

# Introduction to Git

by Ty-Lucas Kelley

# What is Git?

Before I explain Git, let's talk **version control**

# Version Control System (VCS)

A version control system is a tool to manage changes to your code repositories

- snapshots of your code whenever you want
- revert to old versions if new one sucks
- create temporary clones to test new features on
- keep track of contributors

# Back to Git

Git is a **distributed, open source** version control system designed to always be efficient and easy to use.

- loads of features
- fast performance
- tiny footprint

(sorry if I sound like a car salesman)

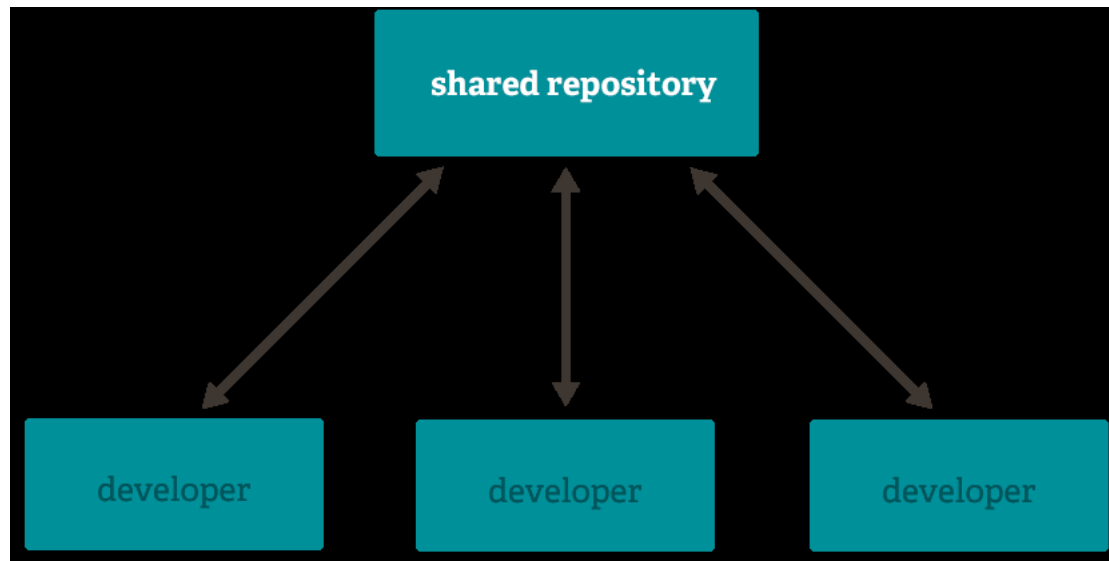
# Fun facts about Git

- created by Linus Torvalds
- 10 years old
- built to replace BitKeeper
  - proprietary VCS that the Linux project used
- used by tons of major companies, including Google, Amazon, and Microsoft



# “Distributed”?

- every developer works on their own local copy (clone) of the code that's on the server
- make changes in your copy and then merge those changes back into the main repository later
- no passing around code on USB drives or emailing files



# Let's install Git

## 1. Linux

- `apt-get install git` **or** `yum install git`,  
etc.

## 2. Mac OS

- `brew install git` **or** `git` (will install if you don't have it already)

## 3. Windows

- <http://git-scm.com/download/win>



# Starter project

Let's make an HTML resume!

- viewable on a desktop or phone
- easily export to PDF for printing
- shows off your HTML and CSS skills

# Getting started

1. Open the terminal (Git Bash on Windows):
2. Set your name
  - `$ git config --global user.name "Your Name"`
3. Set your email
  - `$ git config --global user.email "me@example.com"`

# Cloning the repository

“Cloning” is just a term for creating your own **local copy** of a code repository. Let’s grab the starter code like this:

```
$ git clone https://github.com/hackers-of-umass/resume-template
```

```
tylucaskelley@tlk → hackers-of-umass git clone https://github.com/hackers-of-umass/resume-template
Cloning into 'resume-template'...
remote: Counting objects: 11, done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 11 (delta 0), reused 11 (delta 0), pack-reused 0
Unpacking objects: 100% (11/11), done.
Checking connectivity... done.
```

