You can find the Terminology handout at the bottom of the page. The file is named **Git Key Terms** and is in the Course Materials section.

Terminology

Version Control System / Source Code Manager

A **version control system** (abbreviated as **VCS**) is a tool that manages different versions of source code. A **source code manager** (abbreviated as **SCM**) is another name for a version control system.

Git is an SCM (and therefore a VCS!). The URL for the Git website is https://git-scm.com/ (see how it has "SCM" directly in its domain!).

Commit

Git thinks of its data like a set of snapshots of a mini filesystem. Every time you **commit** (save the state of your project in Git), it basically takes a picture of what all your files look like at that moment and stores a reference to that snapshot. You can think of it as a save point in a game - it saves your project's files and any information about them.

Everything you do in Git is to help you make commits, so a commit is *the* fundamental unit in Git.

Repository / repo

A **repository** is a directory which contains your project work, as well as a few files (hidden by default on Mac OS X) which are used to communicate with Git. Repositories can exist either locally on your computer or as a remote copy on another computer. A repository is made up of commits.

Working Directory

The **Working Directory** is the files that you see in your computer's file system. When you open your project files up on a code editor, you're working with files in the Working Directory.

This is in contrast to the files that have been saved (in commits!) in the repository.

When working with Git, the Working Directory is also different from the command line's concept of the *current working directory* which is the directory that your shell is "looking at" right now.

Checkout

A **checkout** is when content in the repository has been copied to the Working Directory.

Staging Area / Staging Index / Index

A file in the Git directory that stores information about what will go into your next commit. You can think of the **staging area** as a prep table where Git will take the next commit. Files on the Staging Index are poised to be added to the repository.

SHA

A **SHA** is basically an ID number for each commit. Here's what a commit's SHA might look like: `e2adf8ae3e2e4ed40add75cc44cf9d0a869afeb6`.

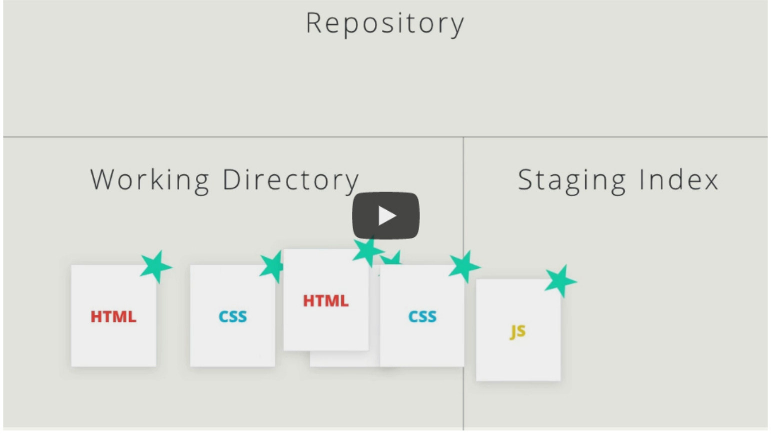
It is a 40-character string composed of characters (0–9 and a–f) and calculated based on the contents of a file or directory structure in Git. "SHA" is shorthand for "Secure Hash Algorithm". If you're interested in learning about hashes, check out our Intro to Computer Science course.

Branch

A **branch** is when a new line of development is created that diverges from the main line of development. This alternative line of development can continue without altering the main line development. This alternative line of development can continue without altering the main line.

Going back to the example of save point in a game, you can think of a branch as where you make a save point in your game and then decide to try out a risky move in the game. If the risky move doesn't pan out, then you can just go back to the save point. The key thing that makes branches incredibly powerful is that you can make save points on one branch, and then switch to a different branch and make save points there, too.

With this terminology in mind, let's take a high-level look at how we'll be using Git by looking at the typical workflow when working with version control.

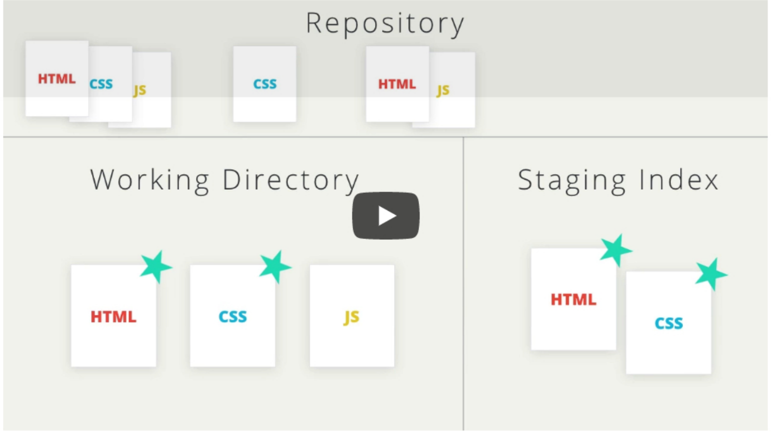


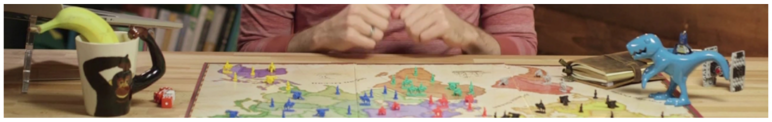
QUIZ QUESTION

The HTML file has HTML and CSS changes on the Staging Index and an additional HTML change in the Working Directory. From what you've learned so far with how committing works, what do you think will get committed if a commit were made right now?

- ○ only the HTML file changes on the Staging Index
- ○ only the CSS file changes on the Staging Index
- ✓ the HTML and CSS file changes on the Staging Index
- ○ the HTML file changes in *both* the Working Directory *and* the Staging Index

SUBMIT





Supporting Materials

Git Terms PDF

NEXT