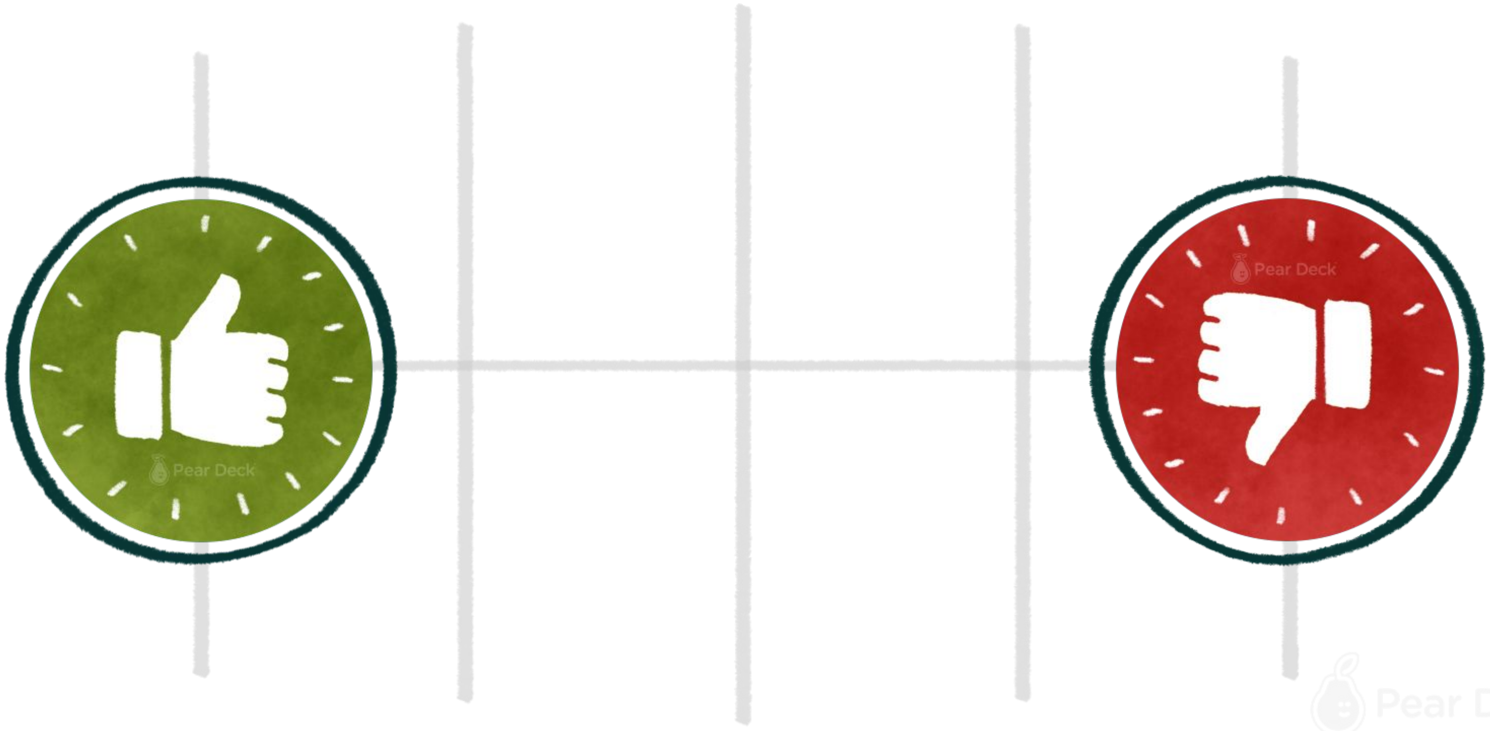
Version Control Systems

What comes to you your mind when you hear this?

Where are we about Git?

Let’s discuss about Git

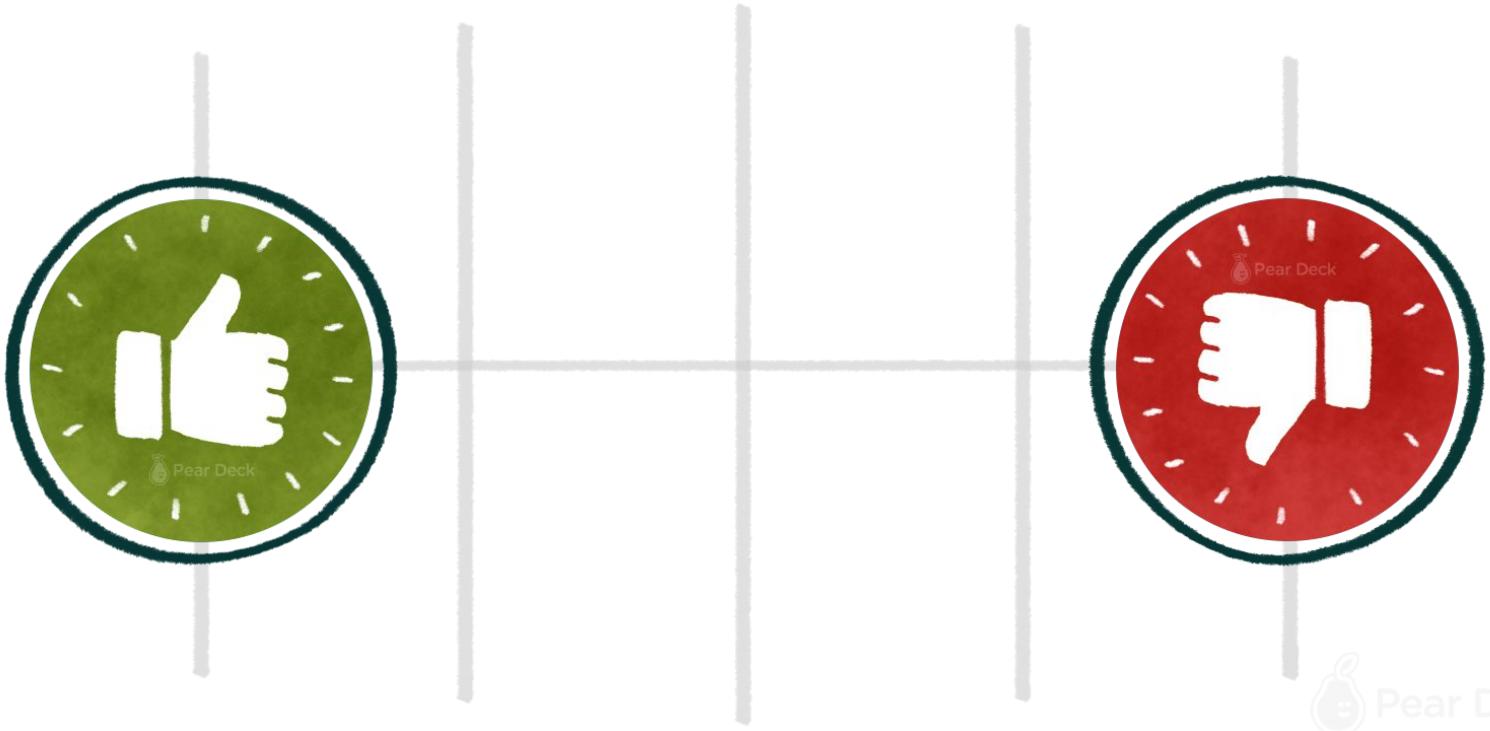
Did you find pre-class work?



Students, drag the icon!

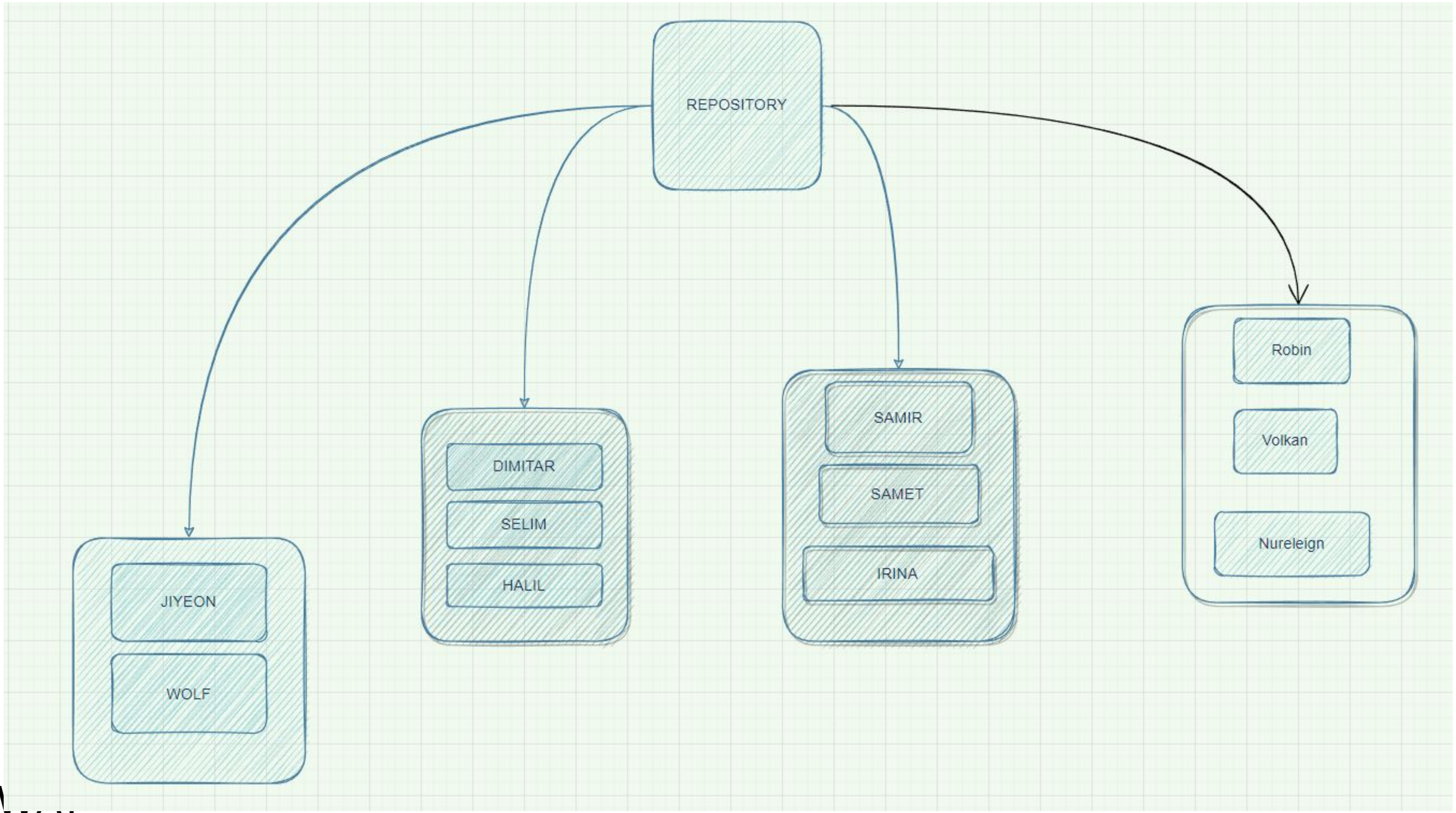
Pear Deck Interactive Slide
Do not remove this bar

Do you have Git installed on your computer?



Pear Deck

How can a team work?



Objectives

- Understand version control
- Familiarize Git terminology
- How to create a git repository
- Understand Git commands
- Understand Git workflow

What is a Version Control System?

Git

SVN

CVS

Mercurial

Perforce

Get Familiar with terms

Working tree

Index

Commit

Branch

Remote

Pull, Push, Log, Checkout, Switch

How to Create a Git Repository?

What is the difference between a regular folder and a git folder?

What does Git Init command actually do?

What is inside a .git folder?

CLARUSWAY©
WAY TO REINVENT YOURSELF

Git File States

Untracked

Modified

Staged

Committed

CLARUSWAY©
WAY TO REINVENT YOURSELF

Git Commands

What is Git CLI?

For which folder do the git commands work for?

Version Control Systems let us to;

- Track changes on files (text / source code files) for you
- Unlimited Undo / Redo
- Time Travel
- Collaborative development environment
- Compare and Share Responsibility
  - What changed
  - When it changed
  - Why it changed
  - Who changed it

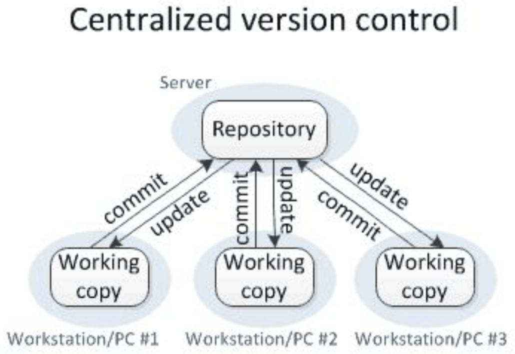
Why do we need Git?

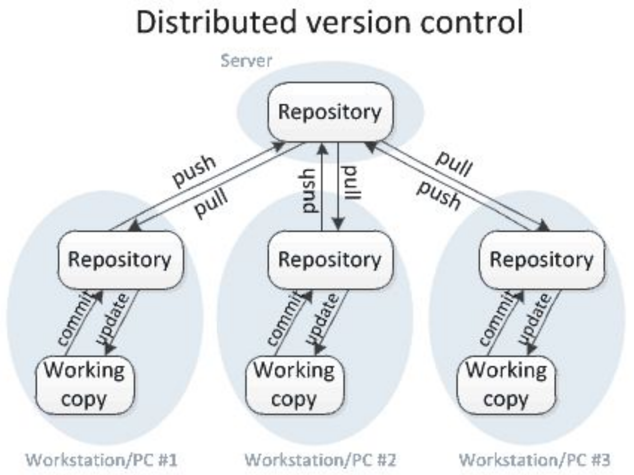
- Backup/Archive/Versioning/History
- Undo Changes
- Comparing
- Collaboration and Teamwork
- Code Review
- Sharing Responsibility

CLARUSWAY©
WAY TO REINVENT YOURSELF

Version Control Systems

- Types
  - Distributed
  - Centralized (Client-Server)



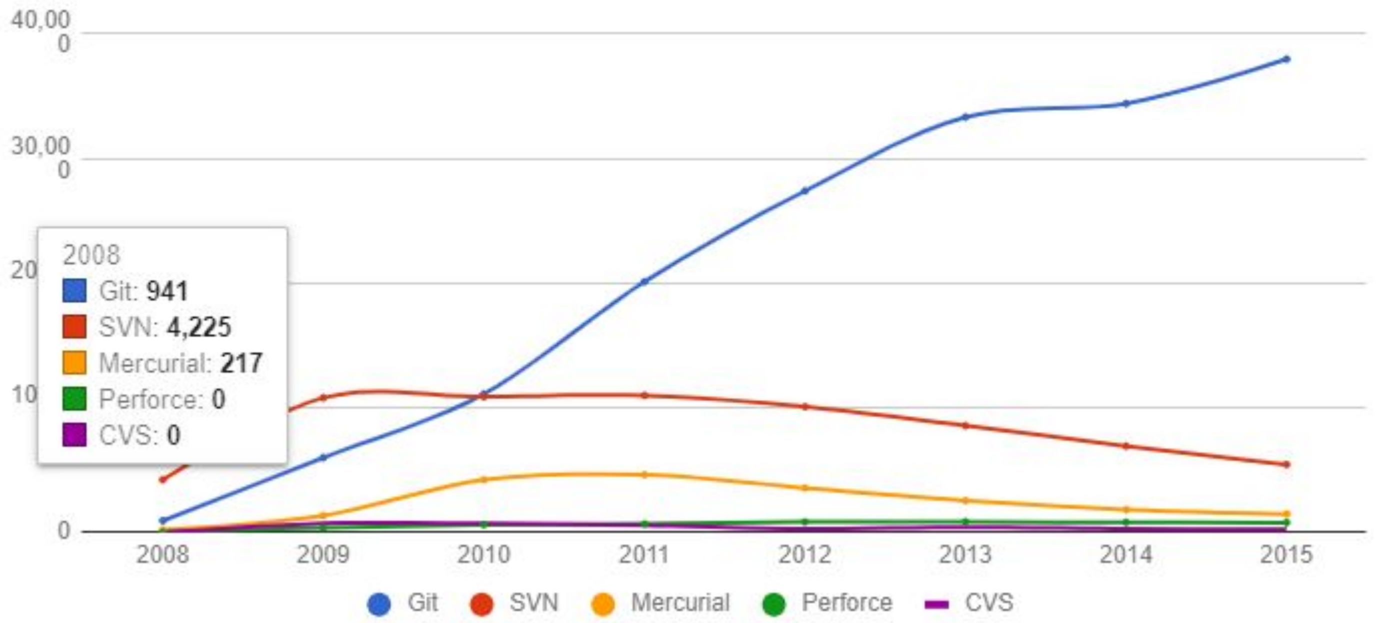


Git

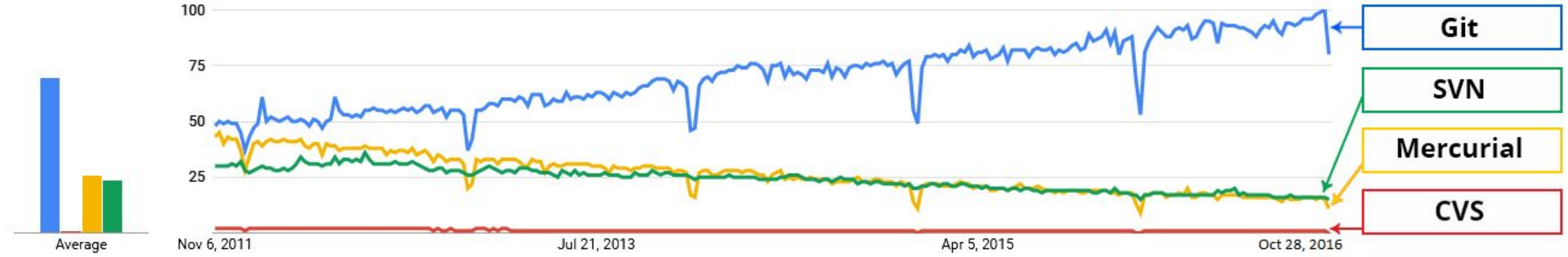
Git is an open source distributed version control system

CLARUSWAY©
WAY TO REINVENT YOURSELF

Popularity


40,000
30,000
20
10
0
2008
Git: 941
SVN: 4,225
Mercurial: 217
Perforce: 0
CVS: 0
2008
2009
2010
2011
2012
2013
2014
2015
Git
SVN
Mercurial
Perforce
CVS

Popularity



Source : https://www.edureka.co/blog/what-is-git/

Git Repository

The Workflow

CLARUSWAY©
WAY TO REINVENT YOURSELF

Workflow

Working Directory

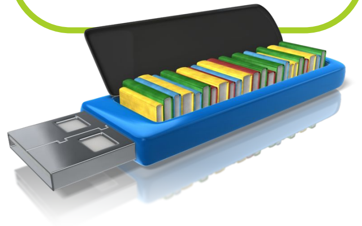
Where you work.Create new files, edit files delete files etc.

