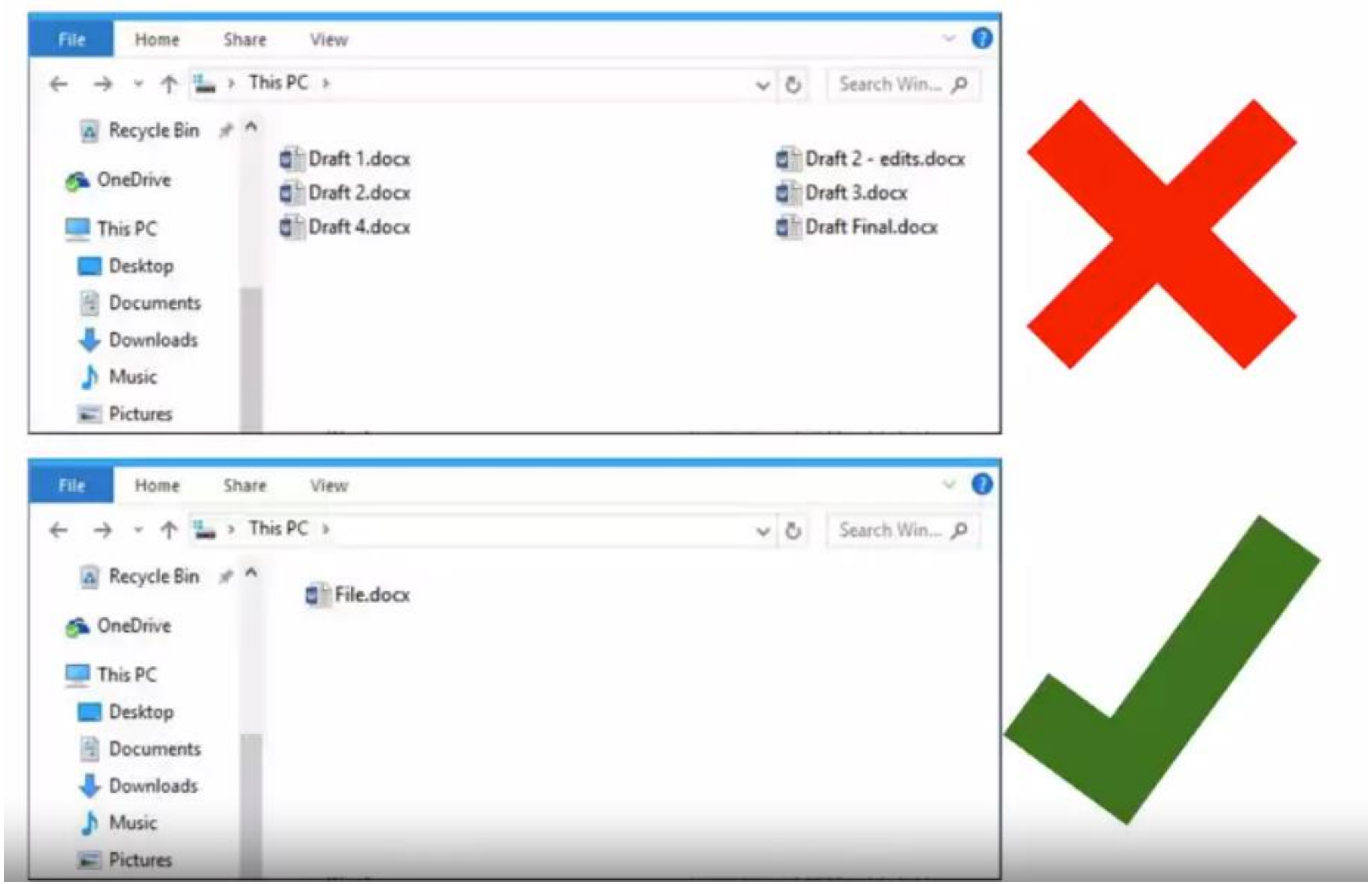
Version Control and Github

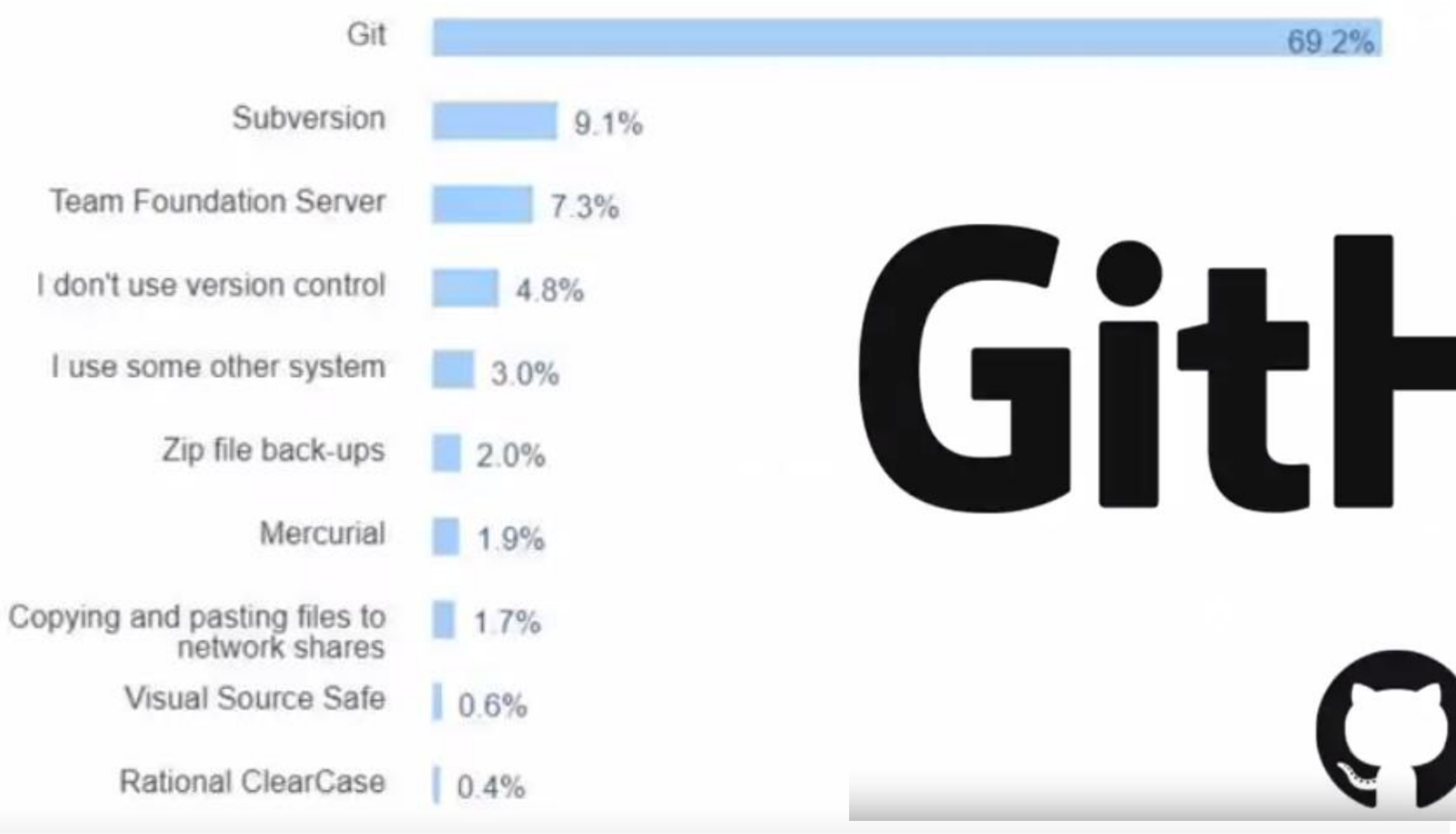
Eunbyeol Lee

2019.11.29

Version control



Git


Git
69.2%
Subversion
9.1%
Team Foundation Server
7.3%
I don't use version control
4.8%
I use some other system
3.0%
Zip file back-ups
2.0%
Mercurial
1.9%
Copying and pasting files to network shares
1.7%
Visual Source Safe
0.6%
Rational ClearCase
0.4%

GitHub

Words involved in working with Git

Repository, AKA Repo

✓A repository is equivalent to the projects folder or directory.