# Where'd the code come from?

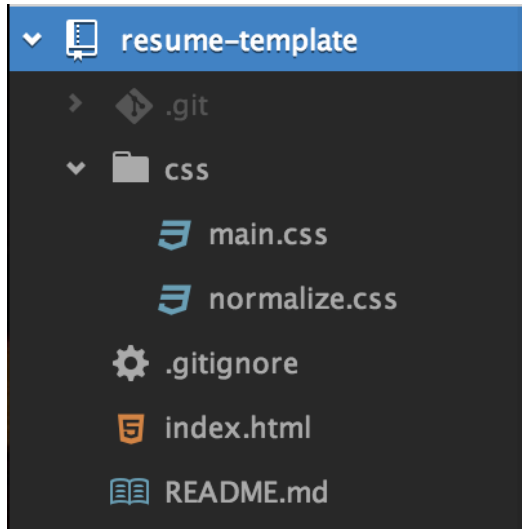
Our project is hosted on [GitHub](#):

- host unlimited public Git repos for free
- pay monthly for private repos
- easily work with collaborators remotely
- browse cool open source projects
- not the same thing as Git!
  - one is a VCS, the other is a hosting service

# Making your first changes

Open the project you cloned in a text editor

- vim
- emacs
- Sublime
- Atom
- etc...



# See what it currently looks like

Before you start making changes, open the `index.html` file in a web browser:

- we can refresh the page every time we make a change to see updates

<% name %>

{ <% title %> }

 <% website %>

 <% twitter handle %>

 <

---

## work experience

<% company %>: <% position %>  
{ <% start %> - <% end %> }

<% description %>

<% company %>: <% position %>  
{ <% start %> - <% end %> }

<% description %>

## education



# Add your name!

In your text editor, replace “<% name %>” on line 15 of `index.html` with your actual name.

- refresh the web browser to see it update!

# Adding your changes to the repo

## 1. `git status`

- see what files have changed since last commit

## 2. `git add index.html`

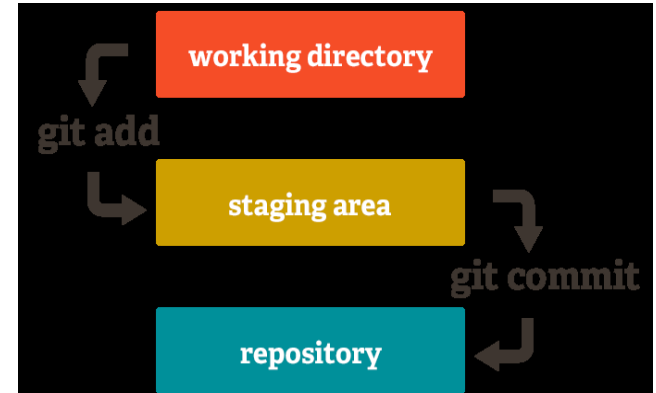
- pick which files are going to be committed to the repo, put them in the “staging area”
- can alternatively add all files with `git add -A`

## 3. `git commit -m “change name on resume”`

- save and describe the changes you made

# What just happened?

We made changes to the code, told Git that we planned on committing those changes eventually (`git add`), and then committed them (`git commit`), which essentially “saves our progress” in case we ever needed to go back to an old version.



# git log

```
commit c904b7a618168d27a13ac6672477419d04b68977
Author: Ty-Lucas Kelley <tylucaskelley@gmail.com>
Date: Tue Mar 24 10:30:51 2015 -0400
```

changed name

```
commit 1f8d0fddd990c46d35d62964d7894729411177af
Author: Ty-Lucas Kelley <tylucaskelley@gmail.com>
Date: Tue Mar 24 09:06:20 2015 -0400
```

template done

```
commit dc4234d7790f71254390aa0e805ffe4ab0f37ca4
Author: Ty-Lucas Kelley <tylucaskelley@gmail.com>
Date: Tue Mar 24 08:40:28 2015 -0400
```

first commit

(END)

# Storing your code remotely

1. Create an account at <https://github.com> and create a new repository called “my-resume”
2. Tell Git that the repository you created is where your code should point to:
  - a. `git remote set-url origin https://github.com/USERNAME/my-resume`
  - b. `git remote -v` to see updated URL
3. Push changes with `git push`!



This repository Search

Explore Gist Blog Help



tylucaskelley



tylucaskelley / resume-template

Unwatch

1

Star

0

test — Edit

3 commits

1 branch

0 releases

1 contributor



branch: master

resume-template / +



changed name



tylucaskelley authored 43 minutes ago

latest commit c904b7a618



css

template done

2 hours ago



.gitignore

template done

2 hours ago



README.md

template done

2 hours ago



index.html

changed name

43 minutes ago

README.md

# resume-template

build an HTML resume

Clone the project and make changes to `index.html` and `main.css` to build your resume.