Staging Area (Index)

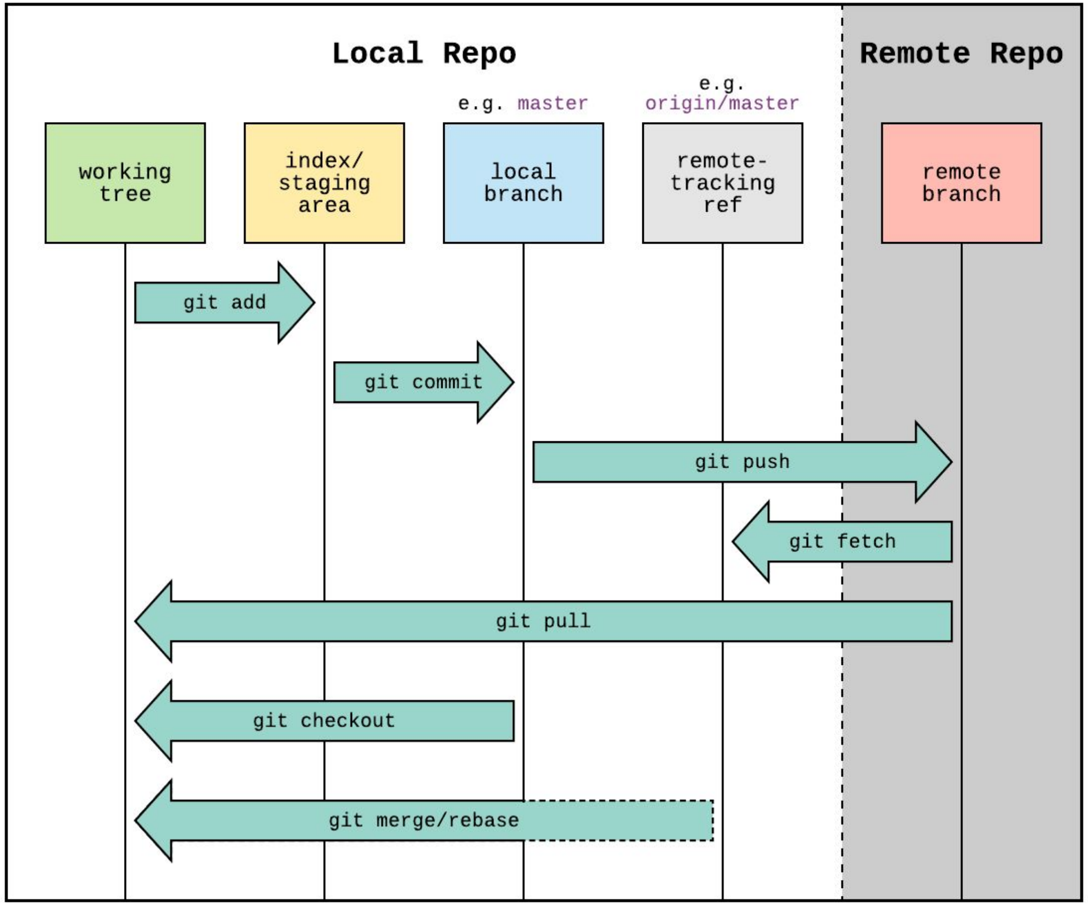
Before taking a snapshot, you're taking the files to a stage. Prepare the files to be committed.

Repository

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



Git Workflow



CLARUSWAY©
WAY TO REINVENT YOURSELF

File Stages

Committed	Unmodified changes from the last commit snapshot
Modified	Changes made to files since last commit snapshot
Staged	Changes marked to be added into the next commit snapshot

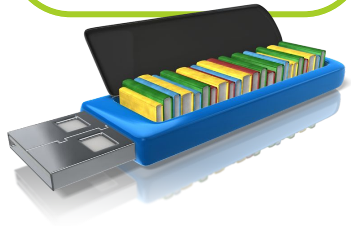
Create a new file

Working Directory

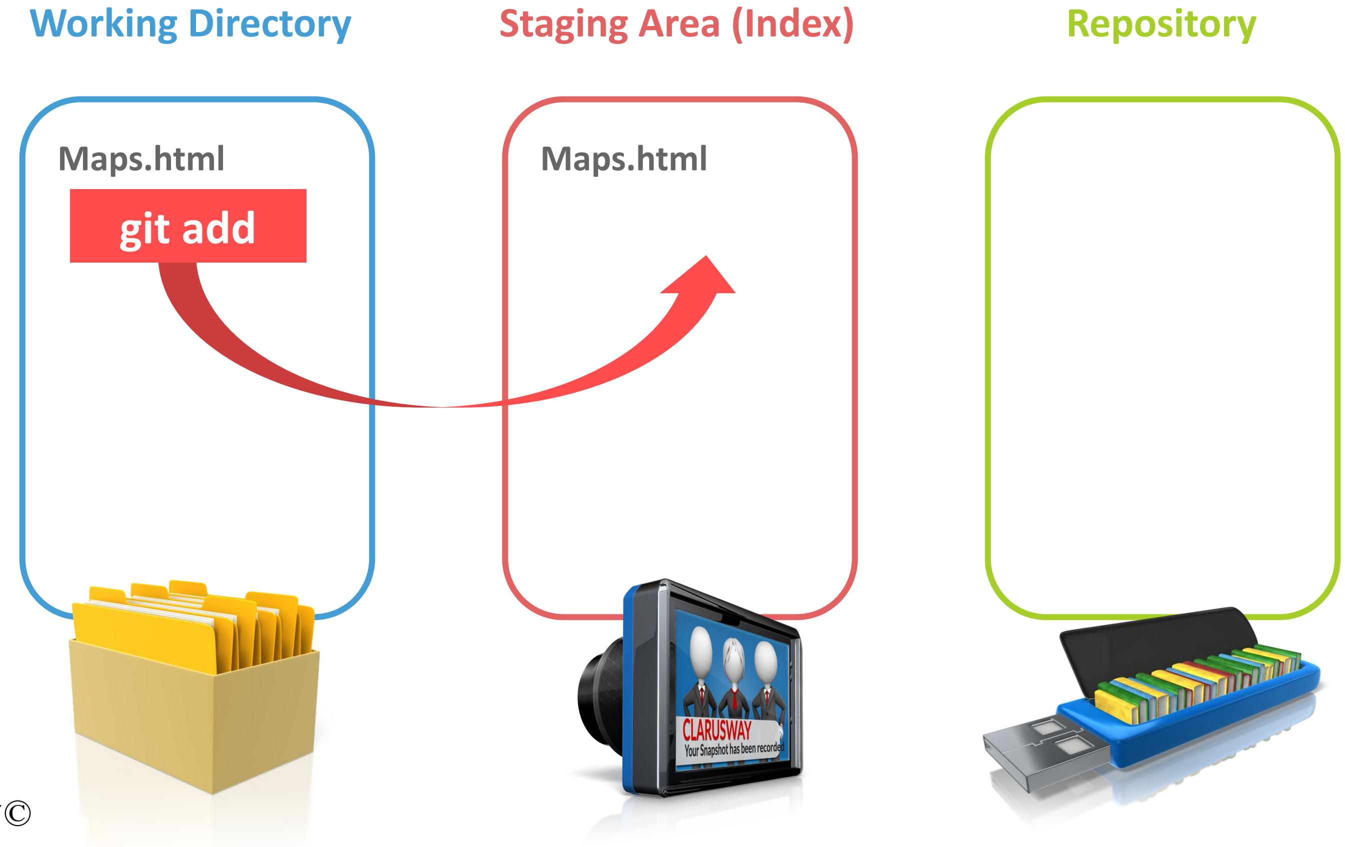
Maps.html
untracked file

Staging Area (Index)

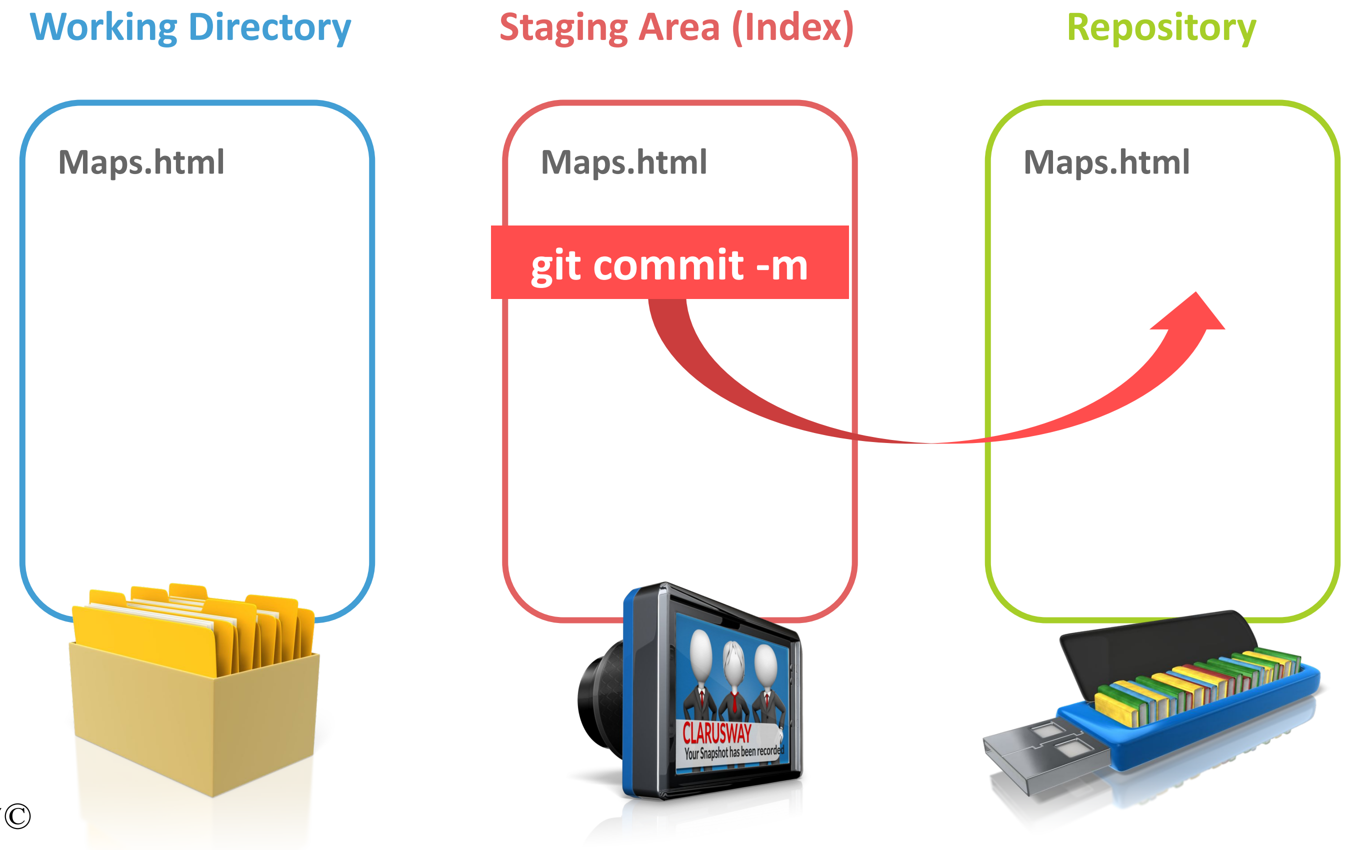
Repository



Track/stage a file

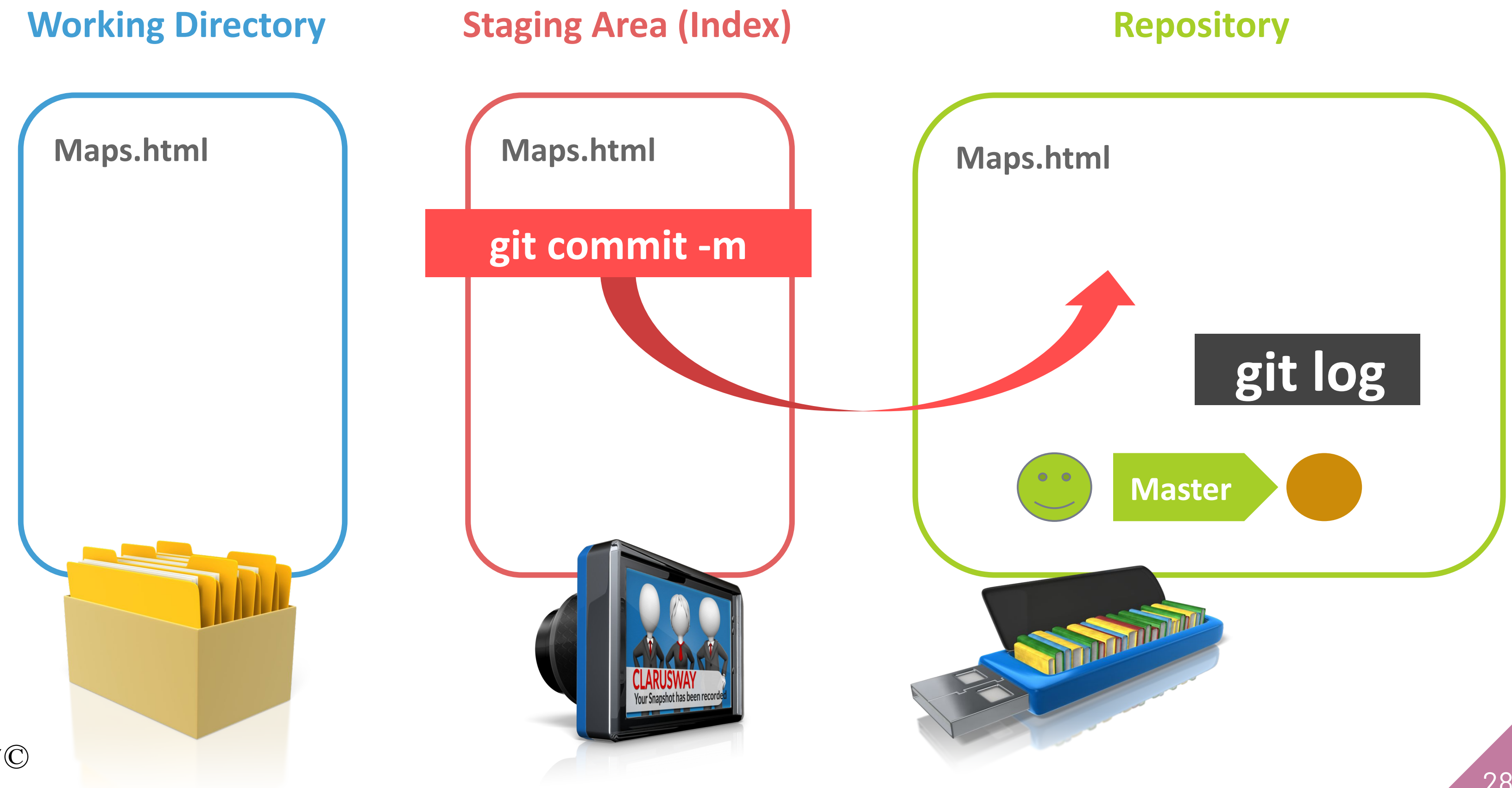


Commit

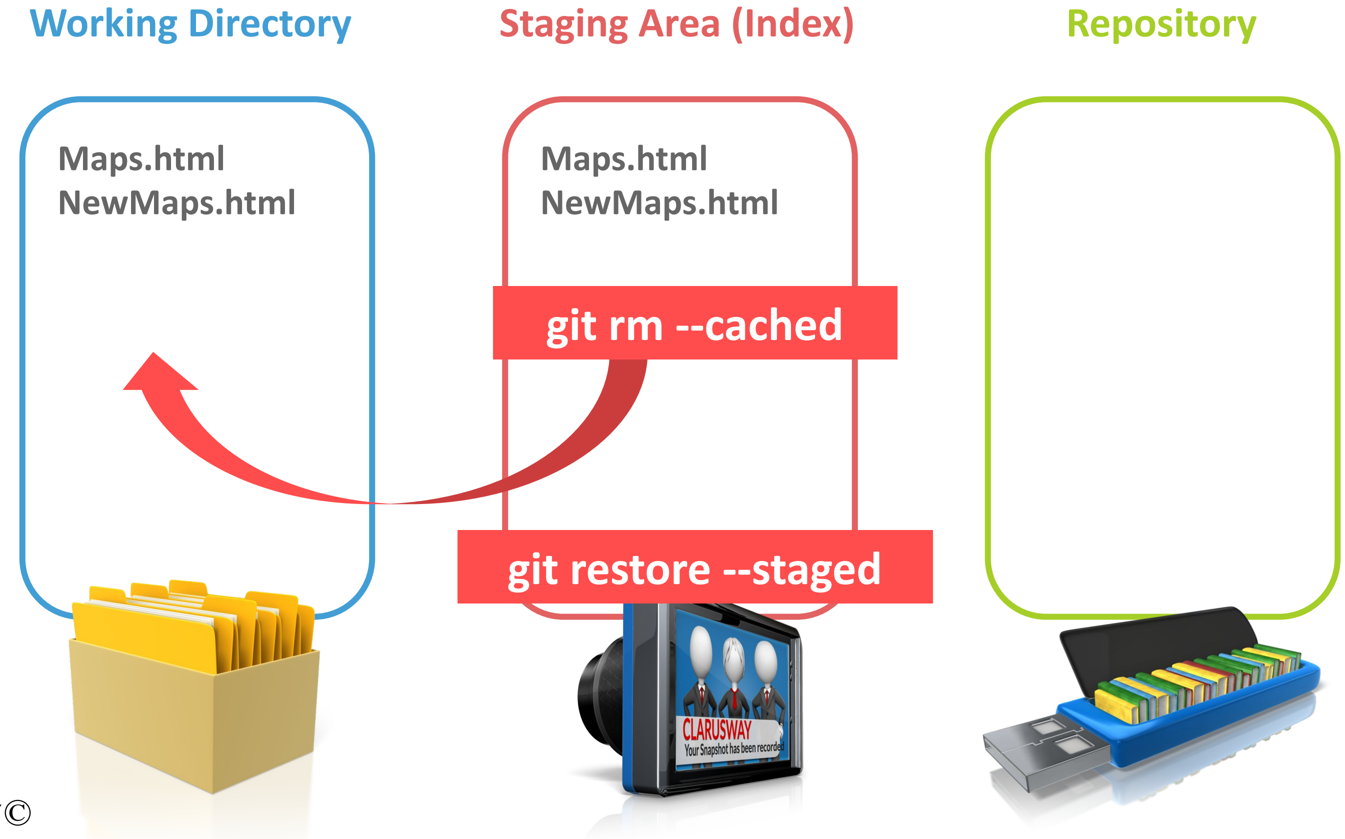


CLARUSWAY©
WAY TO REINVENT YOURSELF

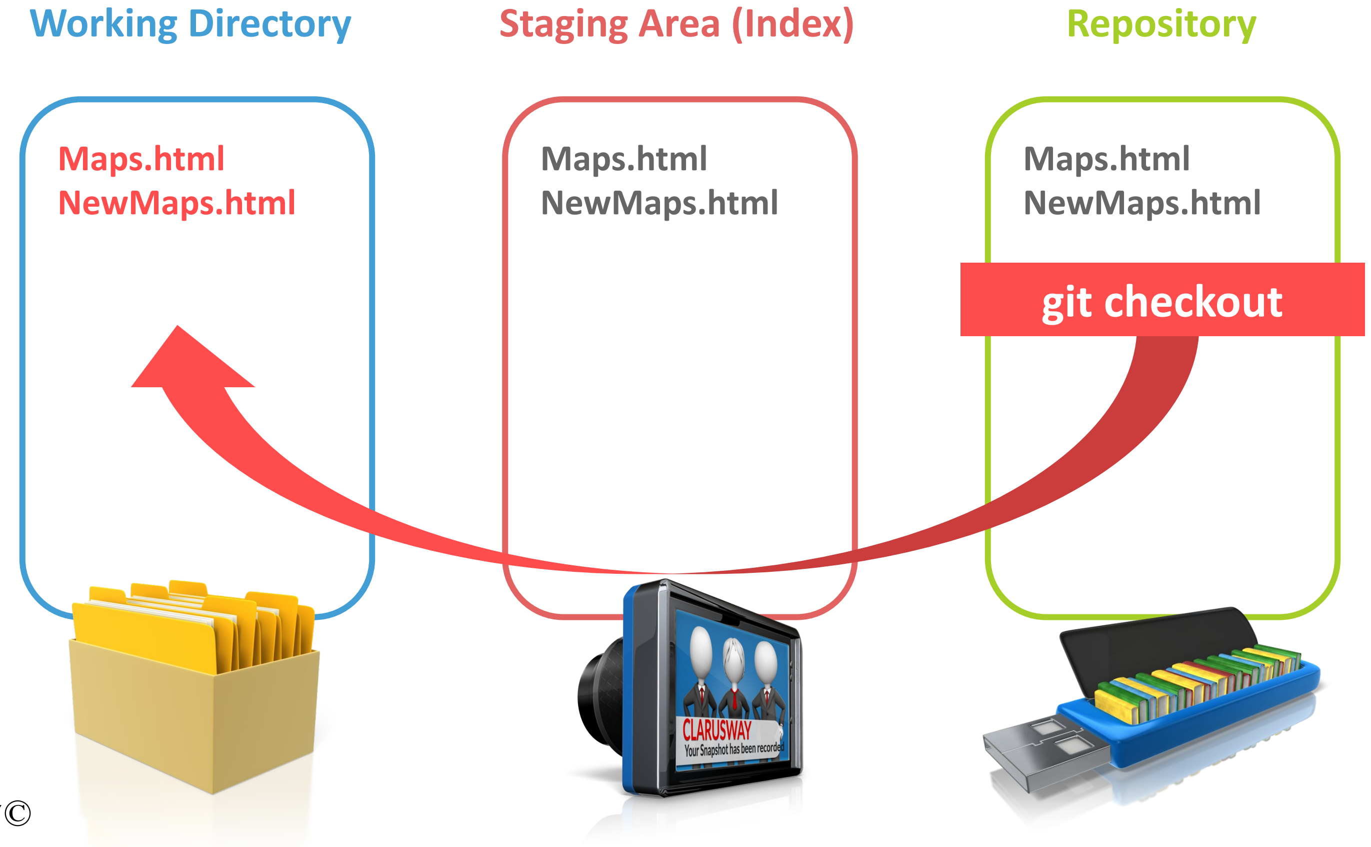
Commit


Working Directory
Staging Area (Index)
Repository
Maps.html
Maps.html
Maps.html
git commit -m
git log
Master
CLARUSWAY
Your Snapshot has been recorded

