Get Acquainted with Git

http://teamtreehouse.com/library/ajax-basics

What is Git?

version control system (vcs)

System to keep track of changes to files.

Let's you undo mistakes, work on different versions of a project, and work with a team of people without stepping on each other's code.

Course

Console Foundations

The console is an important skill for any developer. Many programs can only be used via a command line interface, and often the only access you have to a server will be over a command line interface. Once you are familiar with the basics of the console, you will be able to perform very powerful tasks quickly and easily.

Development Tools

318 318

5 Achievements

Getting Started with the Console

The command line and console can be a scary thing to get started with. Once you understand the history of the command line, and what it's good for, you can get familiar very quickly. The command line provides you the ability to interact with your computer using text commands, often making actions that would be difficult with menus or mouse clicks simple.

Show 12 Steps

Users and Permissions

Almost all computers have the idea of user accounts. Along with that comes permissions which define which users can interact with the different files and programs on the computer. It's important to know how to setup and manage users, and how permissions are applied to files and folders.

Instructor

Jim Hoskins

Jim is a full stack software developer at Treehouse. When he's not writing code, he's blogging, teaching, or speaking at conferences. On Twitter he is @jimrhoskins.

Download videos

Download all the videos from this collection using iTunes. Continue watching videos even when an internet connection is unavailable.

Requires **Apple iTunes**.

Course

Git Basics

If you're serious about pursuing a career as a software developer or designer, at some point, you will need to learn a version control system. A VCS is an important, but sometimes overlooked, tool that is essential for keeping all but the most trivial projects on track. In this course we'll talk about what version control is and how it works before covering the basics of one of the most popular version control systems available today - Git. By the time you're finished with the course, you'll be ready to start using Git on your next project.

Development Tools

Watch Trailer

390

390

6 Achievements

Why Version Control Matters

One of the most essential tools a developer can master is version control, and one of the most widely-used version control systems available today is Git. But what makes Git popular? And why should you learn to use version control anyway? In this stage we'll explore all the benefits of using version control (including: helping you deal with changing requirements, data loss, sharing and collaborating with others), as well as the qualities of Git that make it such a popular choice in the modern development world.

Getting Started With Git

What you'll learn

- What Version Control Does
- How Version Control Works
- What Makes Git a Good Choice
- Using Git to Manage Basic Projects
- Working With GitHub and other Git Hosting Solutions

http://teamtreehouse.com/library/git-basics

Instructor

Tommy Morgan

Tommy's been a software developer for seven years, and was working with a

Installing Git

- **Mac**
  It's installed!

- **Windows**
  http://teamtreehouse.com/library/git-installation-windows

Configuration

```
git config --global color.ui true
git config --global user.name <your_name>
git config --global user.email <your_email>
```

Repository or **“repo”**

The folder containing the files you are tracking — the folder for your project, for example.

Create a repository

```
cd ~/Documents/my_project
git init
```

.gitignore

A file which lists files and file extensions that Git should not track in the repository. “Ignore these files.” Place in the repository folder.

https://github.com/treehouse-dave/gitignore/archive/master.zip

.gitignore (very minimum for a Mac)

```
.DS_Store
```

commits

Keep track of your changes by “committing” changed files to the repository.

commit early, commit often

Your First Commit

```
git add .
git commit -m "Initial Commit"
```

What is Happening in the Repo?

```
git status
```

Add One File

```
git add index.html
```

Commit

```
git commit -m "Add headline to home page"
```

The States of a File in a Git Repo

- **untracked** New file that's never been added to the repo
- **working** Tracked file with new changes
- **staged** File that has been added to the "staging area"
- **committed** File is added to repo and is now tracked by Git

staging

Collect related files that you want to commit together

Let’s Check Our Commits

```
git log
git log --oneline
```

When things go wrong

I Messed Up My Working File (not yet staged)

```
git checkout <path/to/file>
```

I Messed Up A File and Staged It

```
git reset HEAD <path/to/file>
git checkout <path/to/file>
```

Arrgggg. I Messed Up File and Committed It!

```
git checkout HEAD^ <path/to/file>
```

