Git Essentials 1

What is Git?

1) In this video the author explains several version control systems that are non source code related. ***Name 2.***

- MS Word
- Adobe Photoshop
- Wiki/Wikipedia

2) What does it mean when we say a source code versioning system is ***concurrent****?*

It means that there is a place for us to store our code (a "repository" or "repo") and that place can be uploaded to a remote server where multiple developers can work on it at the same time.

3) The author identifies several advantages of a distributed version control system. ***Name 2****:*

- It's faster
- Network access isn't required
- Able to work independently

4) The author identifies several types of files that are ***not*** *as useful in storing changes in GIT. What types of files are these? What does GIT NOT DO with these files?*

The types of files that are not as useful when storing changes in GIT are non-text files such as fonts, images/videos, and music. If changes are made to these types of files, it will track the fact that changes were made to the file, but it will not show the differences in file.

GITting Started

5) Does it matter where I can the GIT command **init**? Why?

I am not 100% certain what this question is asking. I think there is a typo, and a word is missing. I'm taking a stab at answering it, anyhow. Based on the content of the video, I am guessing that the question is asking "Does it matter where I can do/run the git command init," and the answer to this is yes. It does matter where the init command is run. From the command line, the init command must be run from the root of the project folder. This tells GIT to make this project a repo, and track all of the files and changes made to the directory.

6) How would we remove the repo from being tracked by Git?

To remove the repo from being tracked by Git, simply delete the .git directory.

7) What git command does the author use to add all of the files in the folder to the current project?

The git command to add all files in the folder to the project is `git add .`

8) [Critical Thinking] Why are commit messages best written in the **present** tense?

Commit messages are best written in present tense because the commit messages are labelling what the commit does vs. what the creator was doing.

Git Concepts and Architecture

9) [Critical Thinking] What is the advantage of a three tree architecture over a two tree architecture?

The advantage of Git's three-tree architecture is that you can check out a bunch of files, for example, seven. You can then work on those files and make the changes that you need to make to those files, but then you don't have to commit all of them right away. If you are only ready to commit four of the seven files, you can add those four files to the staging index and when you are all ready, commit them as "one changed set."

10) [Critical Thinking] What is the difference between the **git add** command and the **git commit** command?

The `git add` command pushes a set of changes to the staging index, whereas the `git commit` command commits them to the repository.

11) What is the purpose of the **checksum**?

The purpose of the checksum is to discover errors that were potentially introduced while being stored or transmitted.

Making changes to Files

12) What command does the author use to see whether or not there are changes that need to be committed?

The command that you use to check whether there are changes that need to be committed is `git status`

13) When using the **git add** command, is there any difference between adding a new file or editing an existing file?

No, when using `git add` there is no difference between adding a new file or editing an existing one.

14) To view the **differences** between files named **first.html** and **second.html** as they exist in the root of the project/local repo folder, what exact git command and syntax would you use?

git diff – the minus signs indicate the old version and the plus signs indicate the new version.

Using Git with a real project

15) What command does the author use to **stage** every file in the **explore_california** folder to the repository?

To stage every file, use the ***git add .*** command.

16) What command does the author use to **commit** every file in the **explore_california** folder and add a comment?

To commit those files, use the ***git commit -m "initial commit"*** command.

17) What command does the author use to add and commit the files at one time?

git commit -a is saying to add and commit all the files in one single step.

18) [Critical Thinking] The author says there is a caveat for using the shortcut you identified in the previous question. In which 2 circumstances should you not use the shortcut?

You don't want to use the above command (***git commit -a***) if there are files in the directory that you do not want to commit, and also if there are new or deleted files because it does not work for those files.

19) What is the git command you would use to rename **test.htm** to **test.php**, if both are in the root of the repo?

git mv test.htm test.php

20) What command does the author use to add everything in the tours folder to the staging area?

git add tours/

you can also use ***git add tours/****, but * isn't required. Either way will work.