Remove from stage



Checkout from Repo



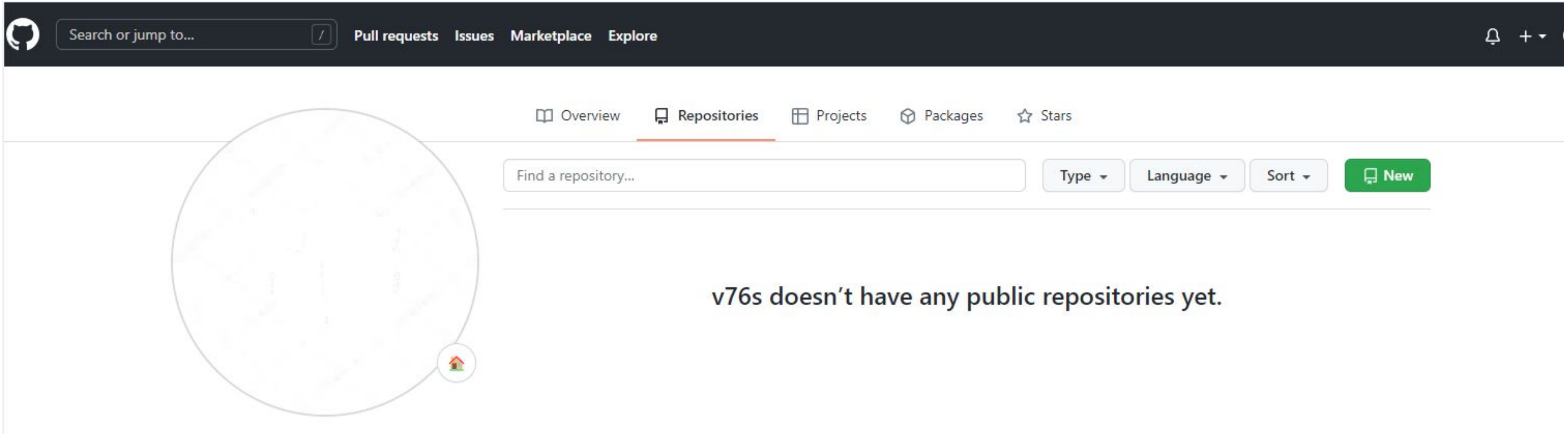
CLARUSWAY©
WAY TO REINVENT YOURSELF

Git User interface tools

- GitHub Desktop
- GitKraken
- Sourcetree
- Tortoise Git
- SmartGit
- GitForce
- Git Cola
- Aurees
- Magit
- Fork

Since we have the UI tools, why do we need to know CLI (Gitbash) commands?

GitHub



v76s doesn't have any public repositories yet.

Theoretical Investigation

```
git –version
git init
git status
git add
git remote add
git commit
git push
git pull
git merge
git log
git branch
git checkout
```

CLARUSWAY©
WAY TO REINVENT YOURSELF

Git Repository

- Let’s check if you have git in your computer

```
git --version
```

- git needs your identity to mark/label changes / editor

```
git config --global user.name “Your Name”
```

```
git config --global user.email “Your Name”
```

```
git config --global core.editor “vim”
```

```
git config --list
```

CLARUSWAY©
WAY TO REINVENT YOURSELF

Track a new file

- let’s create a new file in our project folder

```
touch Maps.html
```

- let’s edit this file

```
vim Maps.html
```

- let’s check the status of our project

```
git status
```

Git Repository

- to create a new local repo

```
git init
```

- to see the commands

```
git
```

- to see the status of your repo

```
git status
```

CLARUSWAY©
WAY TO REINVENT YOURSELF

Git Repository

→ to create a new remote repo and connect it with your local repo (after you create a remote repo on Github/Bitbucket etc.)

```
git clone address
```

Stage files options

→ stage one file

```
git add filename
```

→ stage all files (new, modified)

```
git add .
```

→ stage all changes

```
git add -A
```

→ stage modified and deleted files only

```
git add -u
```

Commit

→ Commit the files on the stage

```
git commit -m ”message”
```

→ Add and commit all tracked files

```
git commit -am ”message”
```

→ amend commit message

```
git commit --amend
```

Recipe for a new project

→ Create a repo	`git init`
→ Create a new file/edit file etc.	
→ Stage/Track your changes	`git add .`
→ Commit changes	`git commit -m "message"`

Time to practice

```
git –version
```

Time to practice

```
git status
git clone https://github.com/v76s/GoogleMaps.git
git add .
git commit -m "Message"
git remote add origin https://github.com/v76s/GoogleMaps.git
git push
git log (--all --decorate --oneline --graph)
git branch
git checkout  BRANCH_NAME
git switch BRANCH_NAME
```

Kahoot!

Objectives

- Review the prior command set
- Familiarize Github features and UI
- Git Desktop install and use
- GitKraken UI introduction

CLARUSWAY©
WAY TO REINVENT YOURSELF

Review Commands 1

```
git version
git init
git status
git add .
git remote add (Alias) https://github.com/(Account)/(Repo).git
git commit
git push (Alias) (Repository)
git push
```

Review Commands 2

update the Maps.html contents with a comment

and execute git pull to receive the updates.

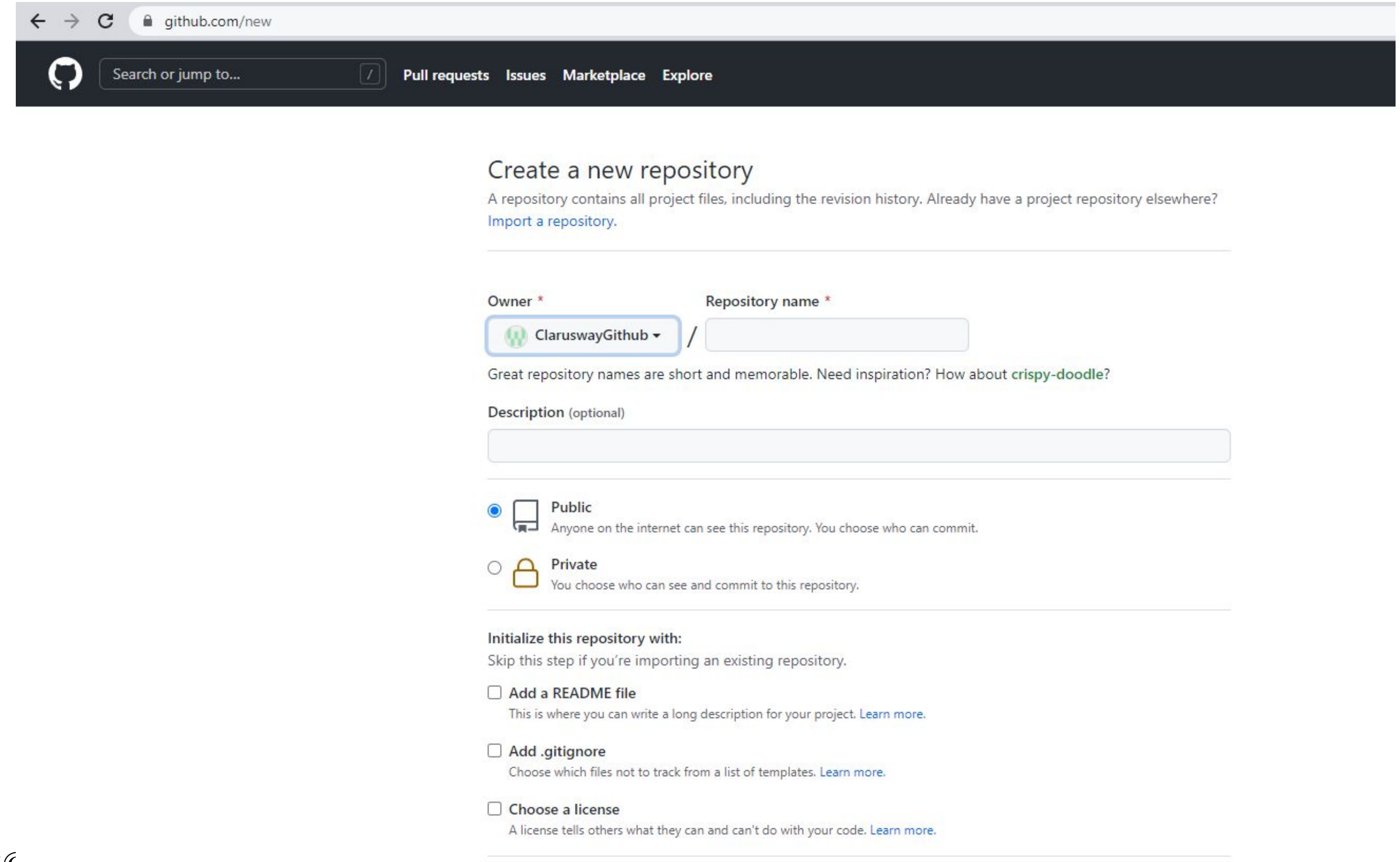
CLARUSWAY©
WAY TO REINVENT YOURSELF

Review Commands 3

Perform several times the enlisted commands :

Create a repository in Github.

GitHub interface



github.com/new

Search or jump to... / Pull requests Issues Marketplace Explore

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * Repository name *

ClaruswayGithub /

Great repository names are short and memorable. Need inspiration? How about crispy-doodle?

Description (optional)

(●) Public
Anyone on the internet can see this repository. You choose who can commit.

( ) Private
You choose who can see and commit to this repository.

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

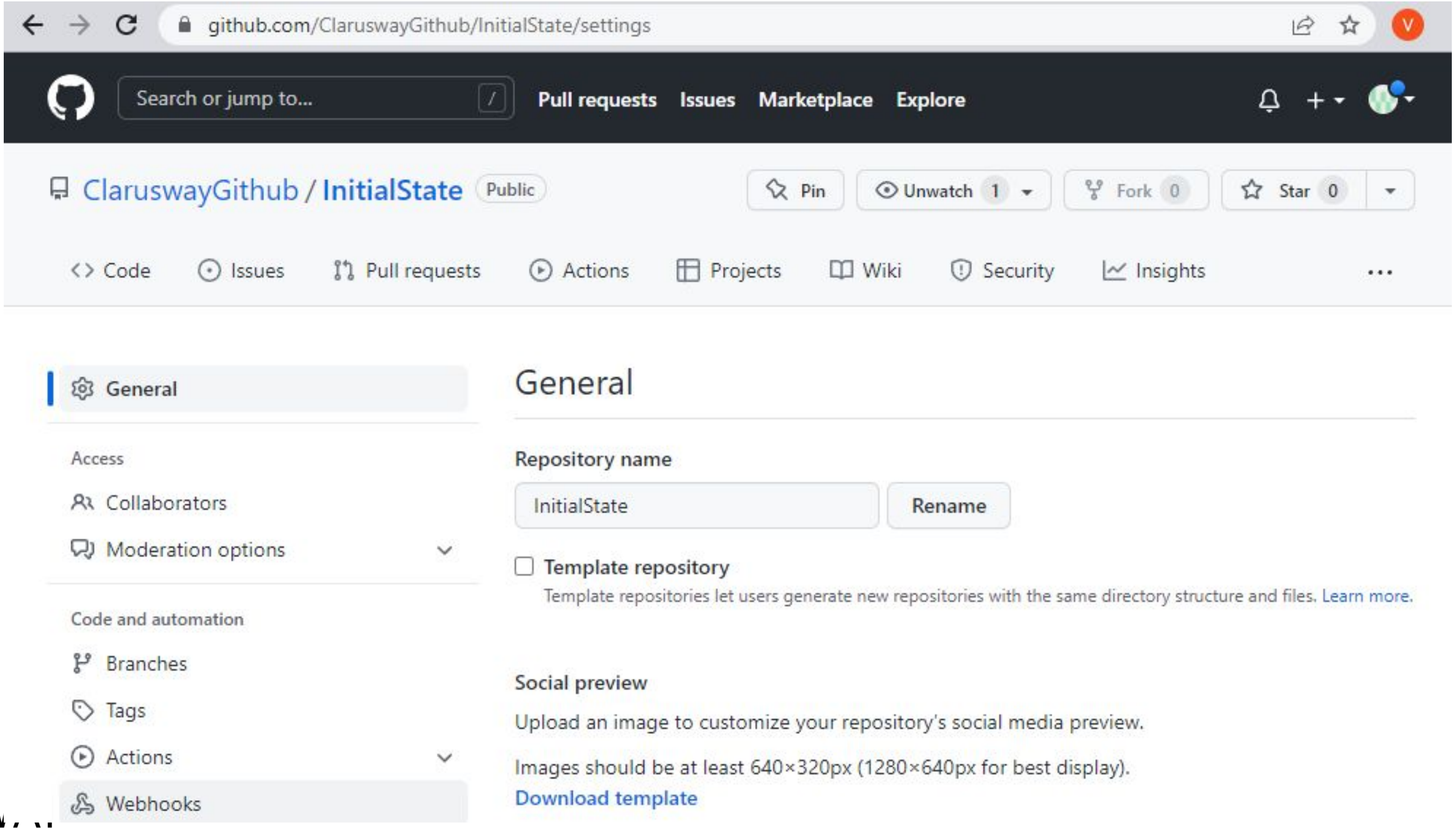
☐ Add a README file
This is where you can write a long description for your project. Learn more.

☐ Add .gitignore
Choose which files not to track from a list of templates. Learn more.

☐ Choose a license
A license tells others what they can and can't do with your code. Learn more.

CLARUSWAY
WAY TO REINVENT YOURSELF

GitHub interface



General

Repository name

InitialState

Rename

Template repository

Template repositories let users generate new repositories with the same directory structure and files. Learn more.

Social preview

Upload an image to customize your repository's social media preview.

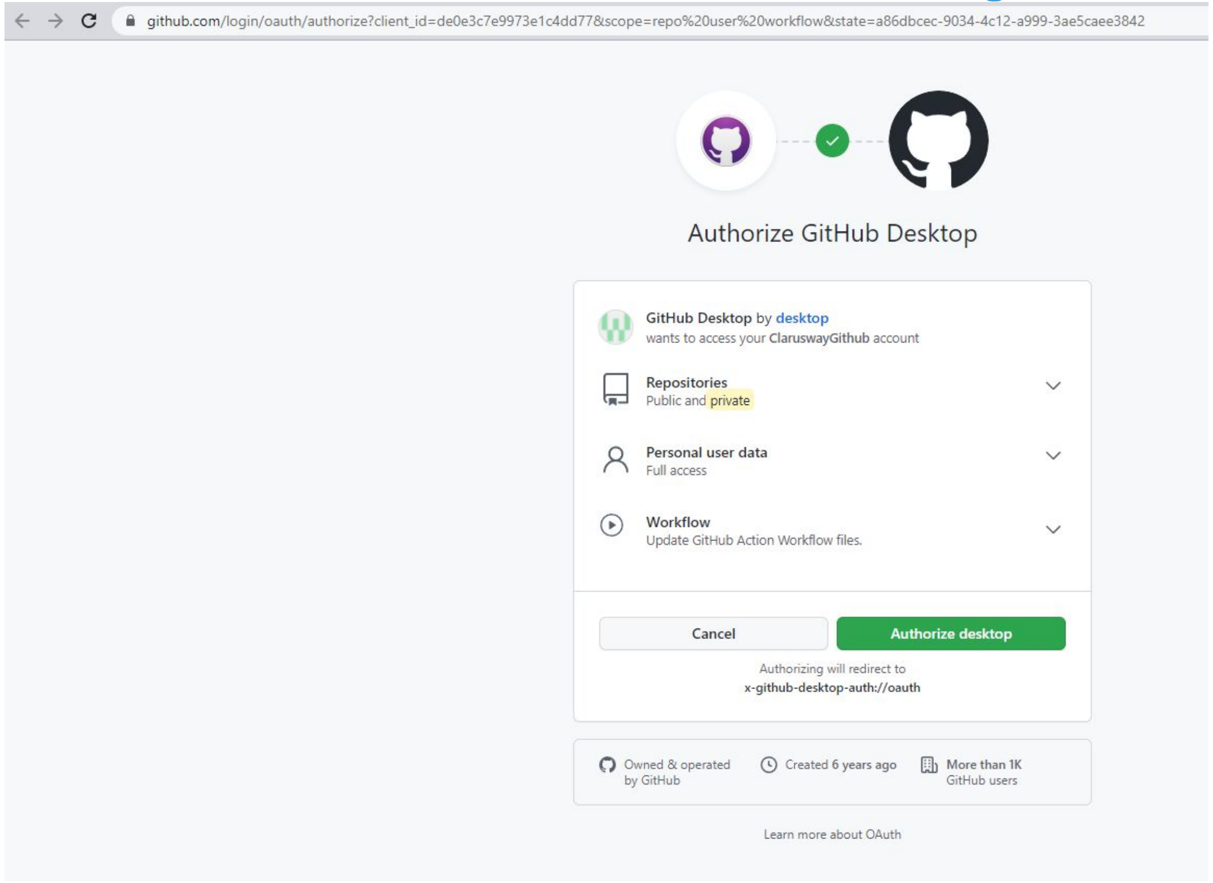
Images should be at least 640×320px (1280×640px for best display).

Download template

GitHub Desktop

https://desktop.github.com/

GitHub Desktop



github.com/login/oauth/authorize?client_id=de0e3c7e9973e1c4dd77&scope=repo%20user%20workflow&state=a86dbcec-9034-4c12-a999-3ae5caee3842

Authorize GitHub Desktop

GitHub Desktop by desktop
wants to access your ClaruswayGithub account

Repositories
Public and private

Personal user data
Full access

Workflow
Update GitHub Action Workflow files.