Code

Issues

0

Pull requests

0

Wiki

Pulse

Graphs

Settings

# What if I mess up?

Let's say you do something to a file and want to revert it to the state it was in at the most recent commit:

- `git checkout -- <filename>`

# Branching and merging

Let's say that the main (a.k.a. “master”) branch of your project is the version that gets deployed to users.

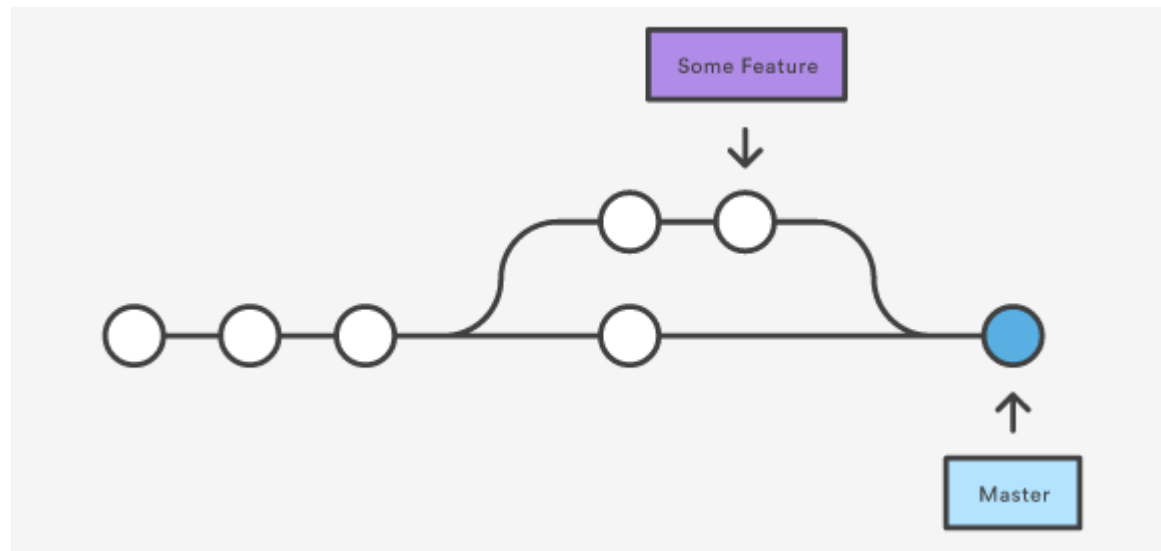
- we don't want it to be broken!



# Branching and merging (pt. 2)

Since we don't want to break the master branch, but still need to do things like develop new features and fix bugs, we can “branch out” from master to make separate changes and commits to.

- and then merge those branches back into master when they're ready



# Creating a branch

Let's work on the resume some more. We're going to create and switch to a branch called `work-experience` to add job info to:

```
$ git checkout -b work-experience
```

```
tylucaskelley@tlk → resume-template git:(master) git checkout -b work-experience  
Switched to a new branch 'work-experience'  
tylucaskelley@tlk → resume-template git:(work-experience) |
```

# Making changes

Add some work experience to the resume by editing `index.html`, then add and commit your changes like before:

```
$ git add -A
```

```
$ git commit -m "add work experience"
```

# Merging back into master

Once you're happy with the changes you've made on your branch and committed everything, go back to master:

```
$ git checkout master
```

## Merging back into master (pt. 2)

And then merge your changes, which will update the master branch:

```
$ git merge work-experience
```

You can then delete the old branch with:

```
$ git branch -D work-experience
```

**\$ git push**

# Some other useful commands

- `git diff <options>`
  - see detailed changes to files between branches, commits, etc.
- `git log --oneline --decorate --stat`
  - more readable git log output with file change details
- `git tag -a 1.0 -m "release version 1.0"`
  - mark current commit as a “version” of your code, for organization (helps if you’re releasing an actual product)



# What do I do now?

- Start building some open source projects and putting them on GitHub!
  - companies look for it when hiring
  - shows off your skills and interests
  - other people can use and contribute to your projects

# Useful links

- Git documentation: <http://git-scm.com/doc>
- Git Tutorial: <http://try.github.io>
- Hackers of UMass GitHub organization:  
<https://github.com/hackers-of-umass>
- Facebook group: <https://www.facebook.com/groups/hackersofumass/>