GIT - Two word introduction

Luigi Capogrosso Jasin Atipi

November 10, 2018

What is version control?

- Version control is a tool that allows you to...

What is version control?

- Version control is a tool that allows you to...
- **collaborate**
  Creating anything with other people, from accademic papers to entire websites and application

What is version control?

- Version control is a tool that allows you to...
- **collaborate**
  Creating anything with other people, from accademic papers to entire websites and application
- **Track and revert changes**
  Mistakes happen. Wouldn't it be nice if you could see the changes that have been made go back in time to fix something that went wrong?

Why Git?

- **Fast!** Access information quickly and efficiently

Why Git?

- **Fast!** Access information quickly and efficiently
- **Distributed!** Everyone has her own local copy

Why Git?

- **Fast!** Access information quickly and efficiently
- **Distributed!** Everyone has her own local copy
- **Scalable!** Enables potentially thousands (millions!) of developers to work on a single project

Why Git?

- **Fast!** Access information quickly and efficiently
- **Distributed!** Everyone has her own local copy
- **Scalable!** Enables potentially thousands (millions!) of developers to work on a single project
- **Branches!** Keep your coding experiments separate from code that is already working

Why Git?

- **Fast!** Access information quickly and efficiently
- **Distributed!** Everyone has her own local copy
- **Scalable!** Enables potentially thousands (millions!) of developers to work on a single project
- **Branches!** Keep your coding experiments separate from code that is already working
- Every one has a local copy of the **shared files** and the **history**

Git has its own vocabulary

- A **repository** is where you keep all the files you want to track

Git has its own vocabulary

- A **repository** is where you keep all the files you want to track
- A **branch** is the name for a separate line of development, with is own history

Git has its own vocabulary

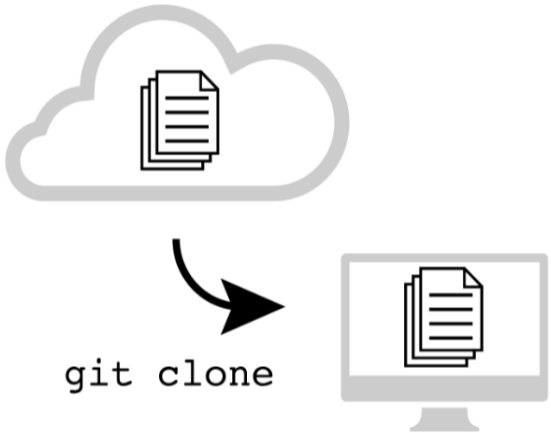
- A **repository** is where you keep all the files you want to track
- A **branch** is the name for a separate line of development, with is own history
- A **commit** is an object that holds information about a particular change

GitHub

- Online git repository
- Free for open source projects

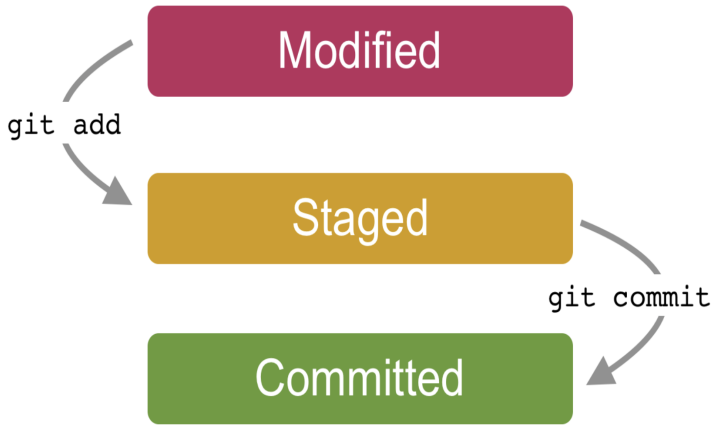
Basics

- `git clone $url`
  copy the whole repository and it's story on the local machine



Basics

- `git add $file`
- `git commit -m "$message"`
  the new release in confirmed and locked in the local repository



Basics

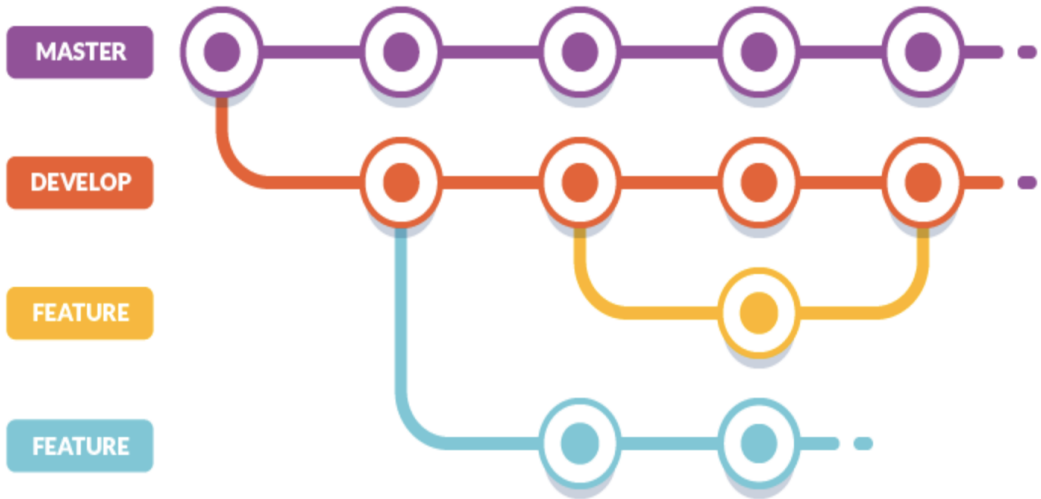
- `git push`
  sends the committed files from the remote repository

Basics

- `git pull`
  downloads the updated files from the remote repository

Basics

- `git branch`
  list all available branches



Basics

- `git checkout $branchname`
  switch from the current branch to `$branchname`

Questions?


THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.