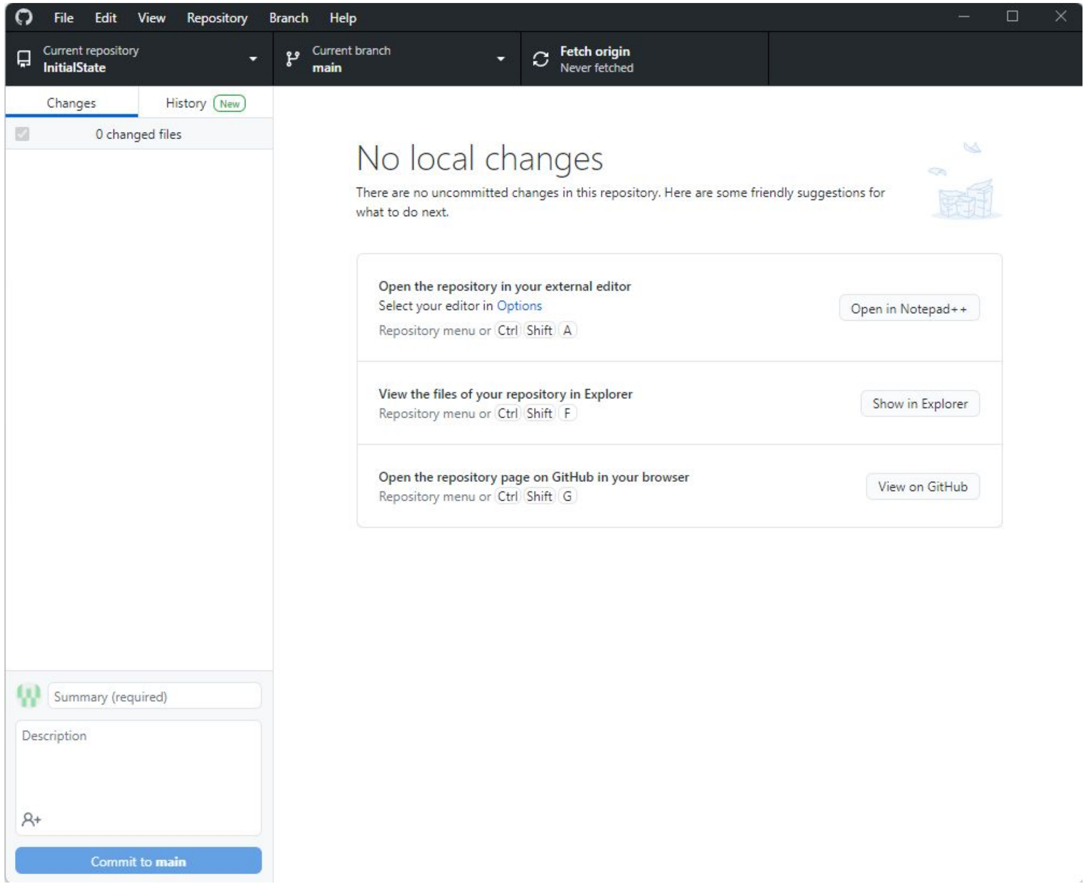
Authorizing will redirect to
x-github-desktop-auth://oauth

Owned & operated by GitHub
Created 6 years ago
More than 1K GitHub users

Learn more about OAuth

CLARUSWAY©
WAY TO REINVENT YOURSELF

GitHub Desktop



No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Open the repository in your external editor
Select your editor in Options
Repository menu or Ctrl Shift A
Open in Notepad++

View the files of your repository in Explorer
Repository menu or Ctrl Shift F
Show in Explorer

Open the repository page on GitHub in your browser
Repository menu or Ctrl Shift G
View on GitHub

CLARUSWAY©
WAY TO REINVENT YOURSELF

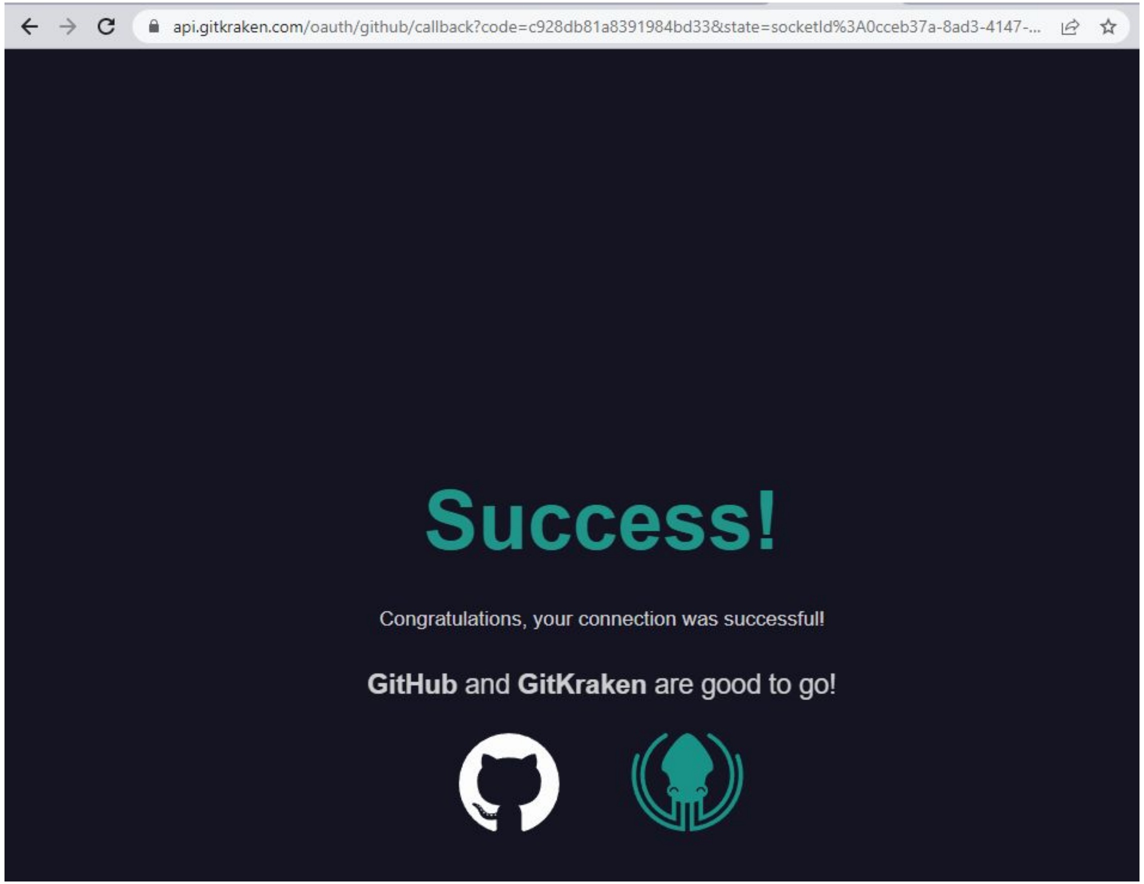
GitKraken Overview

https://www.gitkraken.com/download

GitKraken Overview

https://www.gitkraken.com/download

GitKraken Overview



api.gitkraken.com/oauth/github/callback?code=c928db81a8391984bd33&state=socketId%3A0cceb37a-8ad3-4147-...

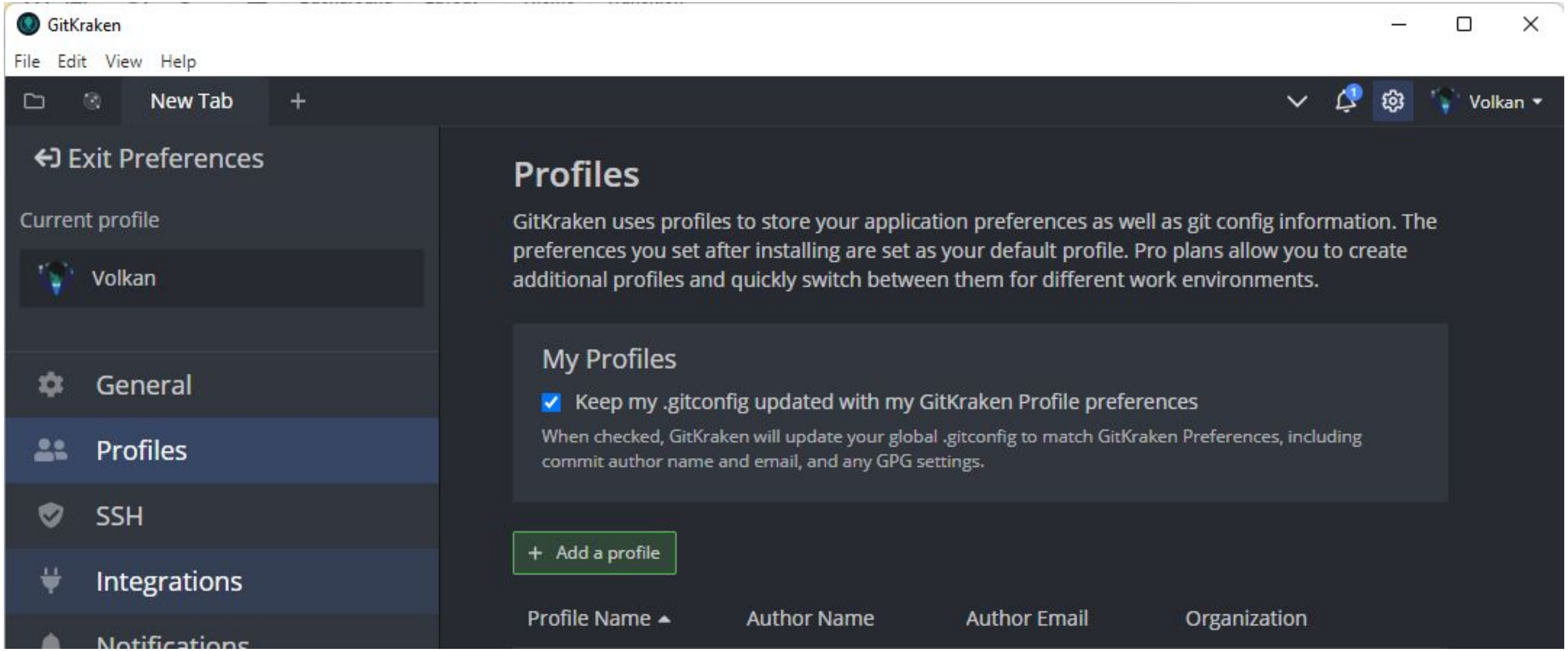
Success!

Congratulations, your connection was successful!

GitHub and **GitKraken** are good to go!

CLARUSWAY©
WAY TO REINVENT YOURSELF

GitKraken Overview



Profiles

GitKraken uses profiles to store your application preferences as well as git config information. The preferences you set after installing are set as your default profile. Pro plans allow you to create additional profiles and quickly switch between them for different work environments.

My Profiles

☑ Keep my .gitconfig updated with my GitKraken Profile preferences

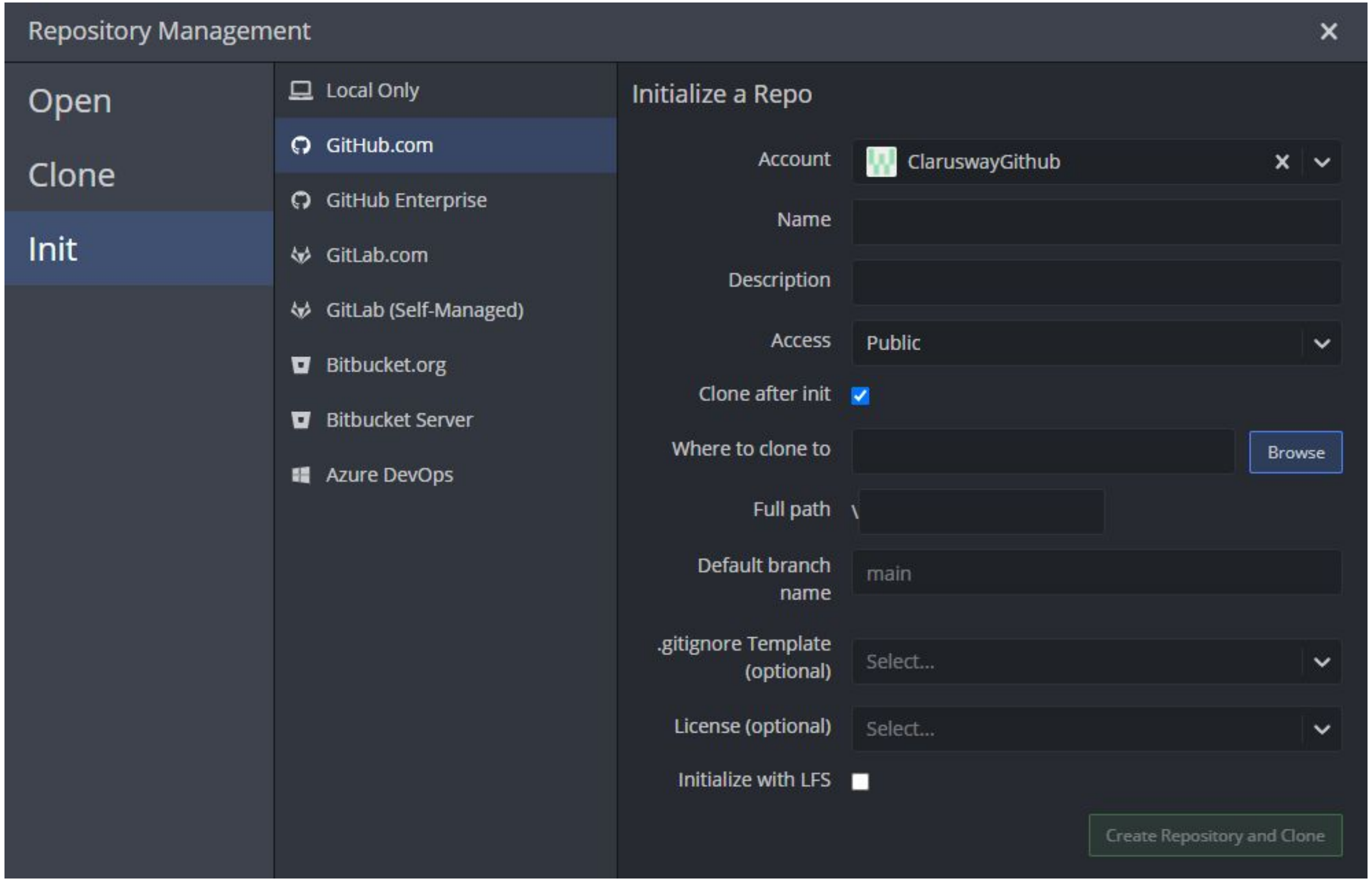
When checked, GitKraken will update your global .gitconfig to match GitKraken Preferences, including commit author name and email, and any GPG settings.

+ Add a profile

Profile Name ▲	Author Name	Author Email	Organization

CLARUSWAY©
WAY TO REINVENT YOURSELF

GitKraken Overview

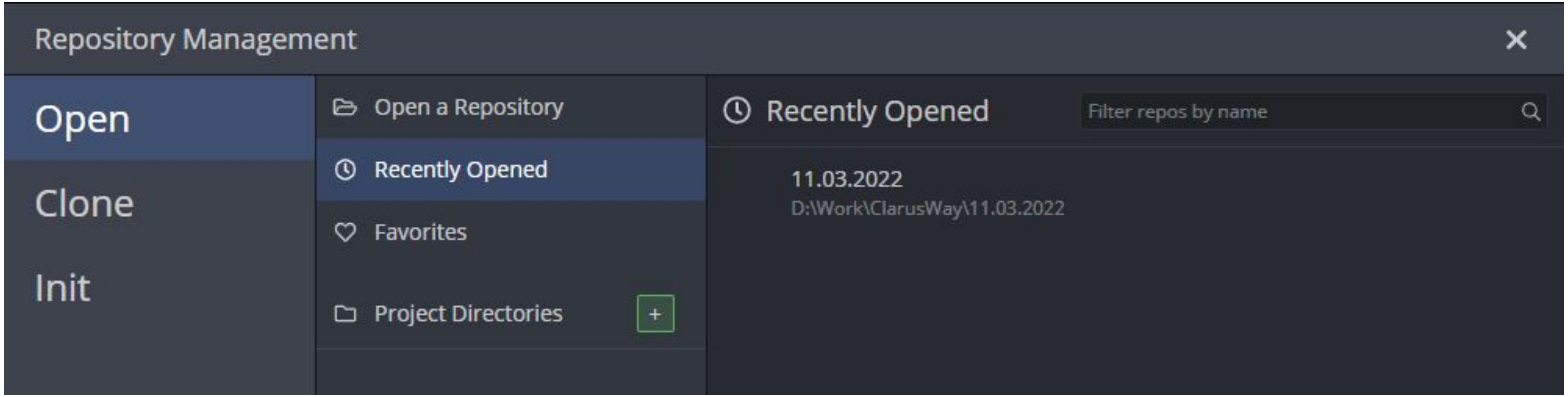


Repository Management

Initialize a Repo

Account	ClaruswayGithub
Name	
Description	
Access	Public
Clone after init	✓
Where to clone to	
Full path	\
Default branch name	main
.gitignore Template (optional)	Select...
License (optional)	Select...
Initialize with LFS	☐

GitKraken Overview


Repository Management

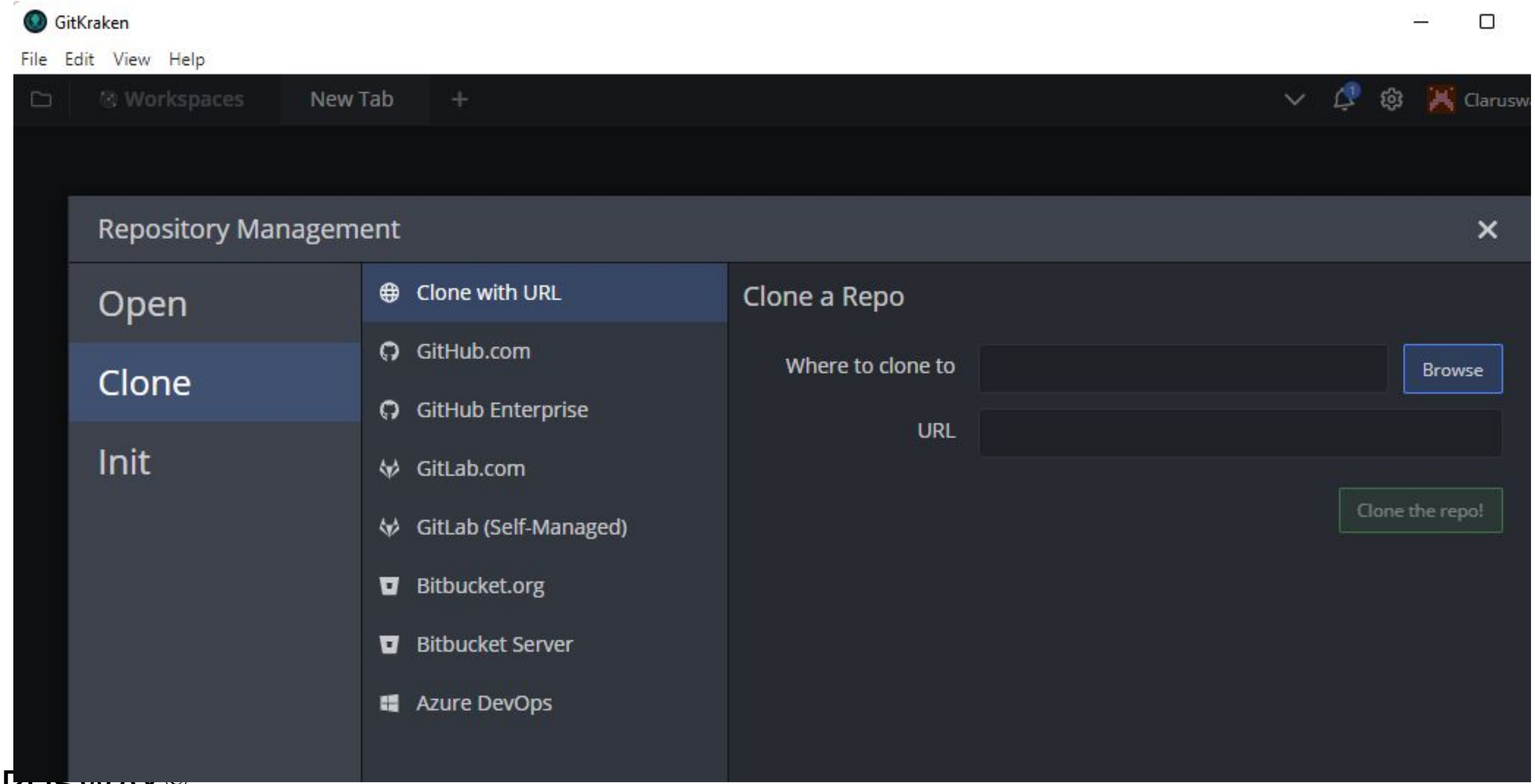
Open
Clone
Init

Open a Repository
Recently Opened
Favorites
Project Directories

Recently Opened
Filter repos by name

11.03.2022
D:\Work\ClarusWay\11.03.2022

GitKraken Overview



GitKraken

File Edit View Help

Workspaces New Tab +

Clarusw

Repository Management

Open

Clone

Init

Clone with URL

GitHub.com

GitHub Enterprise

GitLab.com

GitLab (Self-Managed)