Oh man. I really messed up. Undo commit.

```
git reset --hard HEAD^
```

Check What You've Changed

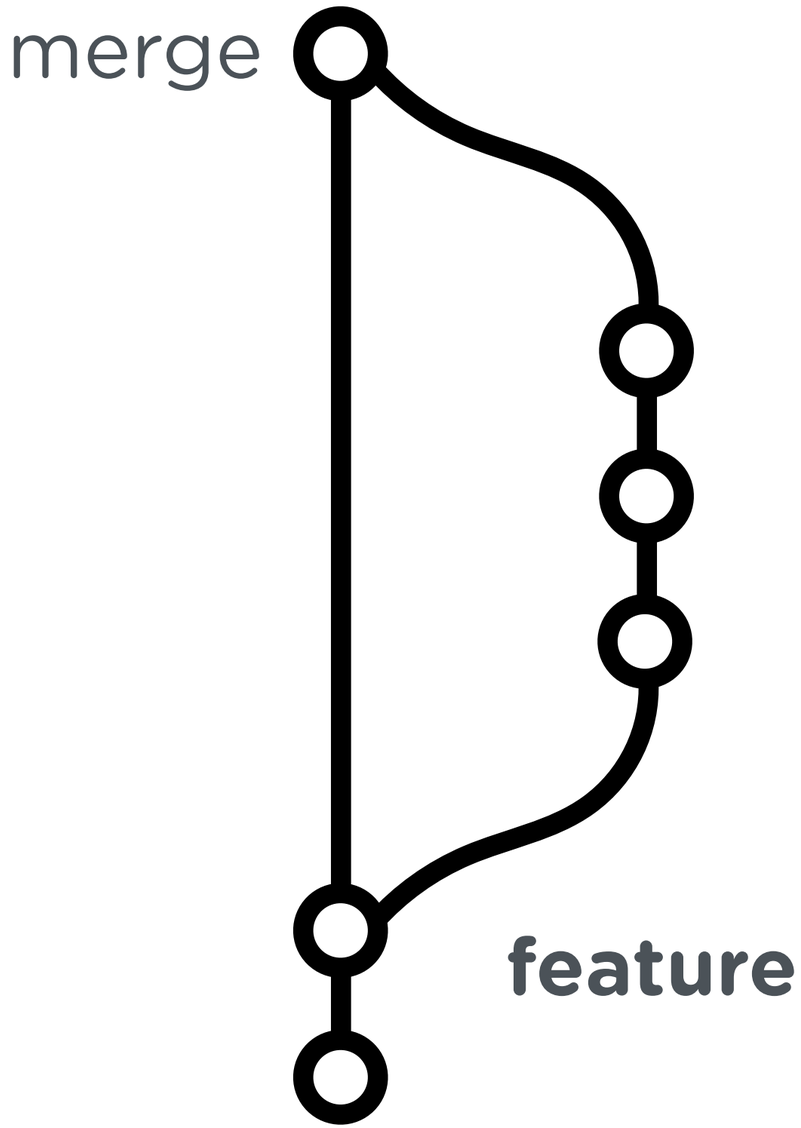
```
git diff
git diff -- <file/directory>
```

branch

Create a separate path for development and adding features

master

The main branch for a repository

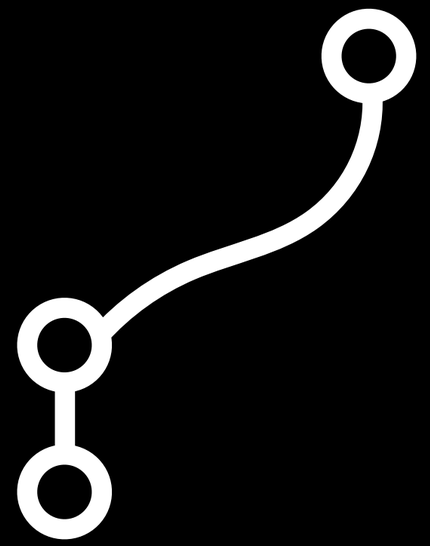


A Simple Workflow

1. Create and checkout a new working branch
2. Work in that branch. Making changes adding features
3. Once everything is working, merge changes back into master branch
4. Delete feature branch
5. Deploy master (push to web server)

