Collaboration | Coursera

Collaboration

Collaboration

When you work on a software project with hundreds of source files and dozens of developers, how do you keep everyone up to date on the latest source? E-mailing files back and forth would be a nightmare. Even if you only have two people working on a project, managing changes between the developers is a first-order concern.

Revision control systems such as Git support notions of *pushing* —sending your changes to another repository (located on another computer)—and *pulling* changes—getting the most recent version from another repository. If you try to push your own changes, but are not up to date with the other repository, Git will require you to pull first, so that you cannot unexpectedly overwrite someone else's changes. When you pull, Git will take care to make sure your changes are integrated with the remote changes.

If you pull and have not made any changes to your own repository (since the last time you pushed to that repository), Git will simply update you to the most recent version of the repository. If you have made changes, Git will try to merge your changes with the remote changes. If you have changed different files, or different parts of the same file, Git will typically handle the merge automatically. However, if Git cannot merge automatically, it will indicate what problems need your attention, and require you to fix them before you push your changes back.

✓

Completed