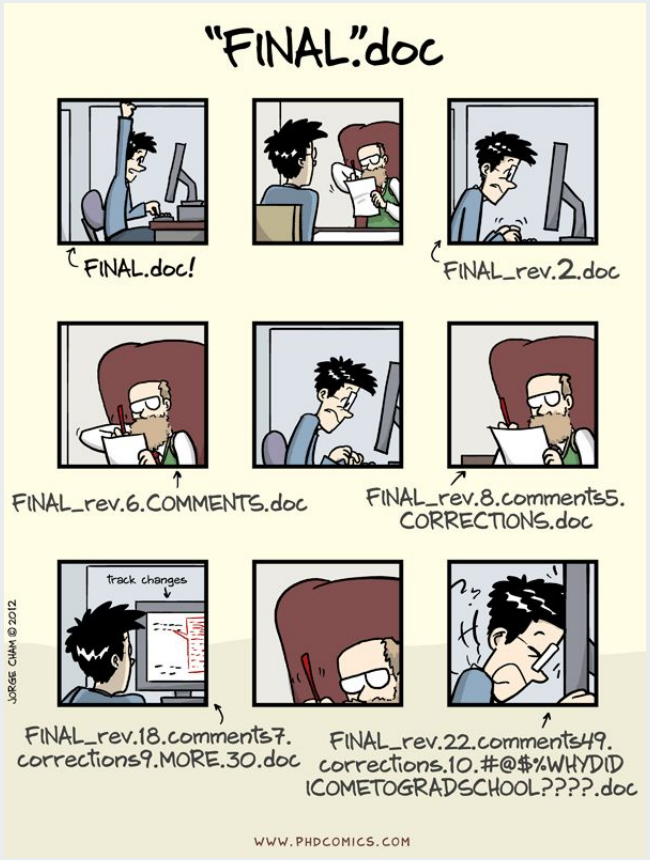
Intro to Git


"FINAL".doc
FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5.
CORRECTIONS.doc
track changes
FINAL_rev.18.comments7.
corrections9.MORE.30.doc
FINAL_rev.22.comments49.
corrections.10.#@$%WHYDID
ICOMETOGRADSCHOOL????.doc
JORGE CHAM © 2012
WWW.PHDCOMICS.COM

Have you ever?

- Made a change to a code, realised it was a mistake and wanted to revert back?
- Lost code or had a backup that was too old? (even on TimeMachine)
- Had to maintain multiple versions of a code?
- Wanted to see the difference between two versions of a code?
- Wanted to prove that a particular change broke or fixed a piece of code?
- wanted to review the history of a code?
- Wanted to submit a change to someone else's code?
- Wanted to share your code, or let other people work on your code?
- Wanted to see how much work is being done, and where, when and by whom?
- Wanted to experiment with a new feature without interfering with a working code?

I these cases, and others, a version control system will make your life a lot easier!

What is version control, git, and github?

Version control

A system that keeps records of your changes, allows for collaborative development, to know who made what changes and when, **to revert any changes and go back to a previous state.**

Git is one of the version control: distributed (users keep entire code and history on their location machines), can make any changes without internet access -> except pushing and pulling changes from a remote server.
Other version control: **Apache subversion (svn)**, **mercurial**, **perforce**

Github, one of the web-based git repository hosting service (free public repositories), with additional features: UI, documentation, bug tracking, feature requests, pull requests, web for the hosted application.
Other git hosting service: **gitlab**, **bitbucket**

Install Git

https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

e.g. Linux (Debian): *sudo apt-get install git*

Tutorial:

https://git-scm.com/book/en/v2

https://try.github.io/levels/1/challenges/1

https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners

Getting Started with Git

First time configuration:
https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup

Start your “Git-ed” project/code:
https://git-scm.com/book/en/v2/Git-Basics-Getting-a-Git-Repository

How does Git work?

Can be complicated at first, but there are a few key concepts

How does Git work?

Key concepts: Snapshots

- The way git keeps track of your code history
- Essentially records what all your files look like at a given point in time
- You decide when to take a snapshot, and of what files (.gitignore)
- Have the ability to go back to visit any snapshot
    - Your snapshots from later on will stay around, too

How does Git work?

Key concepts: Commit

- The act of creating a snapshot -> (can be “verb” or “noun”)
- Essentially, a project is made up of a bunch of commits
  - Before a commit, you need to tell which files that you want to add on to the “index” or “staging”.

How does Git work?