✓All of your version controlled files and the recorded changes are located in a repository.

✓Private or public

✓To commit is to save your edits and the changes made.

Words involved in working with Git

✓A commit is like a snapshot of your files.

✓Git compares the previous version of all of your files in the repo to the current version and identifies those that have changed since then.

✓Those that have changed, it compares the files, loads the changes and uploads the new version of your file.

✓We'll touch on this in the next section, but when you commit a file, typically you accompany that file change with a little note about what you changed and why.

✓ If you find a mistake, you will revert your files to a previous commit.

✓If you want to see what has changed in a file over time, you compare the commits and look at the messages to see why and who.

Words involved in working with Git

- ✓To push is to update the repository with your edits.
- ✓Since Git involves making changes locally, you need to be able to share your changes with the common online repository.
- ✓Pushing is sending those committed changes to that repository so now everybody has access to your edits.

Words involved in working with Git

✓Pulling is updating your local version of the repository to the current version since others may have edited in the meanwhile.

✓Because the shared repository is hosted online in any of your collaborators or even yourself on a different computer could it made changes to the files and then push them to the shared repository.

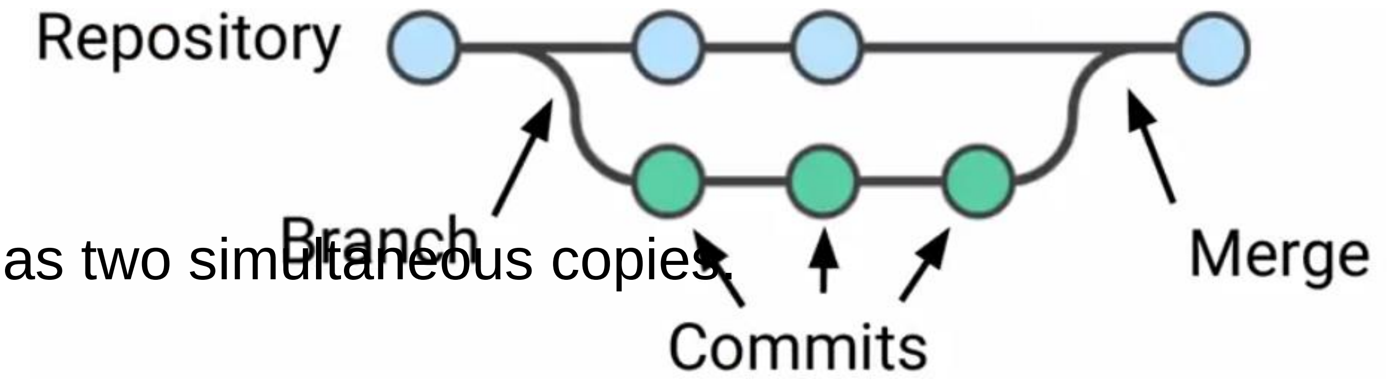
✓You are behind the times, the files you have locally on your computer may be outdated.

✓So, you pull to check if you were up to date with the main repository.