Bitbucket.org

Bitbucket Server

Azure DevOps

Clone a Repo

Where to clone to

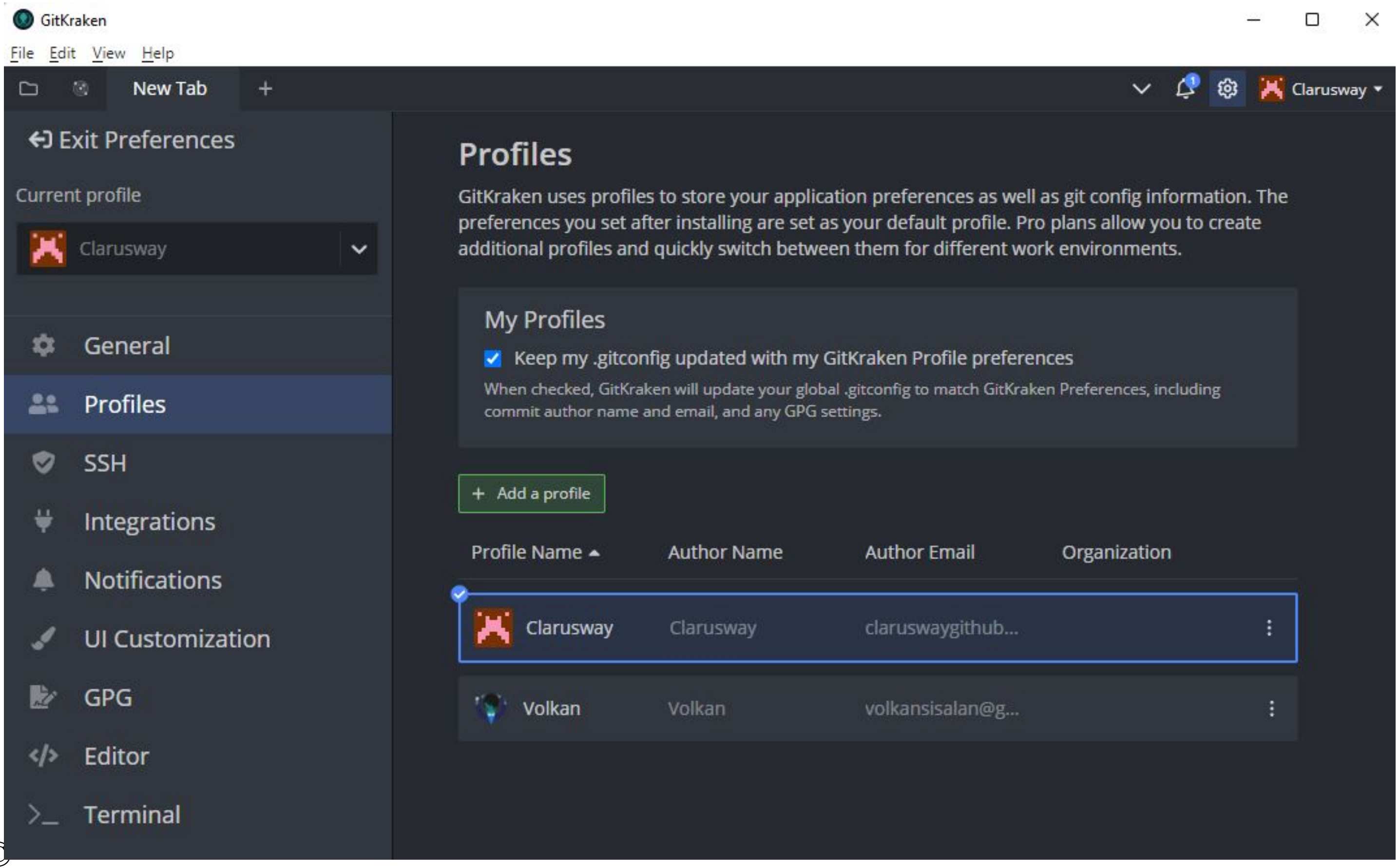
Browse

URL

Clone the repo!

CLARUSWAY©
WAY TO REINVENT YOURSELF

GitKraken Overview



Profiles

GitKraken uses profiles to store your application preferences as well as git config information. The preferences you set after installing are set as your default profile. Pro plans allow you to create additional profiles and quickly switch between them for different work environments.

My Profiles

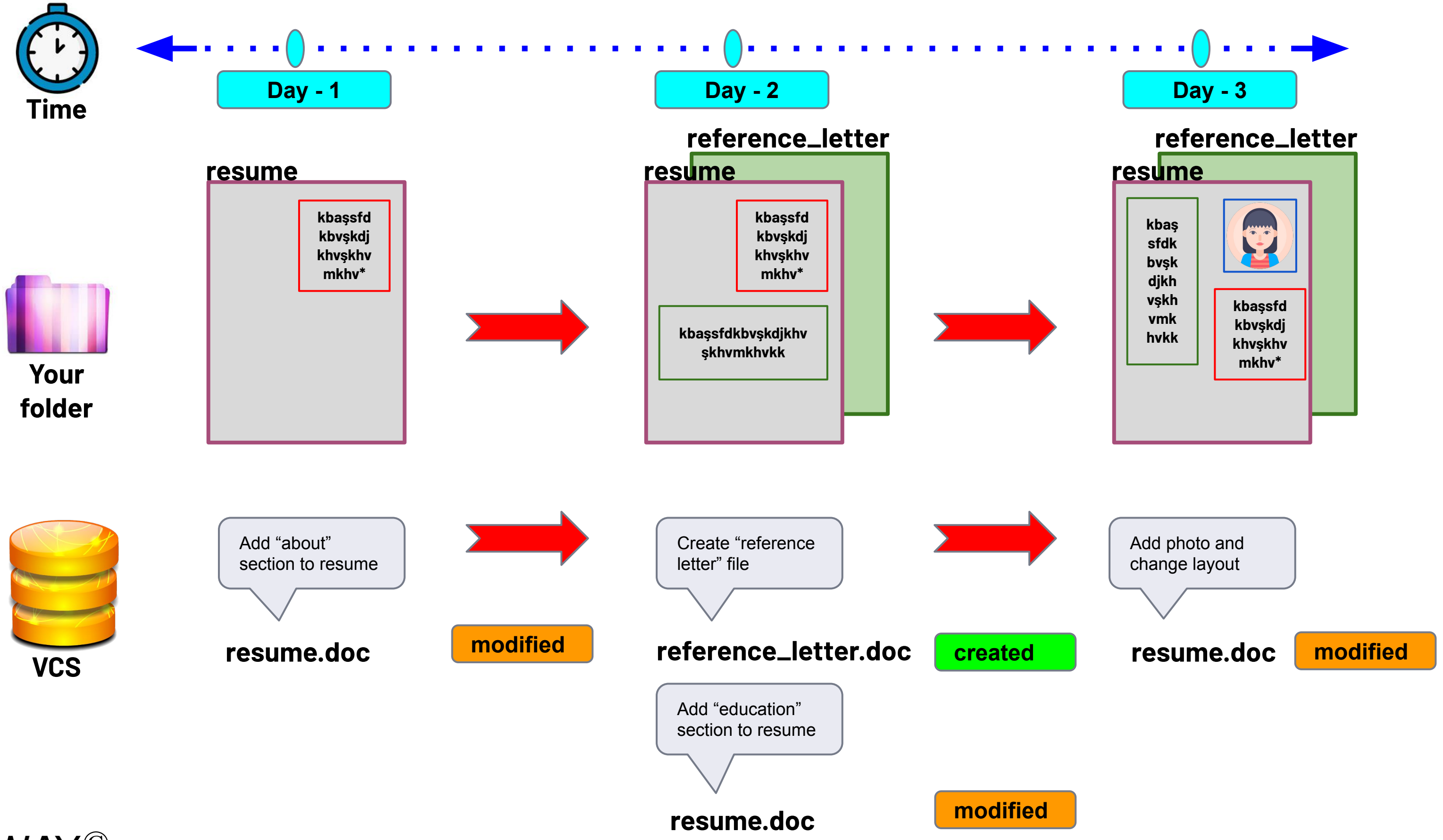
[✓] Keep my .gitconfig updated with my GitKraken Profile preferences

When checked, GitKraken will update your global .gitconfig to match GitKraken Preferences, including commit author name and email, and any GPG settings.

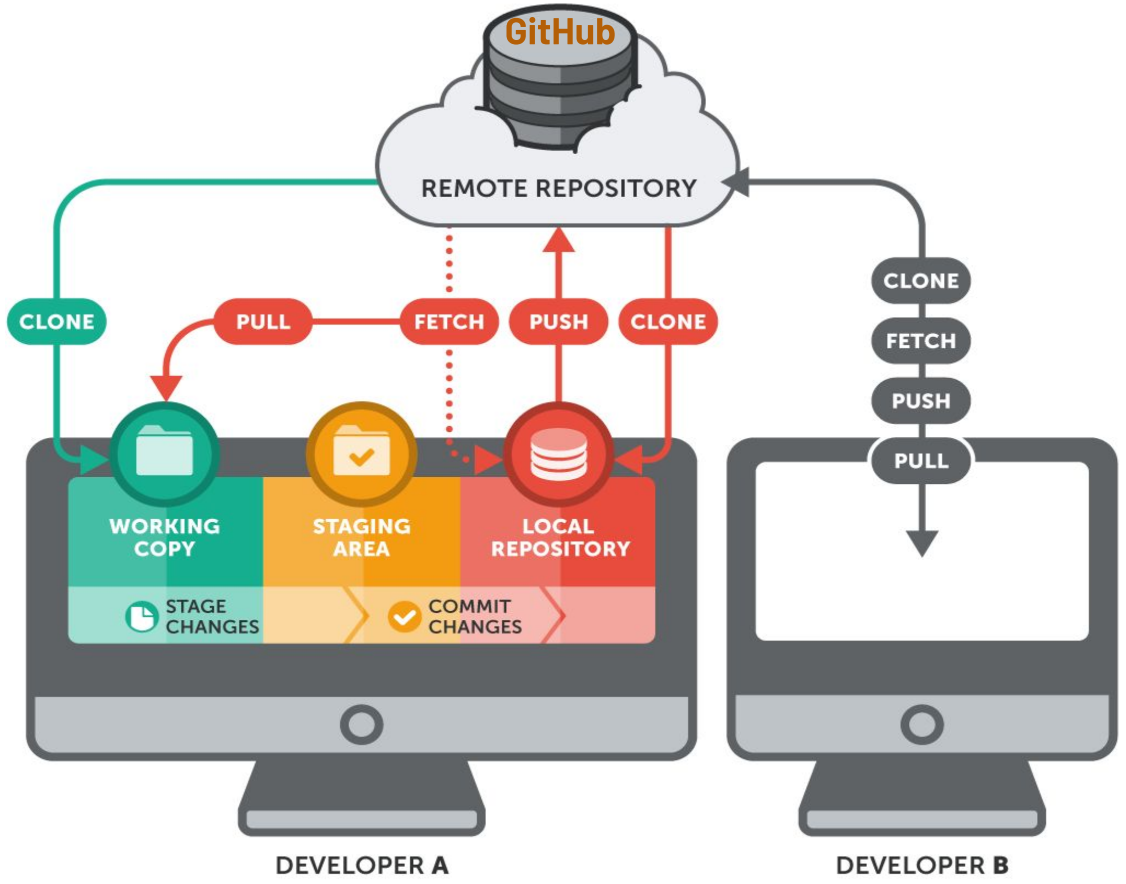
+ Add a profile

Profile Name	Author Name	Author Email	Organization
Clarusway	Clarusway	claruswaygithub...	
Volkan	Volkan	volkansisalan@g...	

What's Version Control?


Time
Day - 1
Day - 2
Day - 3
Your folder
resume
kbaşsfd kbvşkdj khvşkhv mkhv*
reference_letter
resume
kbaşsfd kbvşkdj khvşkhv mkhv*
kbaşsfdkbvşkdjkhv şkhvmkhvkk
reference_letter
resume
kbaş sfdk bvşk djkh vşkh vmk hvkk
kbaşsfd kbvşkdj khvşkhv mkhv*
VCS
Add “about” section to resume
resume.doc
modified
Create “reference letter” file
reference_letter.doc
created
Add “education” section to resume
resume.doc
modified
Add photo and change layout
resume.doc
modified

Git Basics


GitHub
REMOTE REPOSITORY
CLONE
PULL
FETCH
PUSH
CLONE
CLONE
FETCH
PUSH
PULL
WORKING COPY
STAGING AREA
LOCAL REPOSITORY
STAGE CHANGES
COMMIT CHANGES
DEVELOPER A
DEVELOPER B

Local Git operations

git init

git status

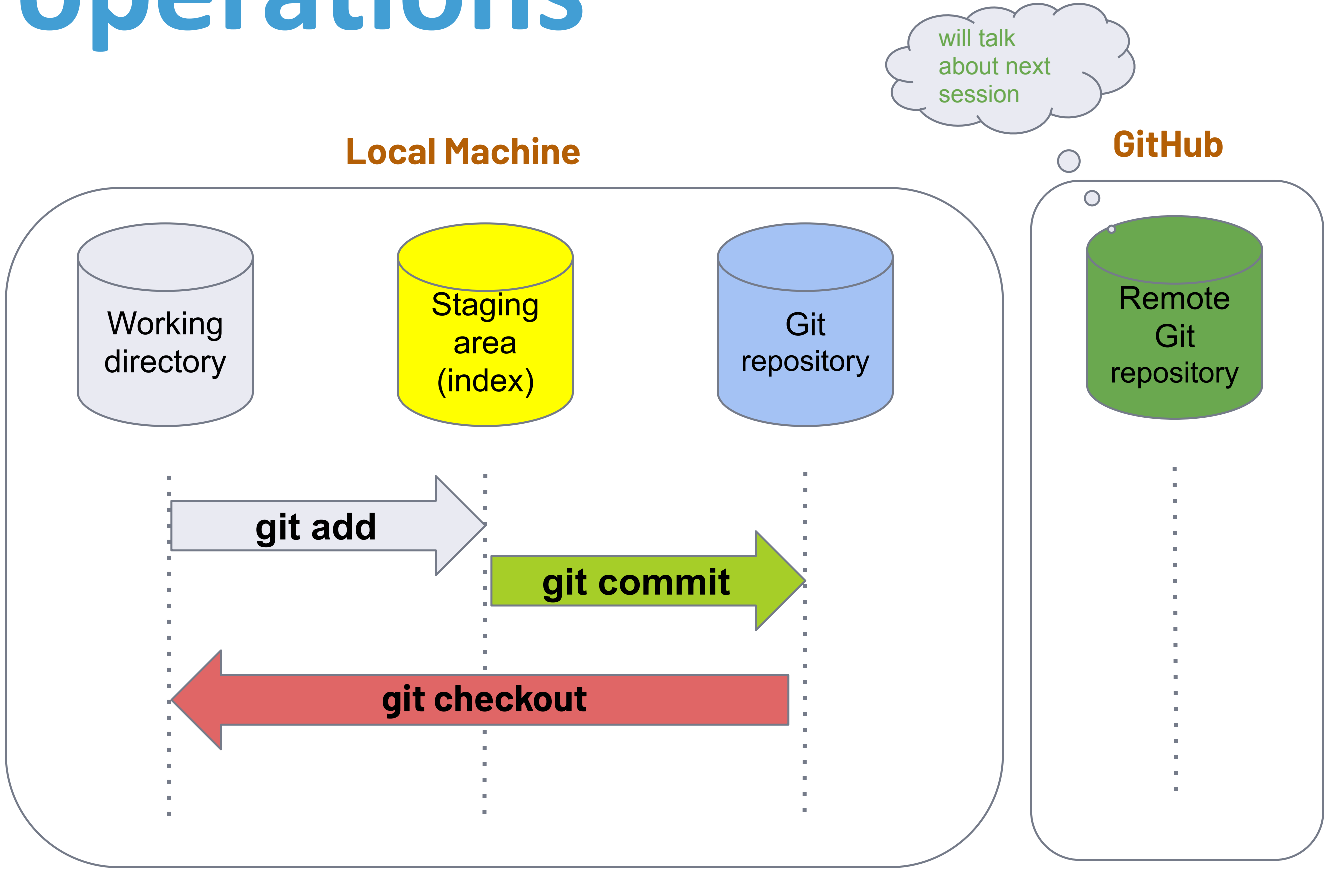
git add .

git rm -cached fileXX

git commit -m “abc”