Create and Checkout New Branch

```
git checkout -b <working_branch_name>
```

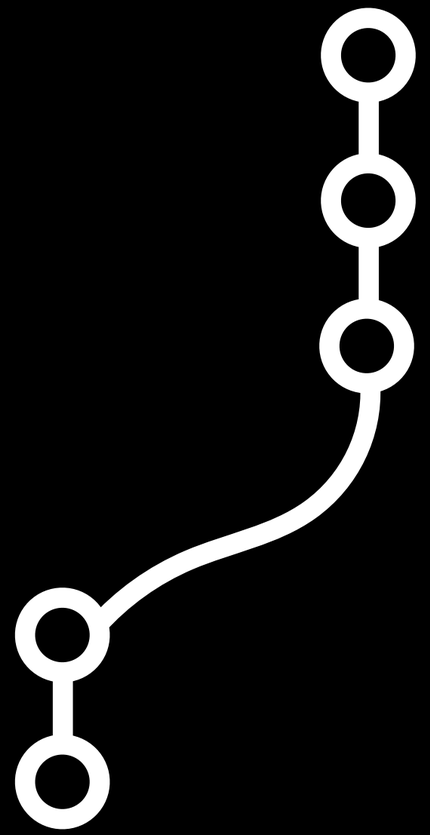


What Branches Do I Have?

```
git branch
```

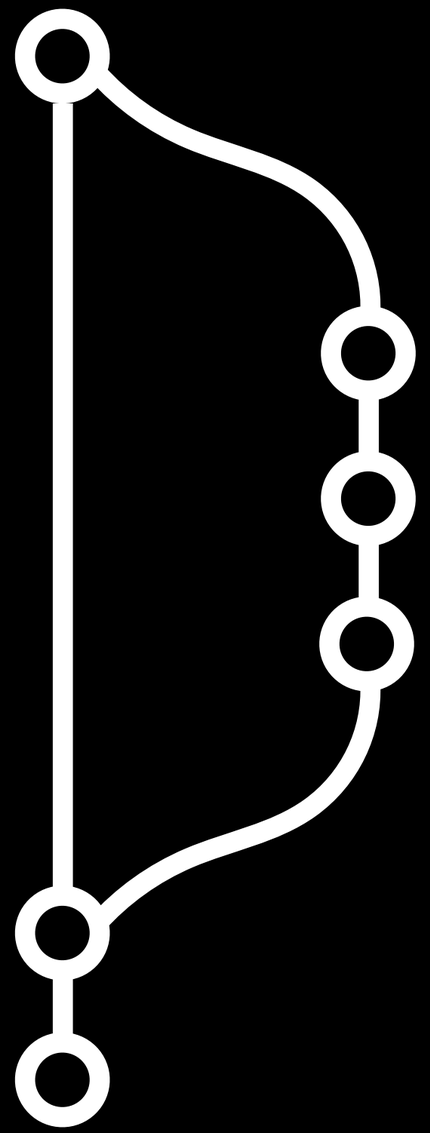
Return to Master

```
git checkout master
```



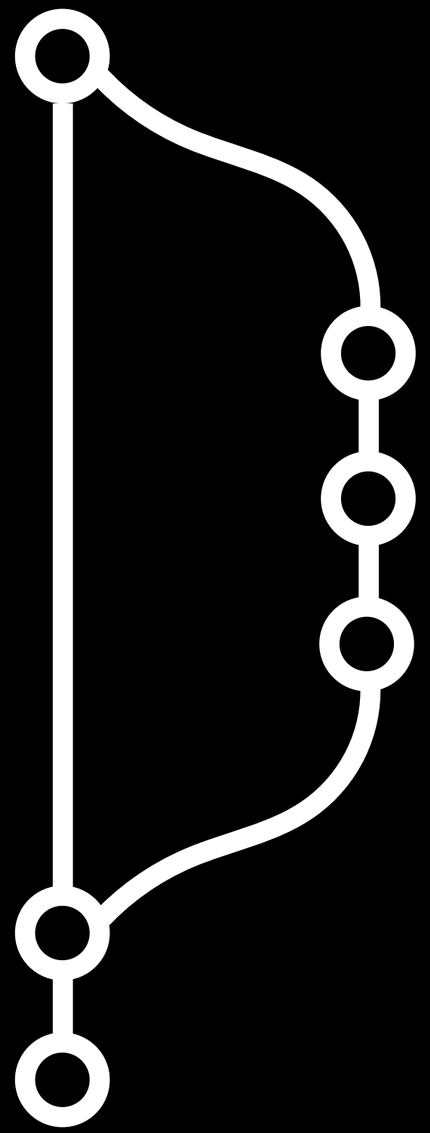
Merge Changes

```
git merge <working_branch_name>
```



Delete Old Branch

```
git -d <working_branch_name>
```



https://github.com/treehouse-dave/get-acquainted-with-git