Staging

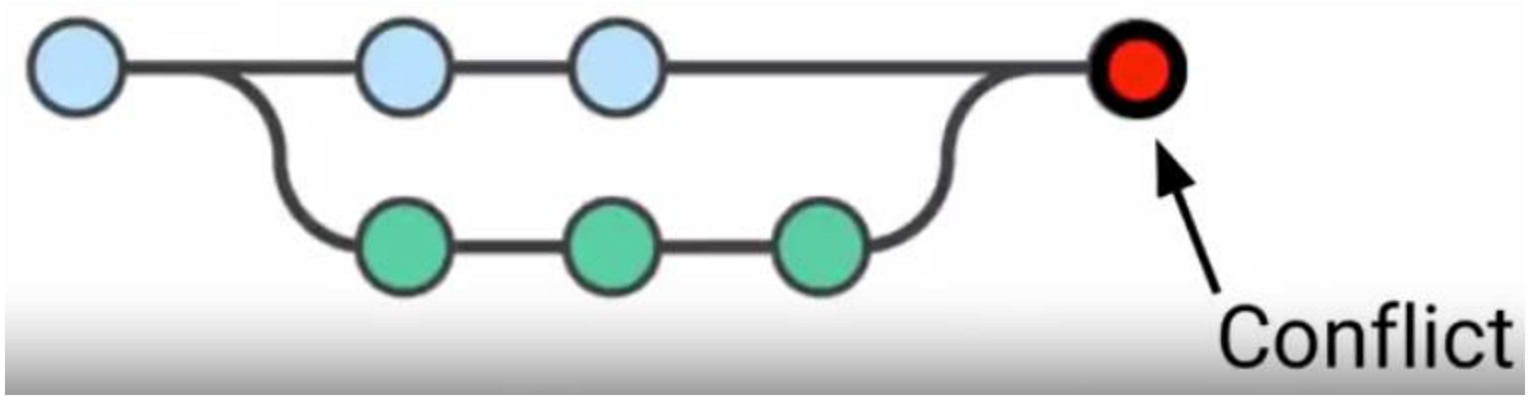
- ✓Staging is the act of preparing a file for a commit.
- ✓For example, if since your last commit you have edited three files for completely different reasons, you don't want to commit all of the changes in one go, your message on why you are making the commit in what has changed will be complicated since three files have been changed for different reasons.
- ✓So instead, you can stage just one of the files and prepare it for committing.
- ✓Once you've committed that file, you can stage the second file and commit it and so on. Staging allows you to separate out file changes into separate commits, very helpful.