Key concepts: Repositories

- Often shortened to ‘repo’
- A collection of all the files and the history of those files
  - Consists of all your commits
  - Place where all your hard work is stored
- Can live on a local machine or on a remote server (GitHub!)
- The act of copying a repository from a remote server is called cloning
  - Cloning from a remote server allows teams to work together
- The process of downloading commits that don’t exist on your machine from a remote repository is called pulling changes
- The process of adding your local changes to the remote repository is called pushing changes

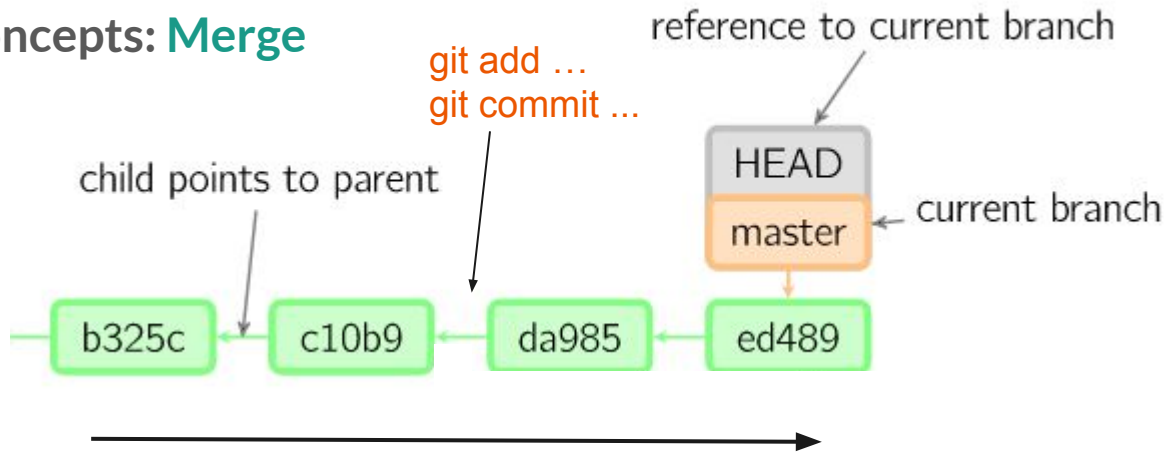
How does Git work?

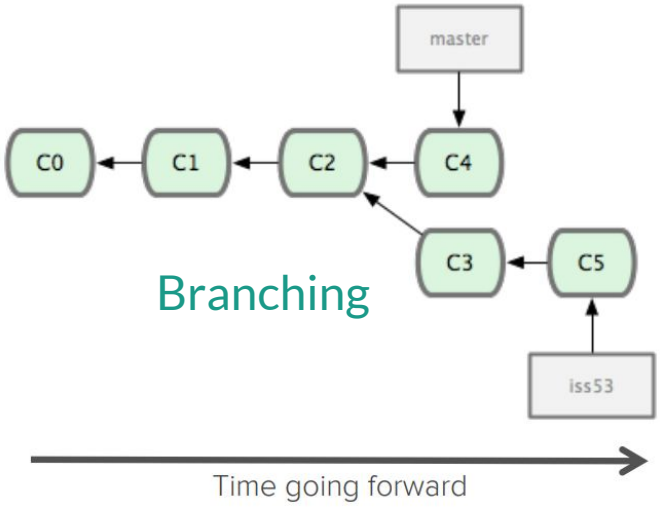
Key concepts: Branches

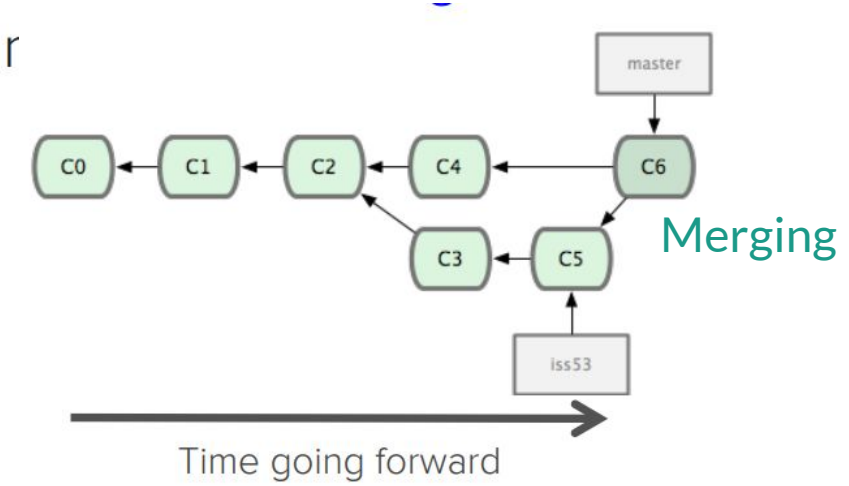
- All commits in git live on some branch, but there can be many, many branches
- The main branch in a project is called the master branch

How does Git work?

Key concepts: Merge







Let’s try the simplest workflow with Git!

Getting Started with GitHub

Make an account!

Tutorial:
https://guides.github.com/activities/hello-world/
https://try.github.io/levels/1/challenges/1
https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners

https://help.github.com/articles/adding-an-existing-project-to-github-using-the-command-line/
https://help.github.com/articles/adding-a-remote/

Let's try a simple collaboration on GitHub

Other tools

GUI e.g. gitKraken, smartGit, etc