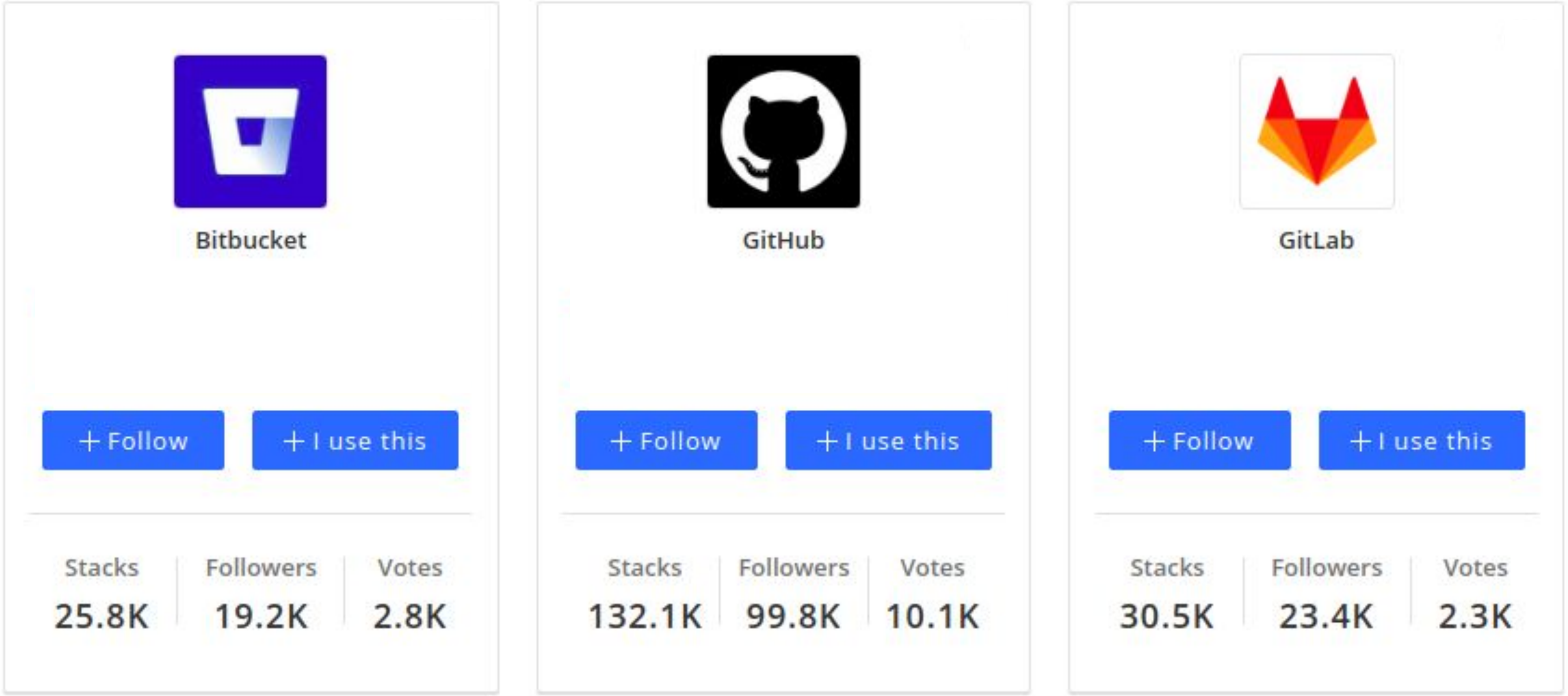
git log

git checkout



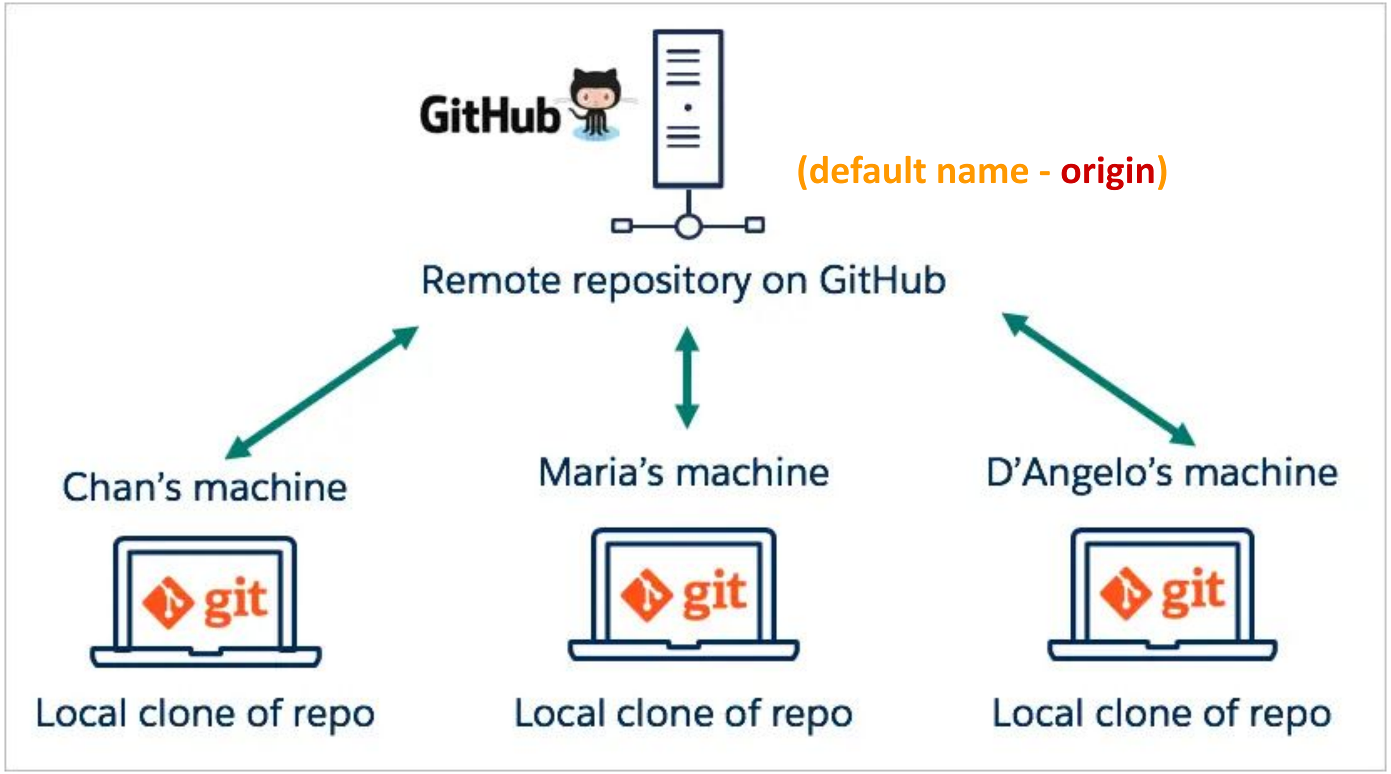
CLARUSWAY©
WAY TO REINVENT YOURSELF

Github - Remote Repository



	Stacks	Followers	Votes
Bitbucket	25.8K	19.2K	2.8K
GitHub	132.1K	99.8K	10.1K
GitLab	30.5K	23.4K	2.3K

Github - Remote Repository



Github - Remote Repository

- Act of copying a repository from remote server to your local machine is called **cloning**
- Cloning allows team to work together
- Downloading commits from others : **fetch, merge**
- Downloading commits from others : **pull (fetch + merge)**
- Uploading your commits (local changes) to remote : **push**

Connecting your local with remote

- connect to remote repo

```
git remote add origin Repo address
```

origin = alias for your repo address

- first push

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

- remove remote origin

```
git remote rm origin
```

CLARUSWAY©
WAY TO REINVENT YOURSELF

Gitignore file usage

How do we use .gitignore file?
How to include multiple files?
How to include folders?
How to include files with specific extensions?

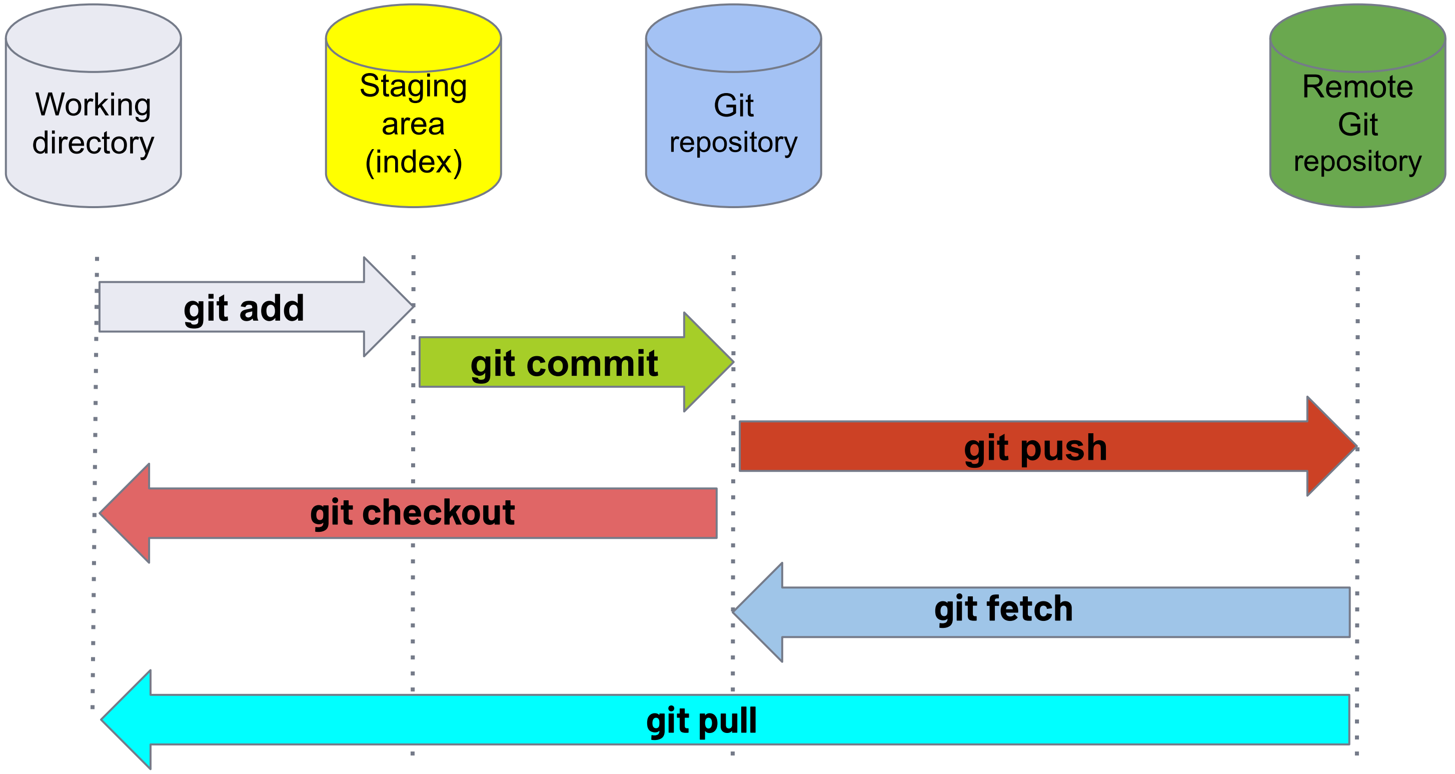
- **Create some files and folders in your local repository.**
- **Create some files with .log extension and exclude them.**
- **Demonstrate how to exclude .log files from add, see the result with status command.**

CLARUSWAY©
WAY TO REINVENT YOURSELF

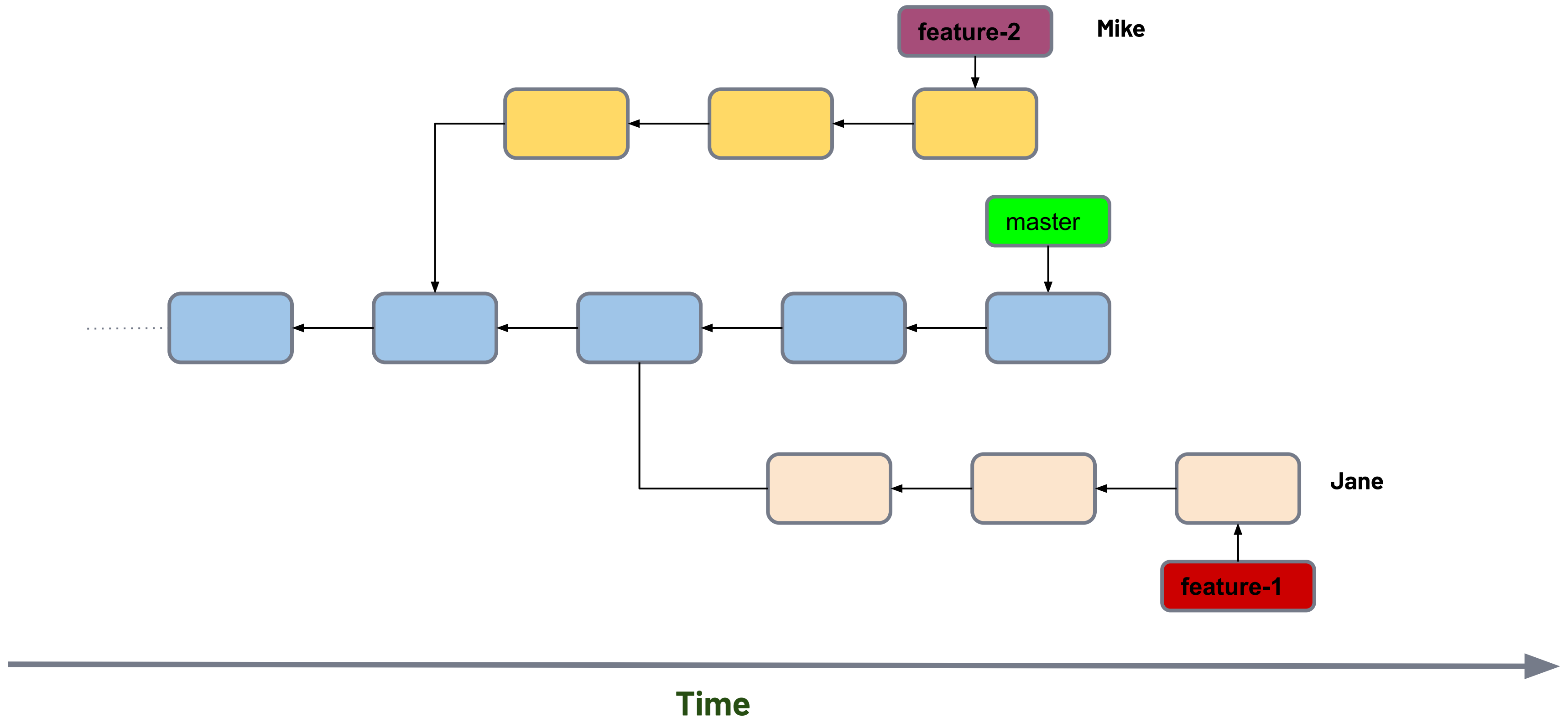
Git stash command

git stash save
Save modified and staged changes
git stash list
list stack-order of stashed file changes
git stash pop
write working from top of stash stack
git stash drop
discard the changes from top of stash stack
git stash clear
clear the entire stash

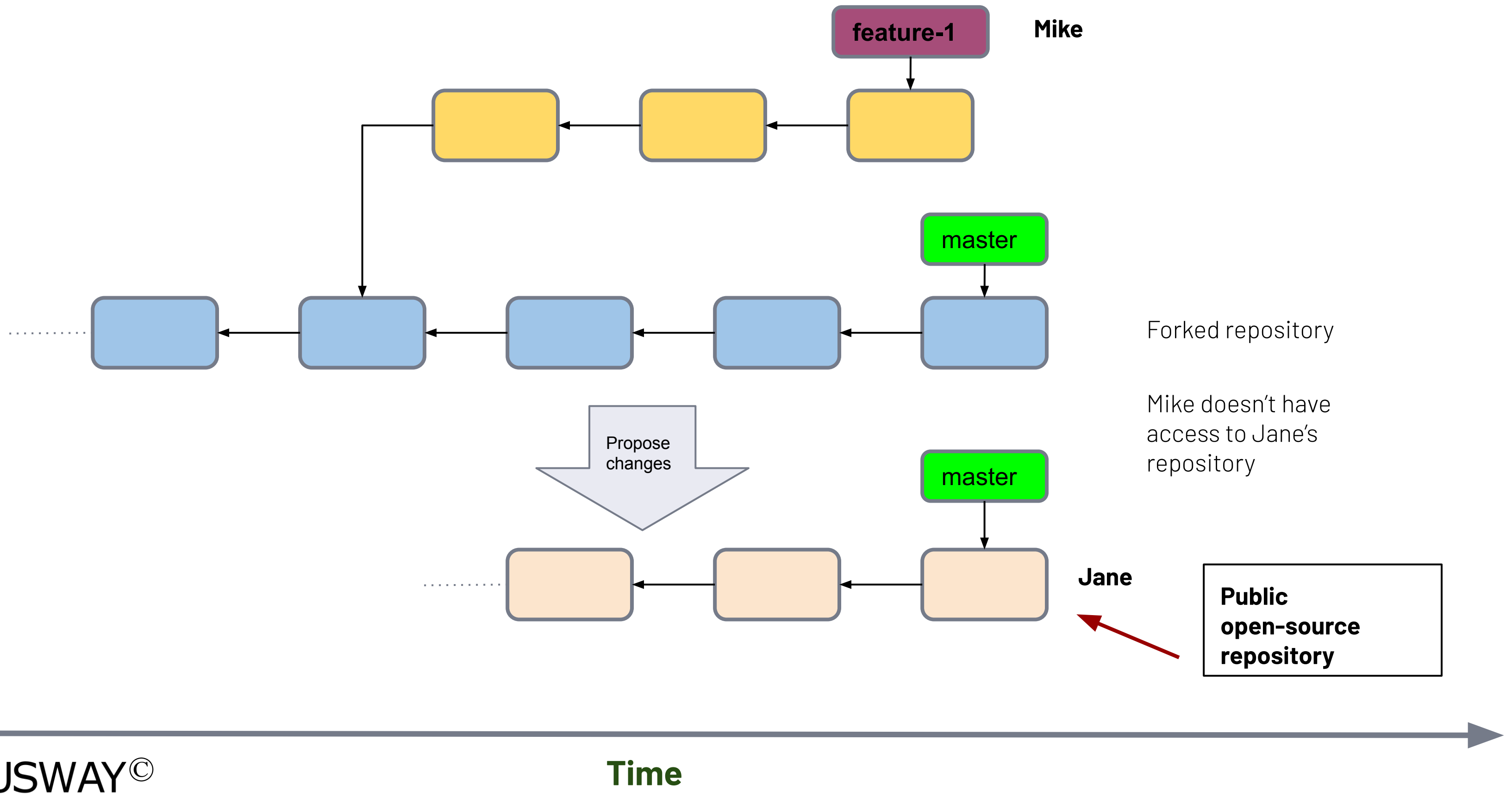
Github - Remote Repository



How collaborators communicate?



How collaborators communicate?



Forked repository

Mike doesn’t have access to Jane’s repository

Public open-source repository

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 - Pull Request process

Time

Mike

Creates new local branch — feature-1

Commits changes to feature-1 locally — commit

Mike is happy with changes and feature works as expected

Pushes changes to remote by creating remote feature-1 branch — feature-1

Creates pull request to start review process by other collaborators

Mike requests Jane to review newly opened pull request

Jane

Jane starts review of the Mike's pull request

Optionally **pulls** updates and checkouts **feature-1** branch to verfy how new feature works.

Add some comments for specific blocks of code and asks for changes — comments

Github - Pull Request process

Time

Mike

Mike is notified about comments and requested changes

Makes additional changes requested by Jane

commit

Pushes changes to remote

Jane

Jane is notified about new commits

Happy with new changes and **approves** pull request

Mike

Merges changes from the feature-1 branch to the main **master** or **release** branch

Closes pull request and deletes feature-1 branch

New feature implemented !

Kahoot

General Git Quiz

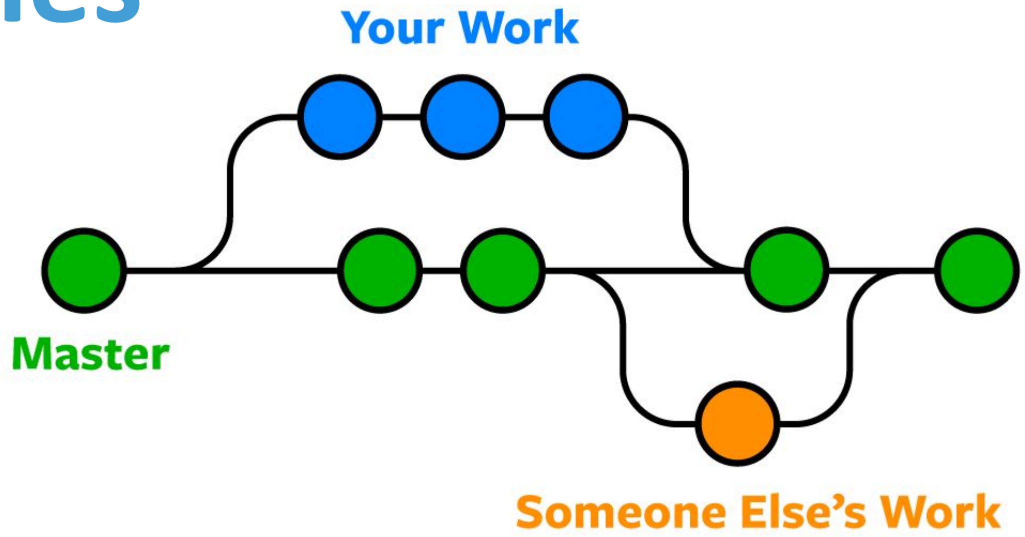
Git

Repo, Commit, Branch, Head

What comes to you your mind when you hear this?