Words involved in working with Git



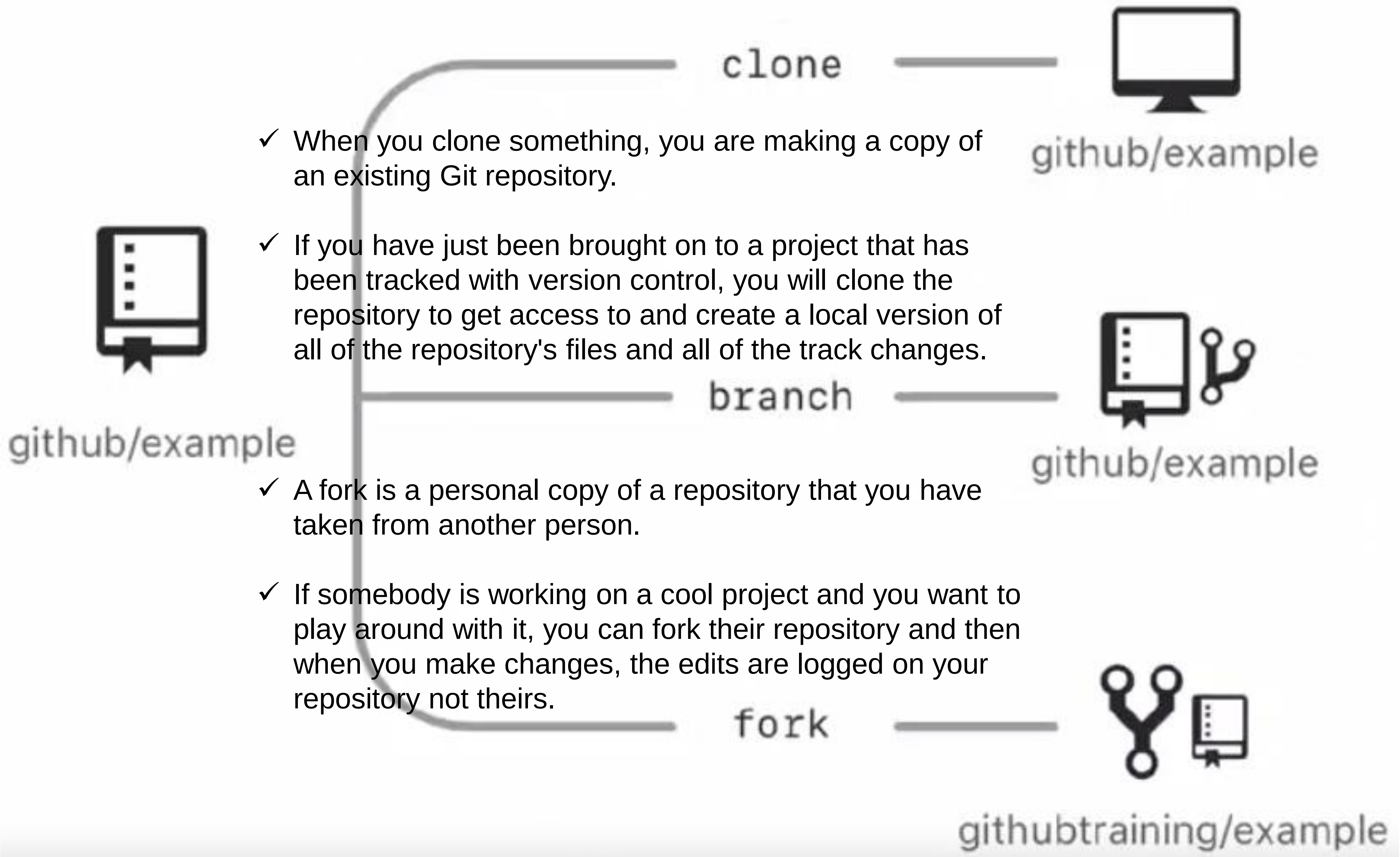
- ✓A branch is when the same file has two simultaneous copies.
- ✓When you were working locally in editing a file, you have created a branch where your edits are not shared with the main repository yet.
- ✓The version that everybody has access to on the repository and your local edited version of the file.
- ✓Until you push your changes and merge them back into the main repository, you are working on a branch.
- ✓Following a branch point, the version history splits into two and tracks the independent changes made to both the original file in the repository that others may be editing and tracking your changes on your branch and then merges the files together.

Words involved in working with Git



- ✓Merging is when independent edits of the same file are incorporated into a single unified file.
- ✓If both people made an edit to the same sentence that precludes one of the edit from being possible, we have a problem.
- ✓Git recognizes this disparity, conflict and asks for user assistance in picking which edit to keep.
- ✓So, a conflict is when multiple people make changes to the same file and Git is unable to merge the edits.

- ✓When you clone something, you are making a copy of an existing Git repository.
- ✓If you have just been brought on to a project that has been tracked with version control, you will clone the repository to get access to and create a local version of all of the repository's files and all of the track changes.
- ✓A fork is a personal copy of a repository that you have taken from another person.
- ✓If somebody is working on a cool project and you want to play around with it, you can fork their repository and then when you make changes, the edits are logged on your repository not theirs.



