CS100: Software Tools & Technologies Lab I

Introduction to Version Control using GIT

Vishwesh Jatala

Assistant Professor

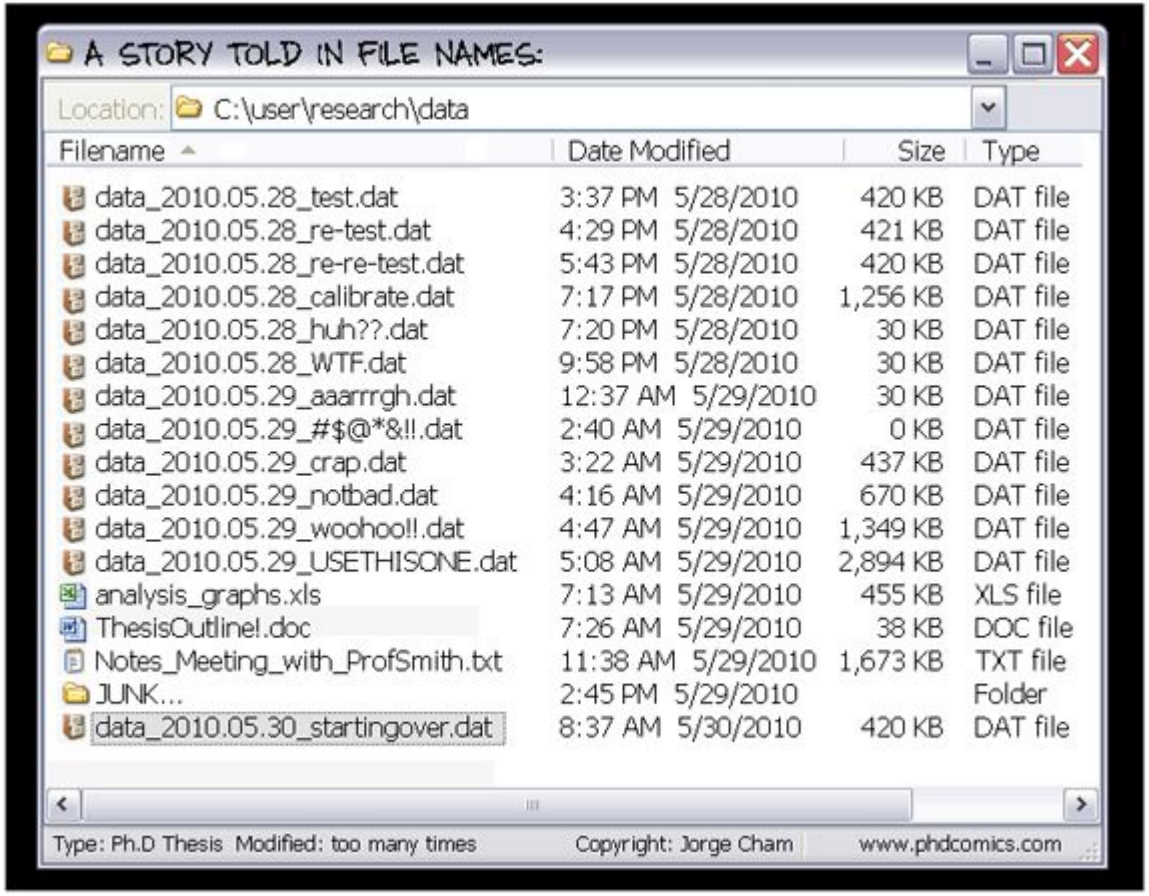
Department of EECS

Indian Institute of Technology Bhilai