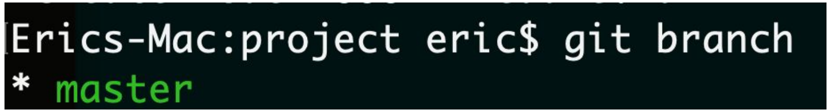
KUSWAY©
REINVENT YOURSELF
Students, write your response!
Pear Deck Interactive Slide
Do not remove this bar

Branches



→ Production of the project lives on master branch

→ Branches are reference to a commit



```
Erics-Mac:project eric$ git branch
* master
```

Branches

→ to see local branches

```
git branch
```

→ to see remote branches

```
git branch -r
```

→ to see all branches

```
git branch -a
```

CLARUSWAY©
WAY TO REINVENT YOURSELF

Creating/switching branches

→ create a new branch

```
git branch Branch name
```

→ switch to a branch

```
git checkout Branch name
```

→ create a new branch and switch to that branch

```
git checkout -b Branch name
```

Deleting branches

- delete a local branch

```
git branch -d Branch name
```

```
git branch -D Branch name
```

- merge a branch

```
git merge Branch name
```

Connecting your local with remote

→ connect to remote repo

```
git remote add origin Repo address
```

origin = alias for your repo address

→ first push

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

→ remove remote origin

```
git remote rm origin
```

CLARUSWAY©
WAY TO REINVENT YOURSELF

Github - Remote Repository

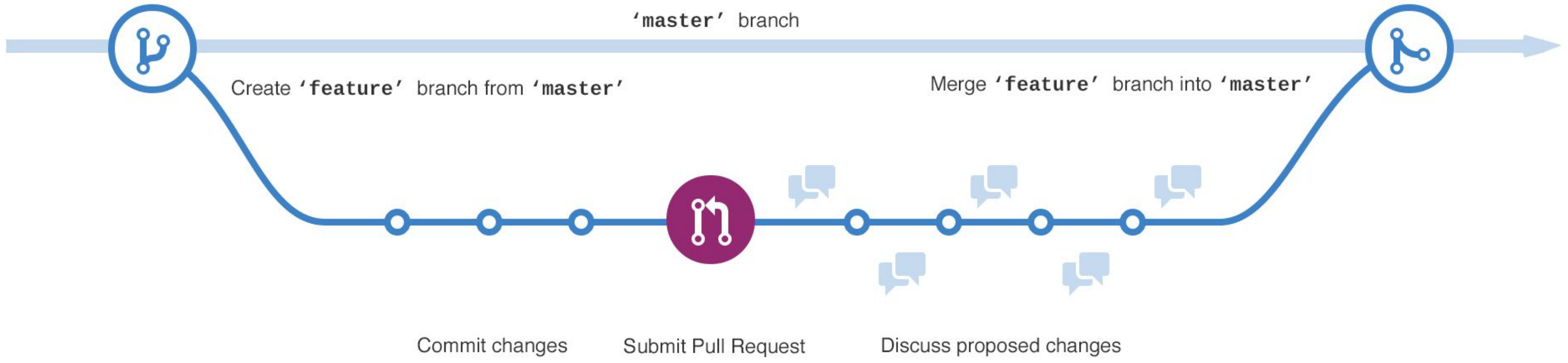
- Act of copying a repository from remote server to your local machine is called **cloning**
- Cloning allows team to work together
- Downloading commits from others : **fetch, merge**
- Downloading commits from others : **pull (fetch + merge)**
- Uploading your commits (local changes) to remote : **push**
- Copying from remote to remote : **fork**

Kahoot

Git and Github terminology

Github - Merge

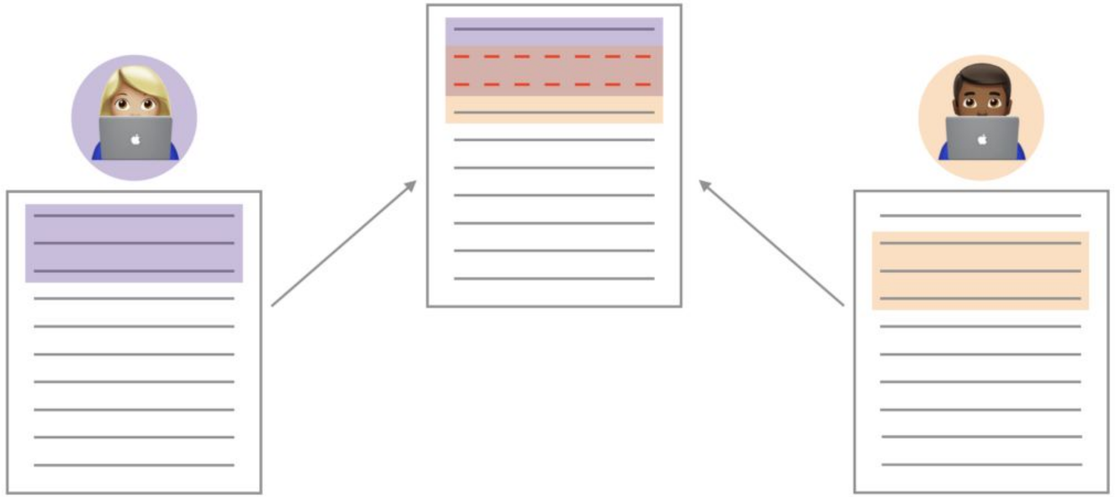
- You **merge** a pull request into the upstream branch when work is completed. Anyone with push access to the repository can complete the merge.


'master' branch
Create 'feature' branch from 'master'
Merge 'feature' branch into 'master'
Commit changes
Submit Pull Request
Discuss proposed changes

CLARUSWAY©
WAY TO REINVENT YOURSELF

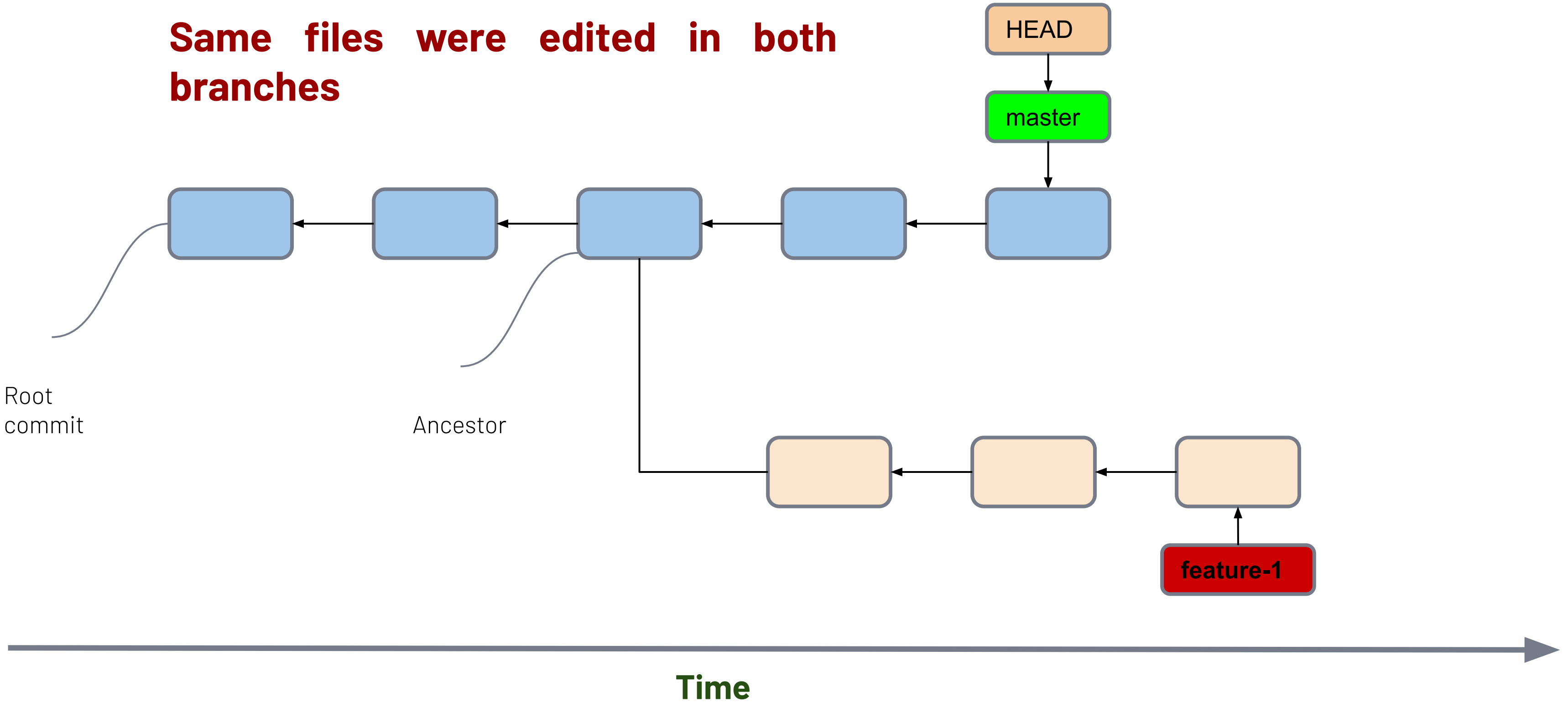
Github - Merge Conflict

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



Merge Conflicts

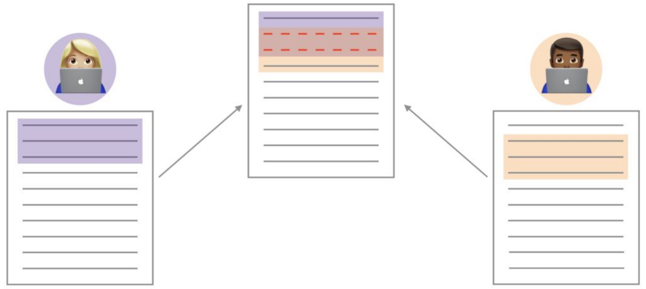
Same files were edited in both branches

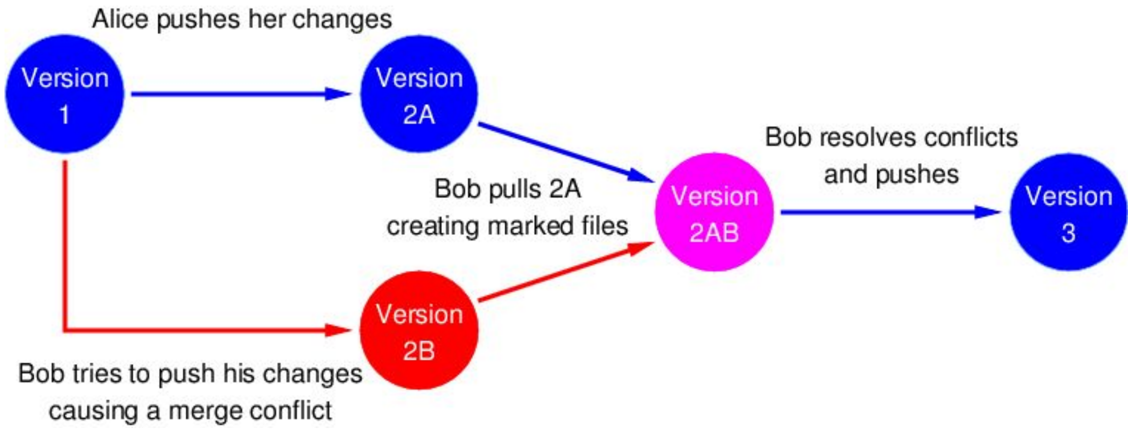


Merge Conflicts

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.

Same files were edited in both branches




Alice pushes her changes
Version 1
Version 2A
Bob pulls 2A creating marked files
Version 2AB
Bob resolves conflicts and pushes
Version 3
Version 2B
Bob tries to push his changes causing a merge conflict

CLARUSWAY©
WAY TO REINVENT YOURSELF

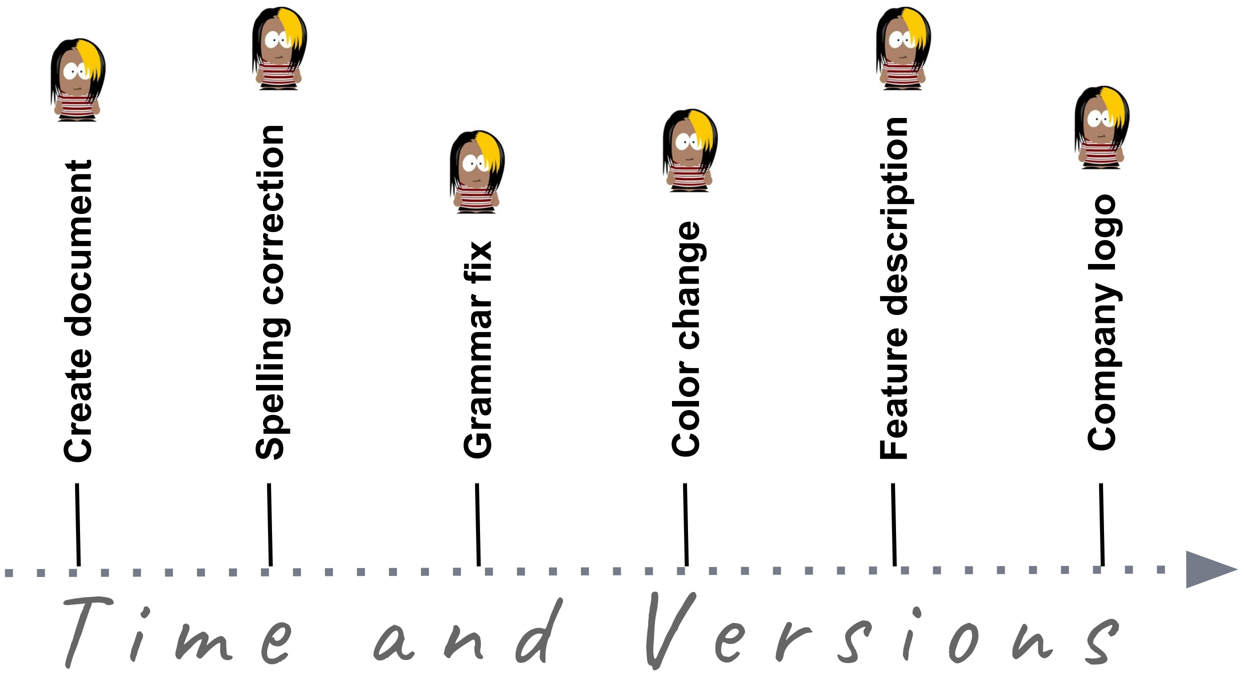
Git

Branch, Head

What comes to you your mind when you hear this?

