

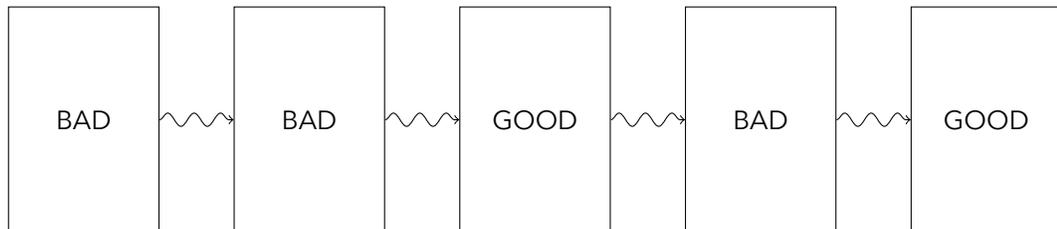
An Itty Bit of Git

WTF is Git?

In addition to editors, compilers and debuggers, probably the most important piece of software that a developer uses is a *version control system*. A version control system (VCS) allows programmers to keep track of their code as it evolves from day-to-day and allows groups of programmers to work together and combine their changes. *git* is a particular VCS. In this course, we are going to teach you to use some of the simpler features of *git*.

Software Development

Here's what the typical development cycle looks like:



Programming is complex enough that it is pretty common to break something while adding something new, so it would be nice to take a snapshot of everything when it works, so when it breaks you can see what changed or to just go back. We are going to show you how to use *git* to get this functionality.

Git terminology

repository The database of the entire history of a project. In *git*, this is stored inside the *working directory* in an invisible subdirectory. For a project a repository is either initialized by the user, in which case it is a completely empty history, or is *cloned* from an preexisting repository on a server somewhere. In this class, you will usually create your own repositories, but for a couple projects, you will be given a repository that you must clone.

working directory This is the subdirectory that contains your project. It will have an invisible *repository* within it.

change Whenever you alter a file in your *working directory* such that it is different from the one that is stored in the *repository*, this is called a *change*. Adding or removing a file is also a *change*. Changes are not saved until they have been *committed* to the repository.

commit A commit is a set of changes that are added atomically to the *repository*. Committing in git is a 2-phase process. First you add the changes to the *index*, an intermediate place, and then you *commit* the index to the repository, which makes a permanent record of the changes in the *log*.

log The complete history of the *commits* made to a repository.

clean A working directory is *clean* if there are no *changes*. That is, there are no files that are different from the ones in the most recent *commit* in the *repository*. In this class, all projects submitted must be in the form of clean repositories.

Git commands

These are the basic git commands that you will need for this class.

- `git init` - Converts a standard directory into a git repository. Note, that there will be no commits initially. You will need to use the commands below to add any existing files to the new repository.
- `git clone <url>` - Copies a repository from a server and creates a working directory on your machine.
- `git add .` - (Don't forget the dot) Adds any changes in the working directory to the index. They are not committed until you issue a `commit` command.
- `git status` - Shows the current state of the working directory, including any files that have changes. This will tell you if your working directory is clean. It will also give you commands in case you want to rollback any of your changes to the version stored in the repository.
- `git log` - Gives the complete history of the repository.
- `git commit -m ``Your Message Here``` - Commits all of the changes in the index to the repository. The messages are required by git. Makes sure that you have added all of your changes to the index before doing this. `git status` should help you figure out if this is the case.

Hints

Use this system to your advantage. For example, it is a great practice to make a commit every time that you have some code that works. For example, you've completed a function and tested it. This way, you have a record of a point in time where things were stable. If, later, your royally screw up, get lost, delete a file, etc., you can rollback to a stable point and try again.

There are many, many more commands for working in teams and for creating alternate histories, but none of those are needed to complete the assignments for this course. git is a powerful tool that is useful any time that you need to keep track of how a directory of information changes over time.