Udacity Git Lecture 1

Notebook: nikunibanka's notebook
Created: 05-01-2015 18:46 **Updated:** 06-01-2015 01:08

- windows fc command to compare 2 files.

Instructor Notes

Cloning a Repository

To clone a repository, run `git clone` followed by a space and the repository URL.

Asteroids URL

Use the following url to clone the Asteroids repository:
https://github.com/udacity/asteroids.git

Exiting `git log`

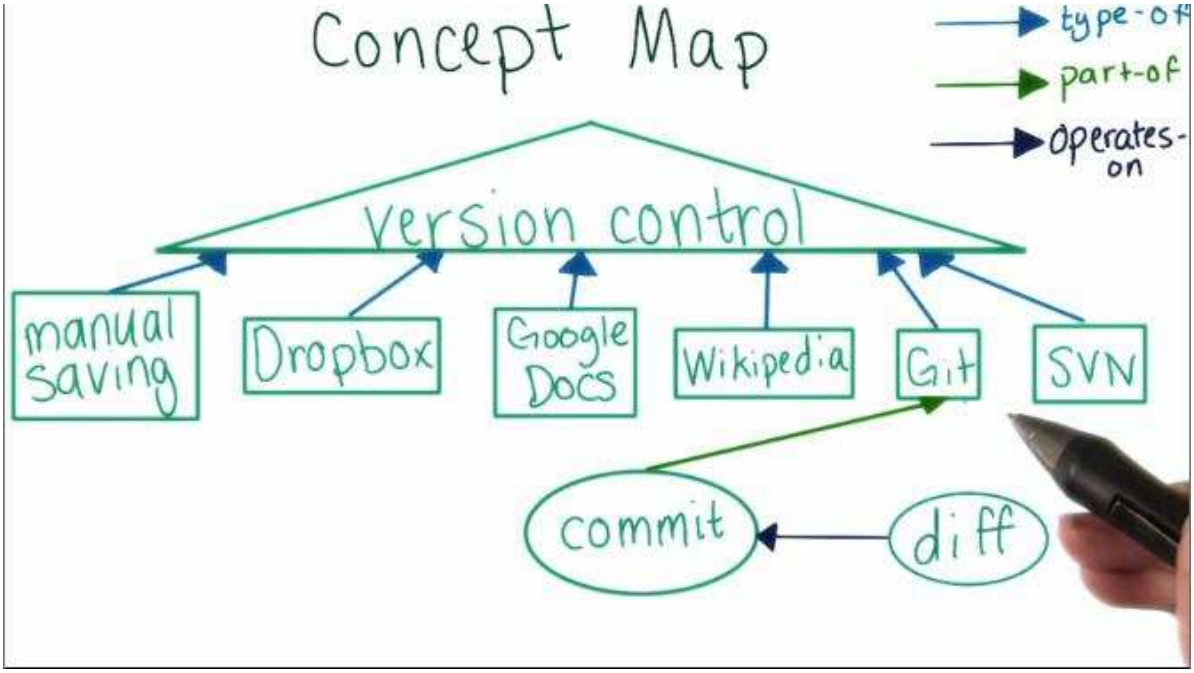
To stop viewing `git log` output, press `q` (which stands for quit).

Getting Colored Output

To get colored diff output, run `git config --global color.ui auto`

Using `git log` and `git diff`

As a reminder, running `git log` will show a list of the recent commits with information about them, including commit IDs. Running `git diff` followed by two commit IDs will compare the two versions of the code in those commits. If you need a refresher, you may want to rewatch this video.



- Git checkout will take the code back to a pervious version. It generates a warning though.
- Laptop@NIKUNJ ~/asteroids (master)
  $ git checkout b0678b161fcf74467ed3a63110557e3d6229cfa6
  Note: checking out 'b0678b161fcf74467ed3a63110557e3d6229cfa6'.

  You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by performing another checkout.

  If you want to create a new branch to retain commits you create, you may do so (now or later) by using -b with the checkout command again. Example:

  git checkout -b new_branch_name

  HEAD is now at b0678b1... Revert controls
- detached head state means that now we have changed the location of the head. so now if we make changes on this code and then commit then a new branch would be created.

Git Errors and Warnings Solution

Should not be doing an octopus

Octopus is a strategy Git uses to combine many different versions of code together. This message can appear if you try to use this strategy in an inappropriate situation.

You are in 'detached HEAD' state

HEAD is what Git calls the commit you are currently on. You can "detach" the HEAD by switching to a previous commit, which we'll see in the next video. Despite what it sounds like, it's actually not a bad thing to detach the HEAD. Git just warns you so that you'll realize you're doing it.

- After typing git checkout on a previous version. And then typing git log will not show the next versions in the log.
- So how to go ahead and we dont know the commit id?
- Easy now. Provided in inst. notes.

- cd ~
  Takes to the home directory.

```
$ mv Downloads/git-prompt.sh.txt git-prompt.sh
cbuckey@Udacity-PC ~
$
```

- mv command moves the file to the current directory where the git is right now and renames it.

```
cbuckey@Udacity-PC ~
$ git config --global core.editor "sublime -n -w"
```

- This changes the default text editor to sublime text.
- Sublime must be replaced with the whole path to the editor's exe.
- -n means sublime will open in a new window.
- -w means that git will wait for you to close sublime before trying to continue.

Instructor Notes

Changing background color

If you prefer the background color of Git Bash to be something other than black, you can change it in the "Defaults" menu under the "Colors" tab. If you like the background color as-is, you don't need to make any changes.

Downloading necessary files

- Save this file in your home directory with the name `git-completion.bash`.
- Save this file in your home directory with the name `git-prompt.sh`.
- Save `bash_profile_course` from the Downloadables section in your home directory with the name `.bash_profile`. (If you're curious to learn more about how bash prompts work, see this page.)

Making Git configurations

Run the following Git configuration commands. The first one will need to be modified if you are using a text editor other than Sublime, or if Sublime is installed in another location for you. See this page for the correct command for a couple of other popular text editors. For any other editor, you'll need to enter the command you use to launch that editor from Git Bash.

```
git config --global core.editor "'C:/Program Files/Sublime Text2/sublime_text.exe' -n -w"
git config --global push.default upstream
git config --global merge.conflictstyle diff3
```

Make sure you can start your editor from Git Bash

If you use Sublime, you can do this by adding the following line to your `.bash_profile`:

```
alias subl="C:/Program\ Files/Sublime\ Text\ 2/sublime_text.exe"
```

Restart Git Bash

You'll need to close and re-open Git Bash before all your changes take effect.