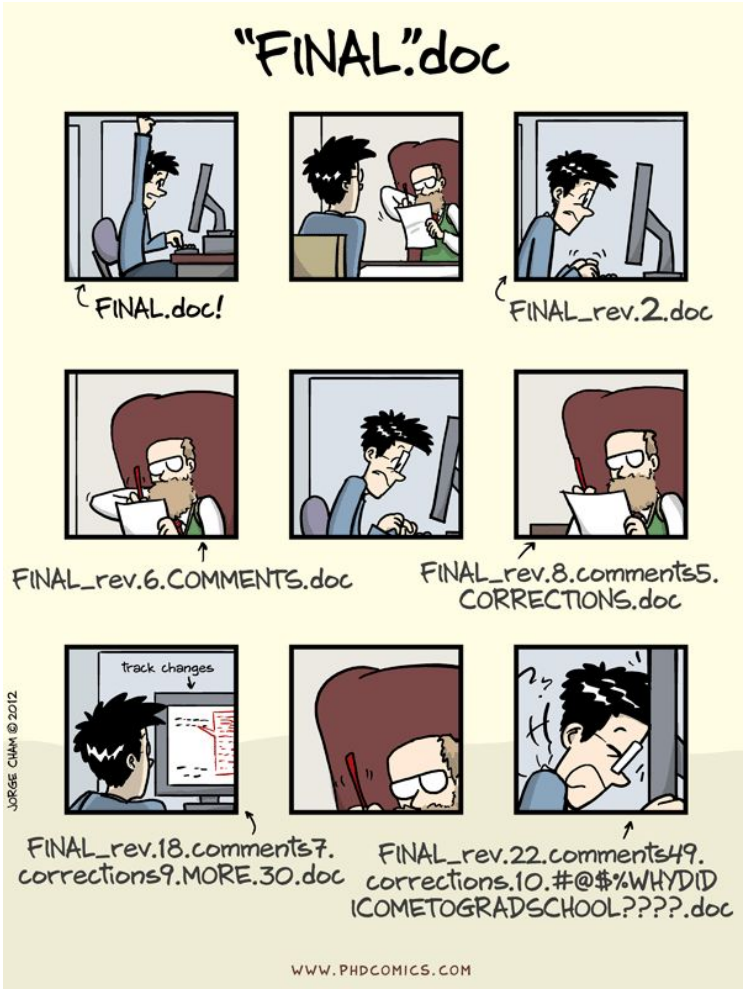
vishwesh@iitbhilai.ac.in

2022-23 W

Why version control?

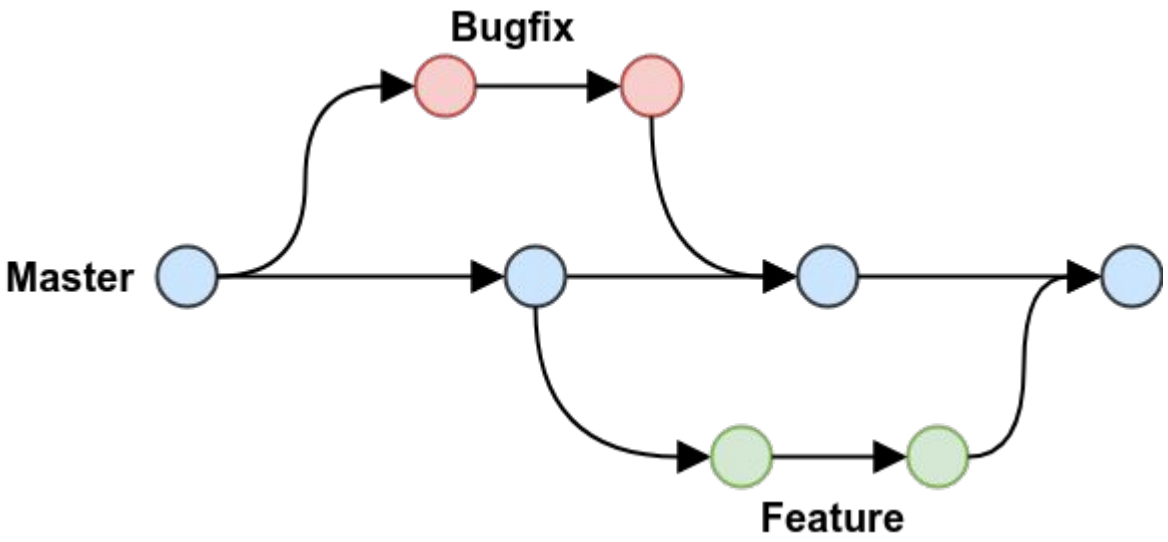


Why version control?



Why version control?

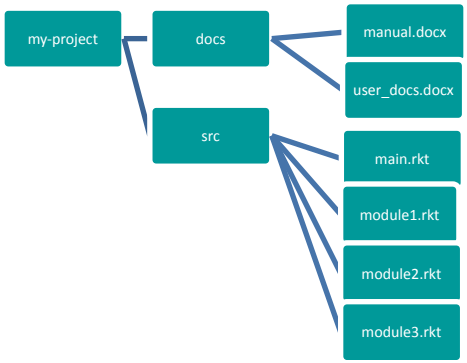
Why version control?