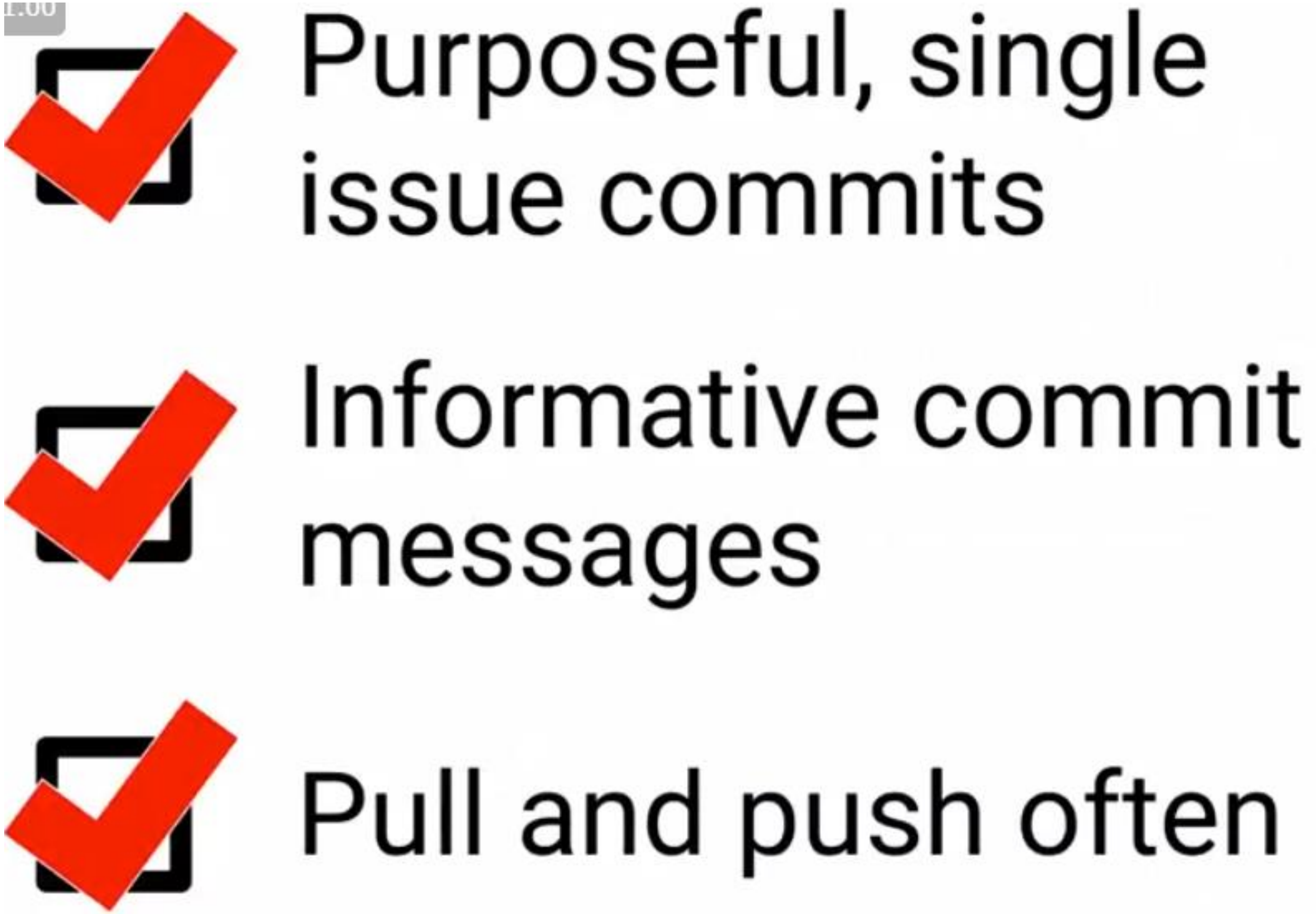
- ✓ When you clone something, you are making a copy of an existing Git repository.

- ✓ If you have just been brought on to a project that has been tracked with version control, you will clone the repository to get access to and create a local version of all of the repository's files and all of the track changes.

- ✓ A fork is a personal copy of a repository that you have taken from another person.

- ✓ If somebody is working on a cool project and you want to play around with it, you can fork their repository and then when you make changes, the edits are logged on your repository not theirs.

Keep in mind to establish good habits



- ☑ Purposeful, single issue commits
- ☑ Informative commit messages
- ☑ Pull and push often