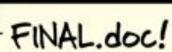
Git and GitHub

Versioning

"FINAL".doc









Versioning







FINAL_rev.6.COMMENTS.doc

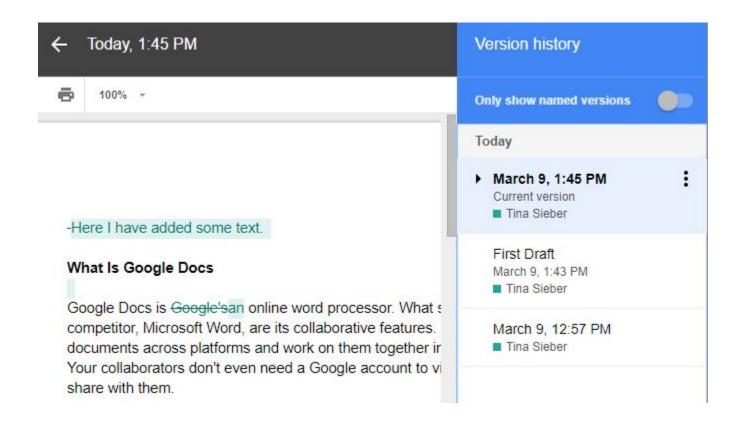
FINAL_rev.8.comments5.

Versioning



Git vs Google Docs

	saves current state	keeps track of
Git	on request	directory tree
Google Docs	automatically	single file



Git: upsides

- collaboration
 - sharing
 - peer review
- backup
 - can go back to any version
 - simple and secure way to distribute to other machines

Git: downsides

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL. COOL. HOU DO WEUSE IT? NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOUNLOAD A FRESH COPY.

Local workflow

- initialize a repository
 - o git init

Working copy	Staging area	Commits
my-file.txt data .gitignore		

Local workflow

- initialize a repository
 - o git init
- add files to staging area (where you prepare the next commit)
 - git add my-file.txt

Working copy	Staging area	Commits
my-file.txt data .gitignore	my-file.txt	

Local workflow

- initialize a repository
 - o git init
- add files to staging area (where you prepare the next commit)
 - git add my-file.txt
- create a commit
 - o git commit -m 'change this and that'

Working copy	Staging area	Commits
my-file.txt data .gitignore		my-file.txt 3c4a change this

Diff

A shorthand for "differences". Usually between working copy and the last commit.

'-' means line removed

'+' means line added

A sample from course materials repo:

```
``pull`` will do a merge for you
- ``git checkout --theirs``
+ ``git checkout --theirs {file path}``
  in conflict, choose the version from the remote branch
```

Branches

- branches help you manage multiple versions of your code
- important for programmers, but we can't ignore it completely
- we'll be using one local branch called master
- on a remote, there is a twin branch, origin/master

GitHub

"Social part of programming"



Pushing to a remote [server]

- add a remote
 - o git remote add origin {where}
- first push also create a remote branch from the local one
 - git push -u origin local-branch
- push upload commits
 - git push



Pulling from a remote

- download commits and merge into working copy
 - o git pull

Pulling from a remote

- download commits and merge into working copy
 - o git pull
- simple as that, unless there is a merge conflict



Resolving conflicts - binary files

- check what's conflicted
 - o git status
- binary files are simple, choose either the incoming (theirs) or ours
 - git checkout --ours binary-file.jpg
 - git add binary-file.jpg
- commit the resolution, without any message
 - git commit
 - recently available: git merge --continue

Resolving conflicts - text files

- check what's conflicted
 - o git status
- text files need to be edited
 - o nano my-file.txt
 - git add my-file.txt
- commit the resolution, without any message
 - git commit
 - recently available: git merge --continue

Resolving conflicts

In text files, if both branches modify the same line, a conflict has to be resolved by deleting all the unwanted text (also the marks).