Git version-control system

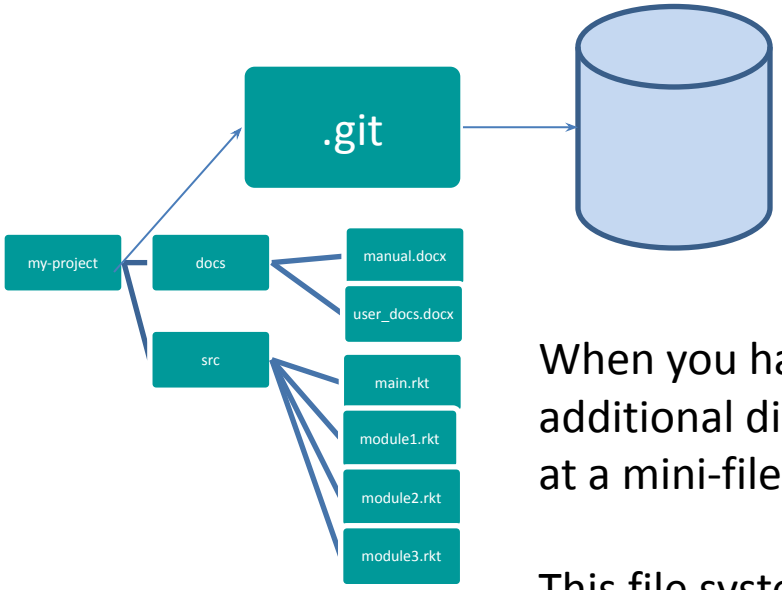
- You keep your files in a *repository* on your local machine.
- You synchronize your repository with a repository on a server.
- If you move from one machine to another, you can pick up the changes by synchronizing with the server.
- If your partner uploads some changes to your files, you can pick those up by synchronizing with the server.

Your files



Here are your files, sitting in a directory called my-project

Your files in your git repository

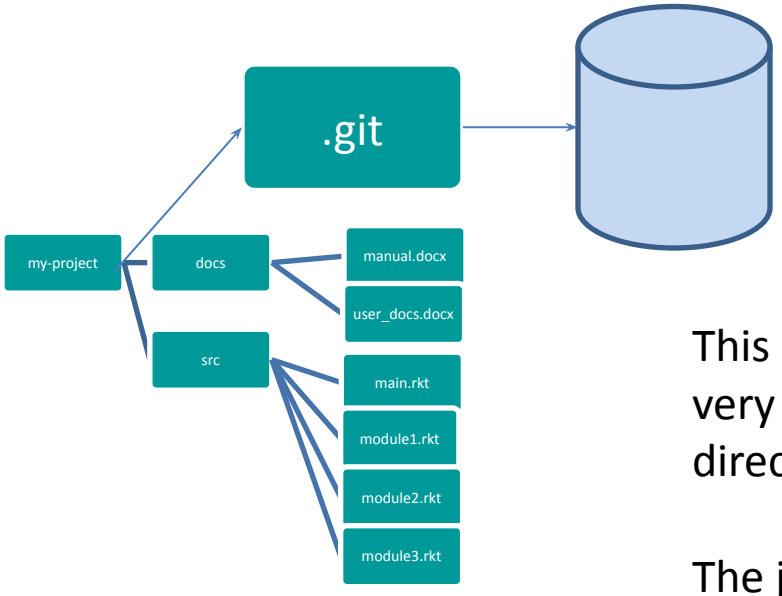


When you have a git repository, you have an additional directory called .git, which points at a mini-filesystem.

This file system keeps all your data, plus the bells and whistles that git needs to do its job.

All this sits on your local machine.

The git client



This mini-filesystem is highly optimized and very complicated. Don't try to read it directly.

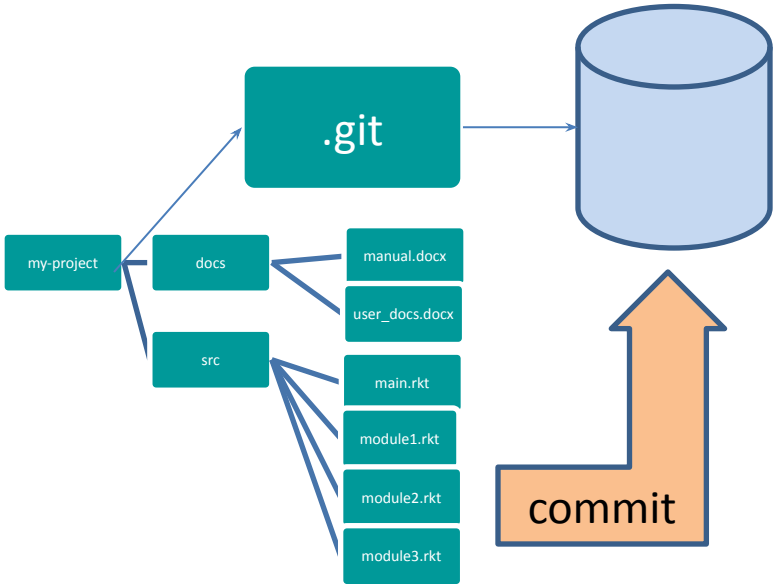
The job of the git client (either Github for Windows, Github for Mac, or a suite of command-line utilities) is to manage this for you.

Your workflow

You edit your local files directly.

