

Version Control with Git

(index.html)

Automated Version Control

✳ Learning Objectives

- Understand the benefits of an automated version control system.
- Understand the basics of how Git works.

We'll start by exploring how version control can be used to keep track of what one person did and when. Even if you aren't collaborating with other people, automated version control is much better than this situation:

"FINAL".doc



FINAL.doc!



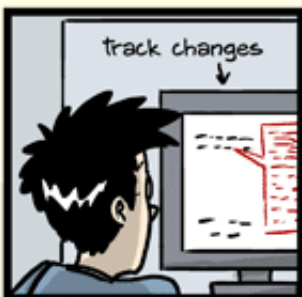
FINAL_rev.2.doc



FINAL_rev.6.COMMENTS.doc



FINAL_rev.8.comments5.
CORRECTIONS.doc



FINAL_rev.18.comments7.
corrections9.MORE.30.doc



FINAL_rev.22.comments49.
corrections.10.#@\$%WHYDID
ICOMETOGRADSCHOOL????.doc



JORGE CHAM © 2012

WWW.PHDCOMICS.COM

(<http://www.phdcomics.com>)

"Piled Higher and Deeper" by Jorge Cham, <http://www.phdcomics.com>

We've all been in this situation before: it seems ridiculous to have multiple nearly-identical versions of the same document. Some word processors let us deal with this a little better, such as Microsoft Word's "Track Changes" or Google Docs' version history.

Version control systems start with a base version of the document and then save just the changes you made at each step of the way. You can think of it as a tape: if you rewind the tape and start at the base document, then you can play back each change and end up with your latest version.



Figure: Changes are saved sequentially

Once you think of changes as separate from the document itself, you can then think about “playing back” different sets of changes onto the base document and getting different versions of the document. For example, two users can make independent sets of changes based on the same document.

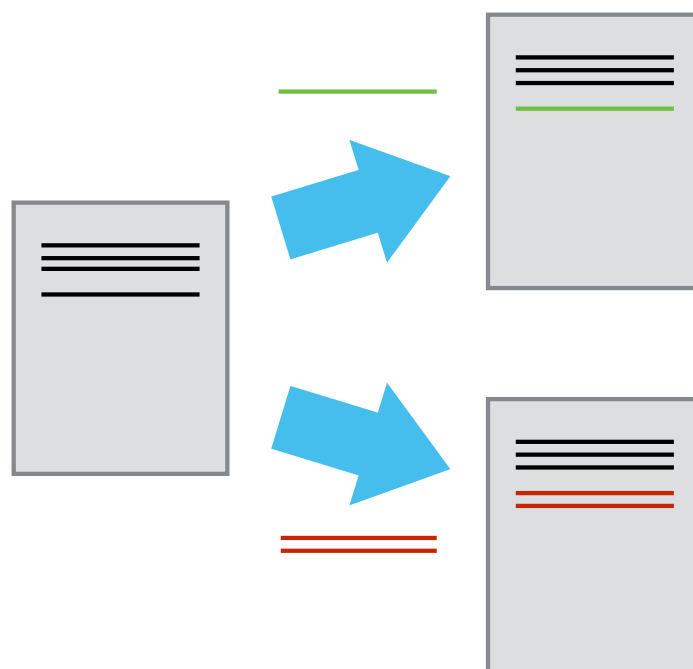


Figure: Different versions can be saved

If there aren't conflicts, you can even try to play two sets of changes onto the same base document.

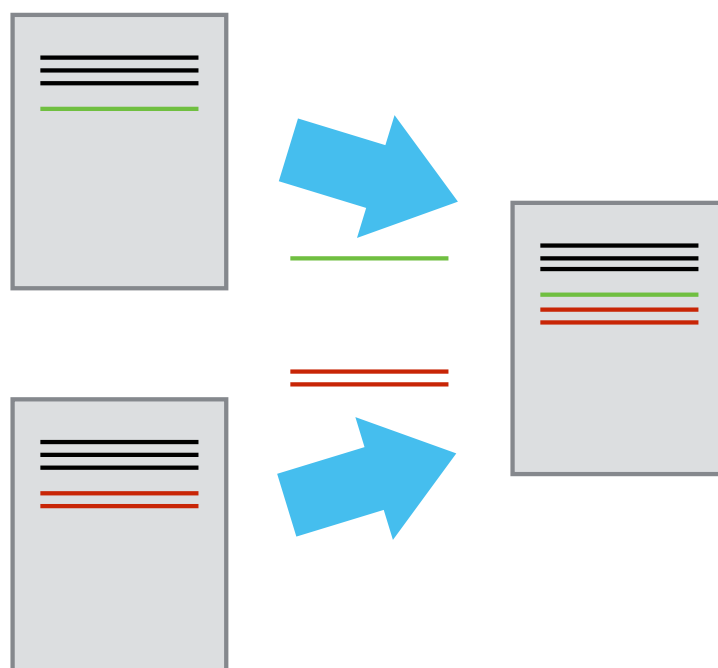


Figure: Multiple versions can be merged

A version control system is a tool that keeps track of these changes for us and helps us version and merge our files. It allows you to decide which changes make up the next version, called a [commit](#) ([reference.html#commit](#)), and keeps useful metadata about them. The complete history of commits for a particular project and their metadata make up a [repository](#) ([reference.html#repository](#)). Repositories can be kept in sync across different computers facilitating collaboration among different people.

✈ The long history of version control systems

Automated version control systems are nothing new. Tools like RCS, CVS, or Subversion are considered now legacy systems, offering more limited capabilities than modern tools, such as Git and [Mercurial](#) (<http://swcarpentry.github.io/hg-novice/>). In particular, the latter are distributed, meaning that they don't need a centralized server to host the repository.

Software Carpentry (<http://software-carpentry.org>)

Source (<https://github.com/swcarpentry/git-novice>)

Contact (<mailto:admin@software-carpentry.org>)

License ([LICENSE.html](#))