Git Branches

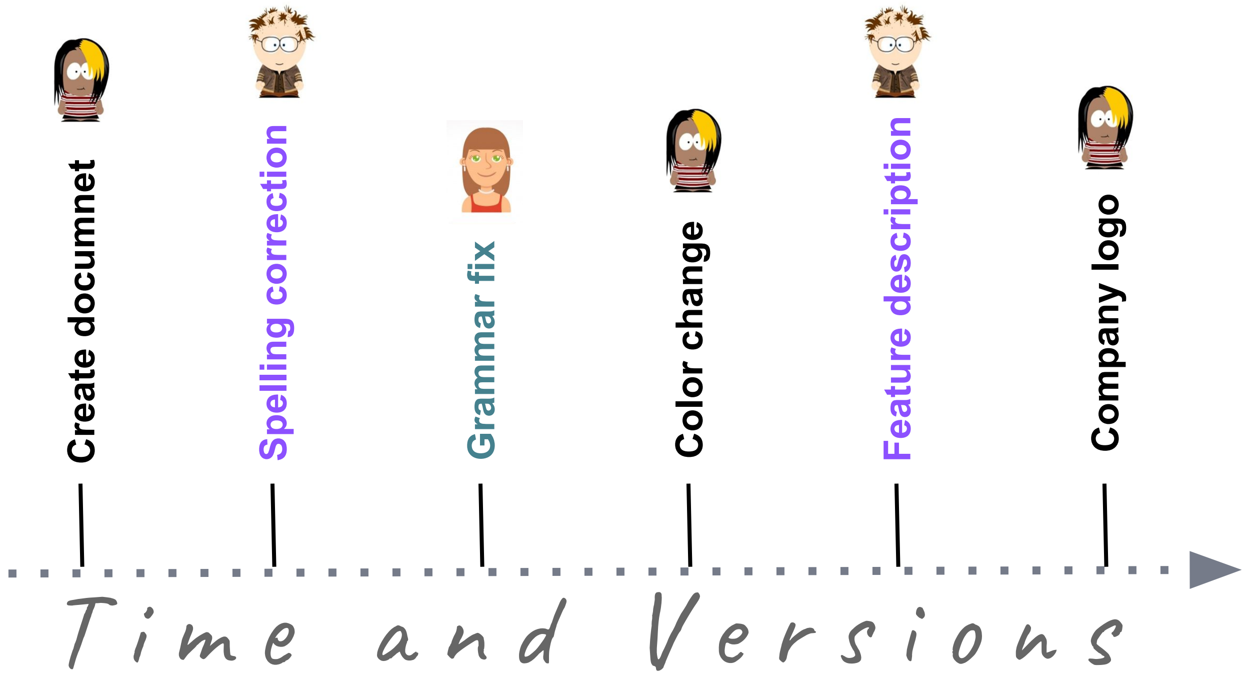
History Tracking



CLARUSWAY©
WAY TO REINVENT YOURSELF

Git Branches

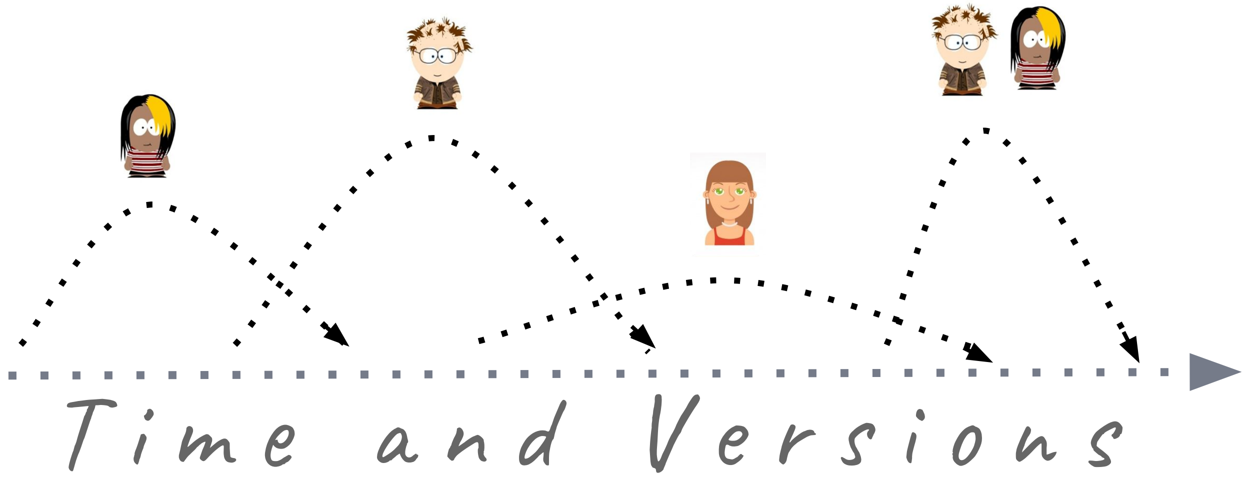
Collaborative History Tracking



CLARUSWAY©
WAY TO REINVENT YOURSELF

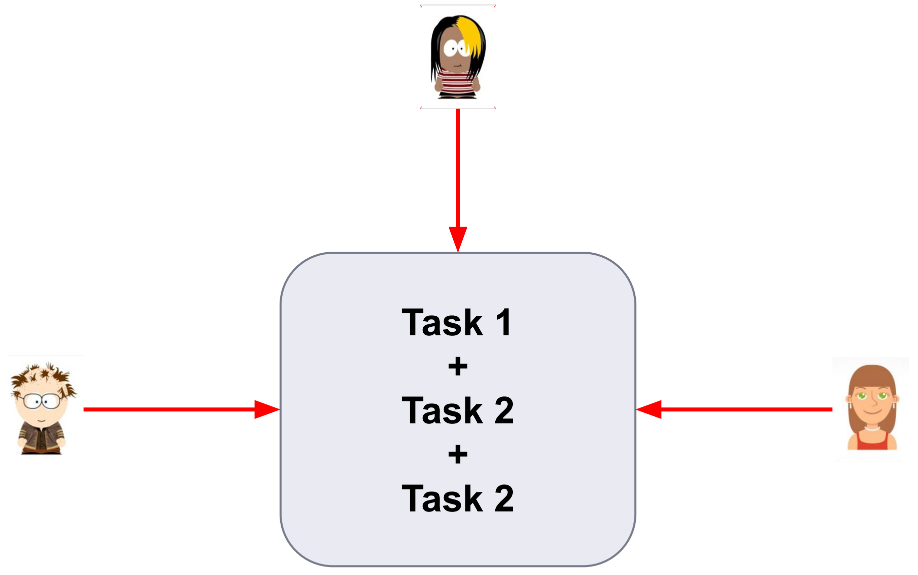
Git Branches

Collaborative History Tracking



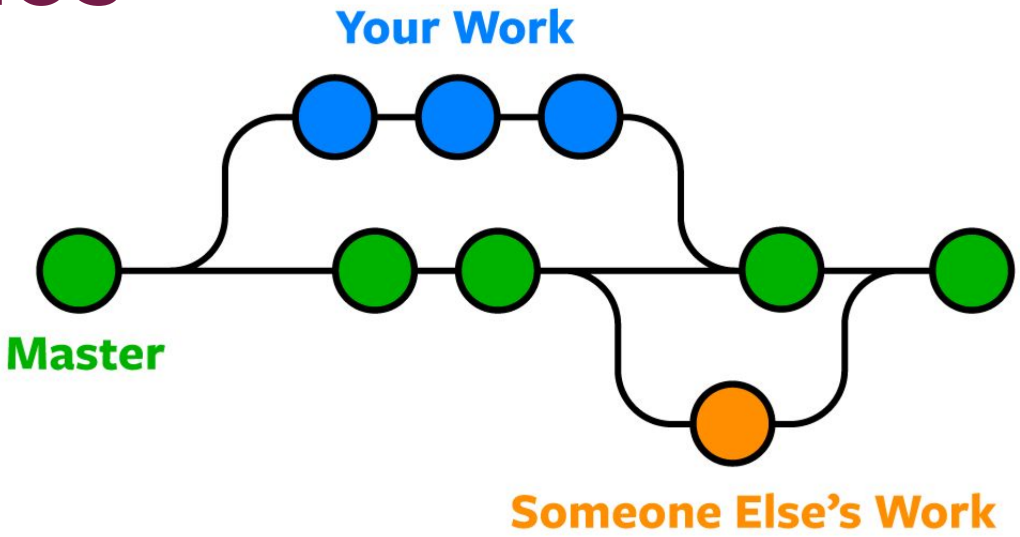
Git Branches

Collaboration

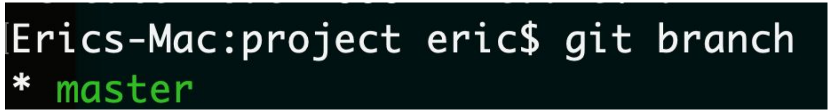


Combined Tasks

Branches



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

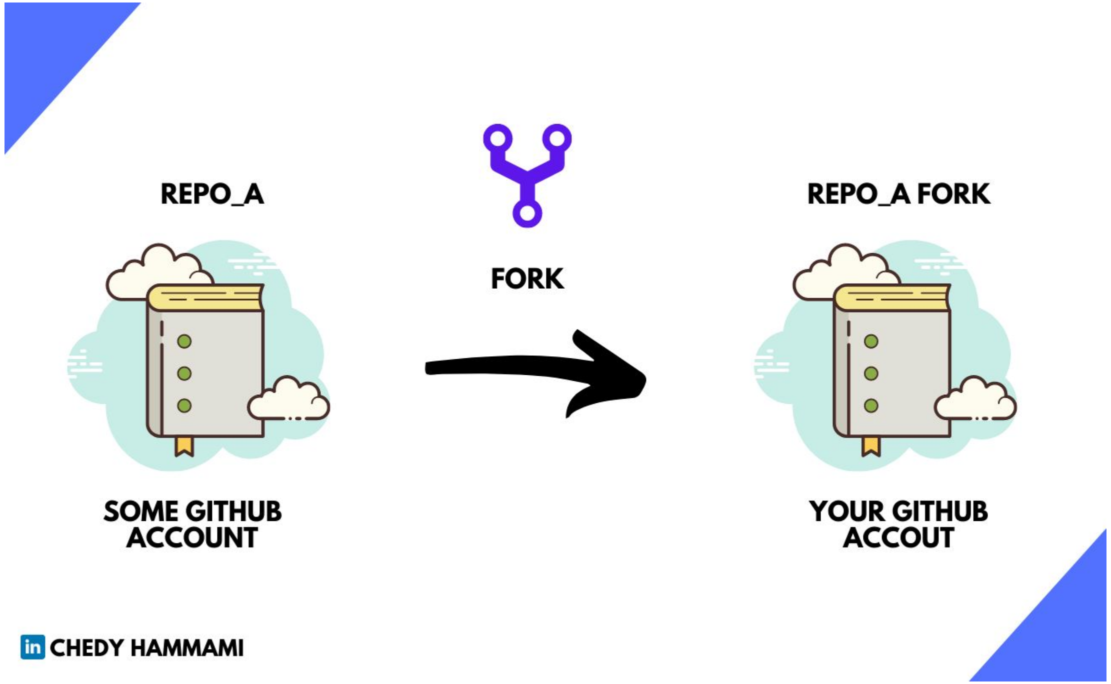


```
Erics-Mac:project eric$ git branch
* master
```

Bonobo Git Server

Lets install Bonobo Git Server on a specific server and publish it.

Github - Fork

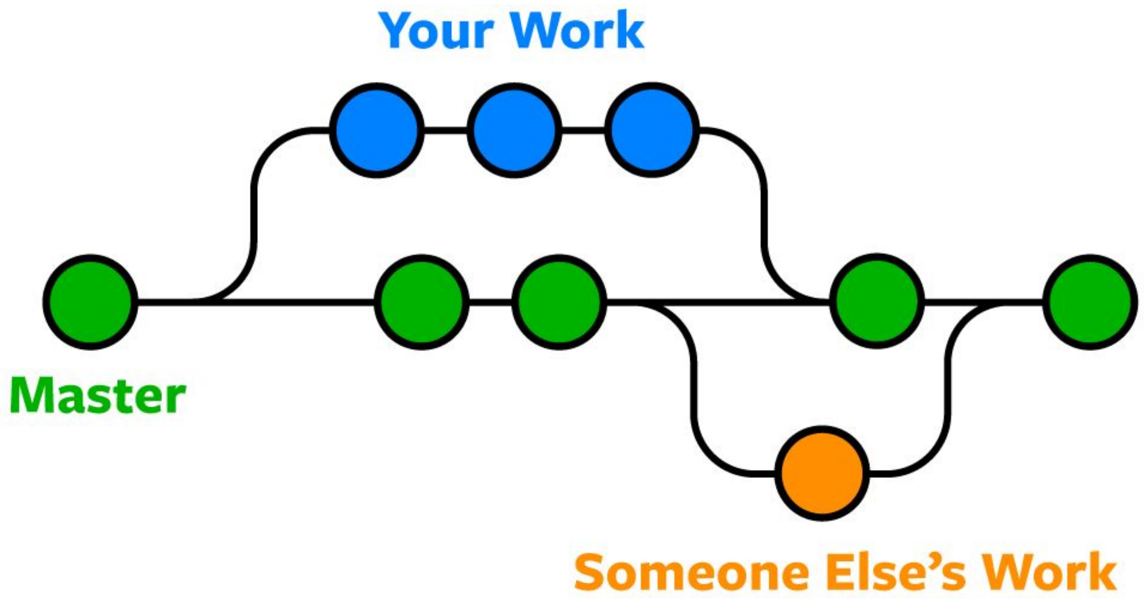


A fork is a copy of a repository.

CLARUSWAY©
WAY TO REINVENT YOURSELF

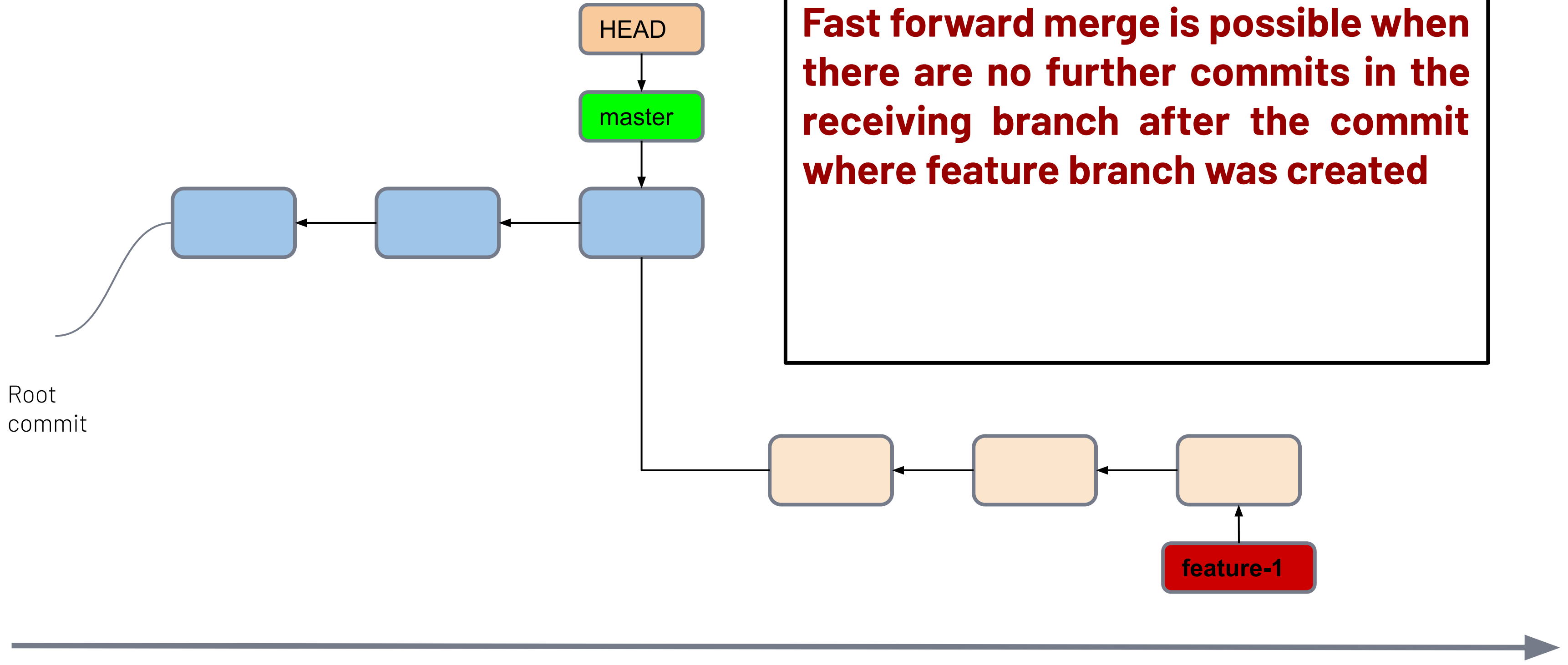
Recap-Branches

```
git branch branch_name
git branch
git branch -a
git checkout branch_name
git checkout -b branch_name
git branch -d branch_name
git branch -D branch_name
git merge branch_name
```



CLARUSWAY©
WAY TO REINVENT YOURSELF

Fast forward merge

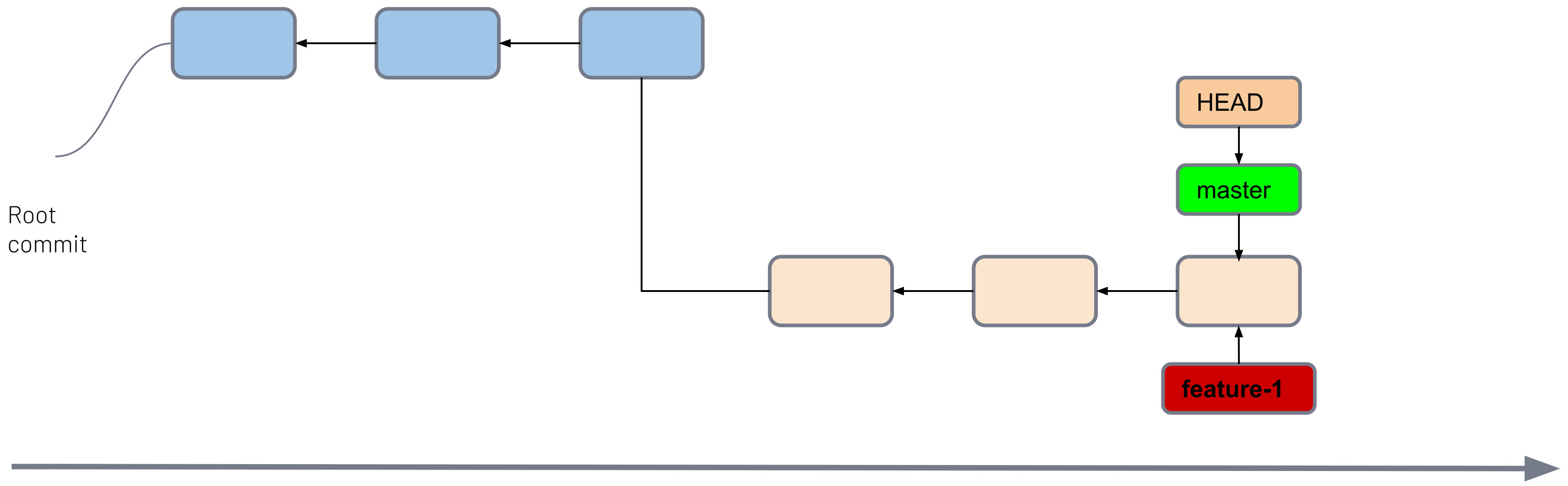


Fast forward merge is possible when there are no further commits in the receiving branch after the commit where feature branch was created

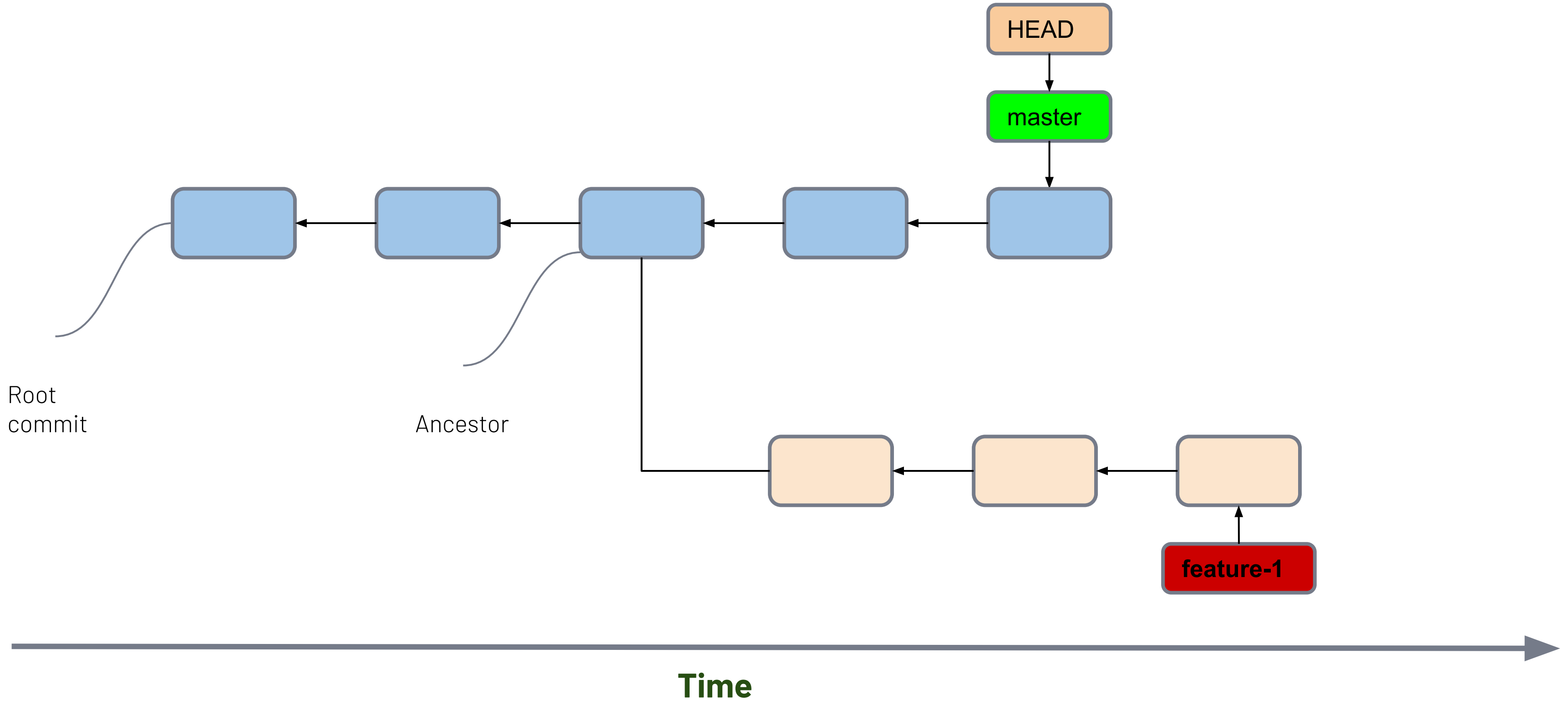
Time

Fast forward merge

git merge <feature-branch>

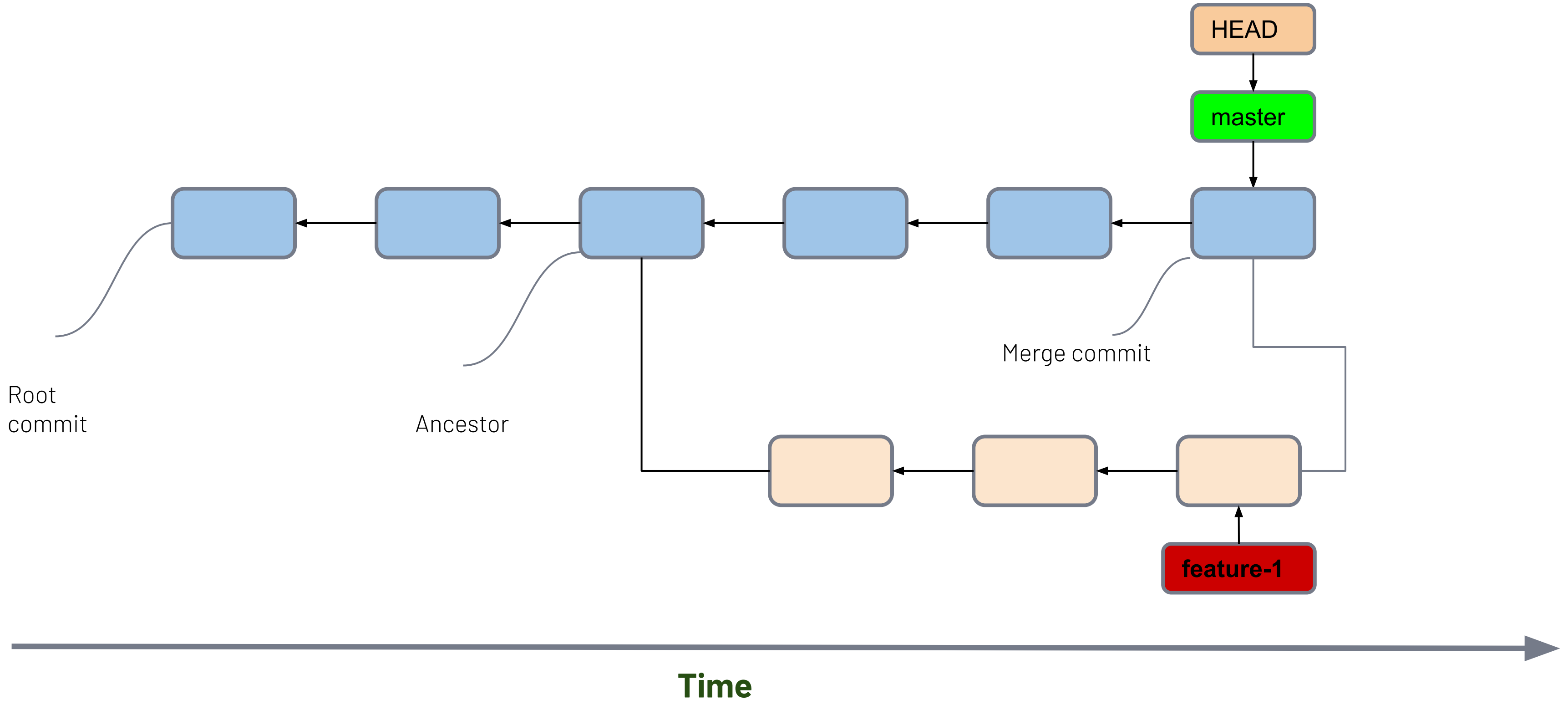


3-way merge



CLARUSWAY©
WAY TO REINVENT YOURSELF

3-way merge



CLARUSWAY©
WAY TO REINVENT YOURSELF

THANKS!