- You can edit, add files, delete files, etc., using whatever tools you like.
- This doesn't change the mini-filesystem, so now your mini-fs is behind.

A Commit



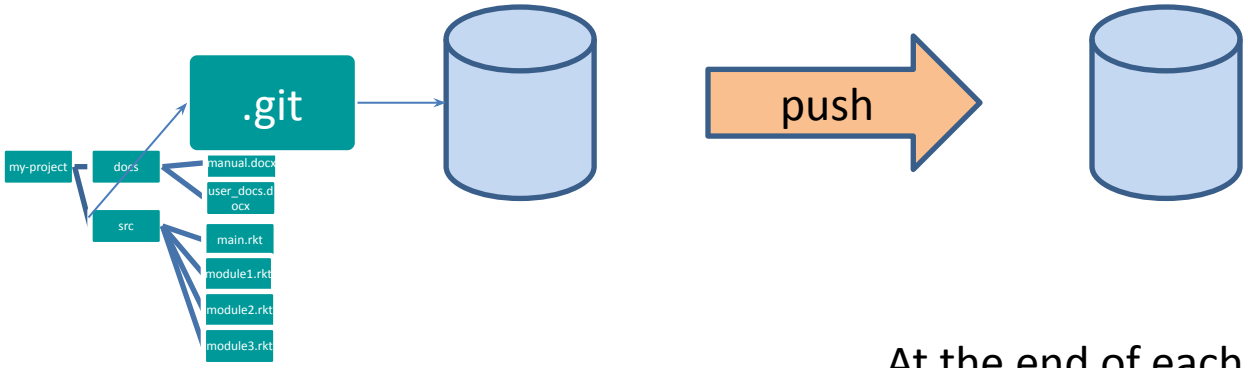
When you do a “commit”, you record all your local changes into the mini-fs.

The mini-fs is “append-only”. Nothing is ever over-written there, so everything you ever commit can be recovered.

Synchronizing with the server (1)

your local machine

a server, somewhere on the internet, eg. github.com

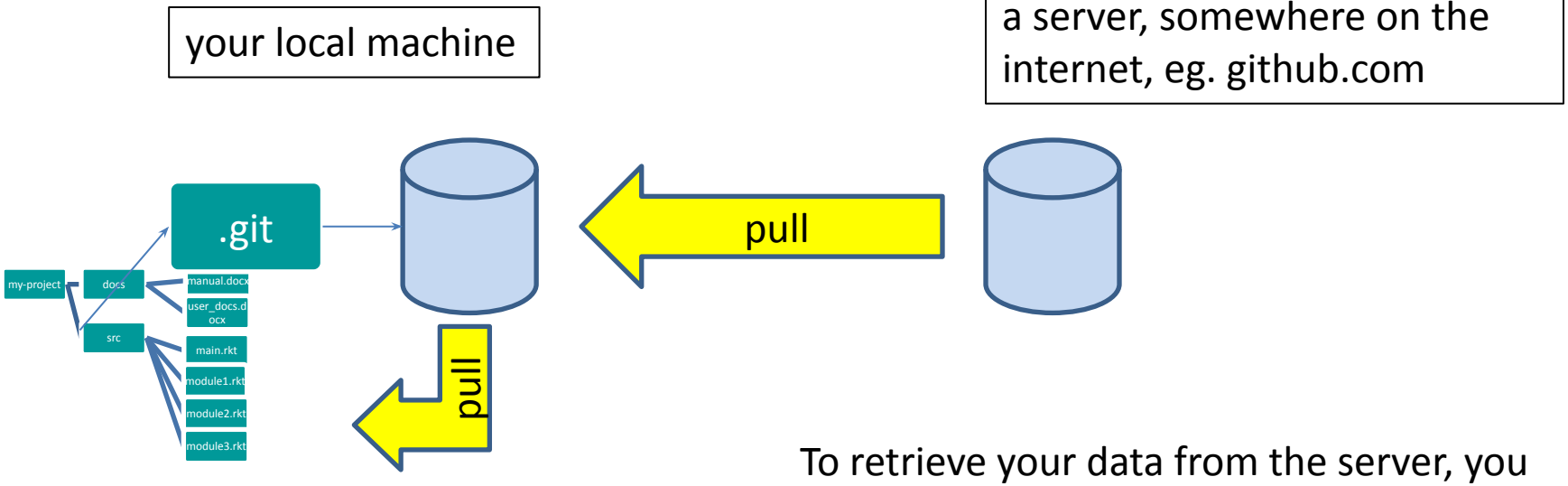


At the end of each work session, you need to save your changes on the server. This is called a "push".

Now all your data is backed up.

- You can retrieve it, on your machine or some other machine.
- We can retrieve it

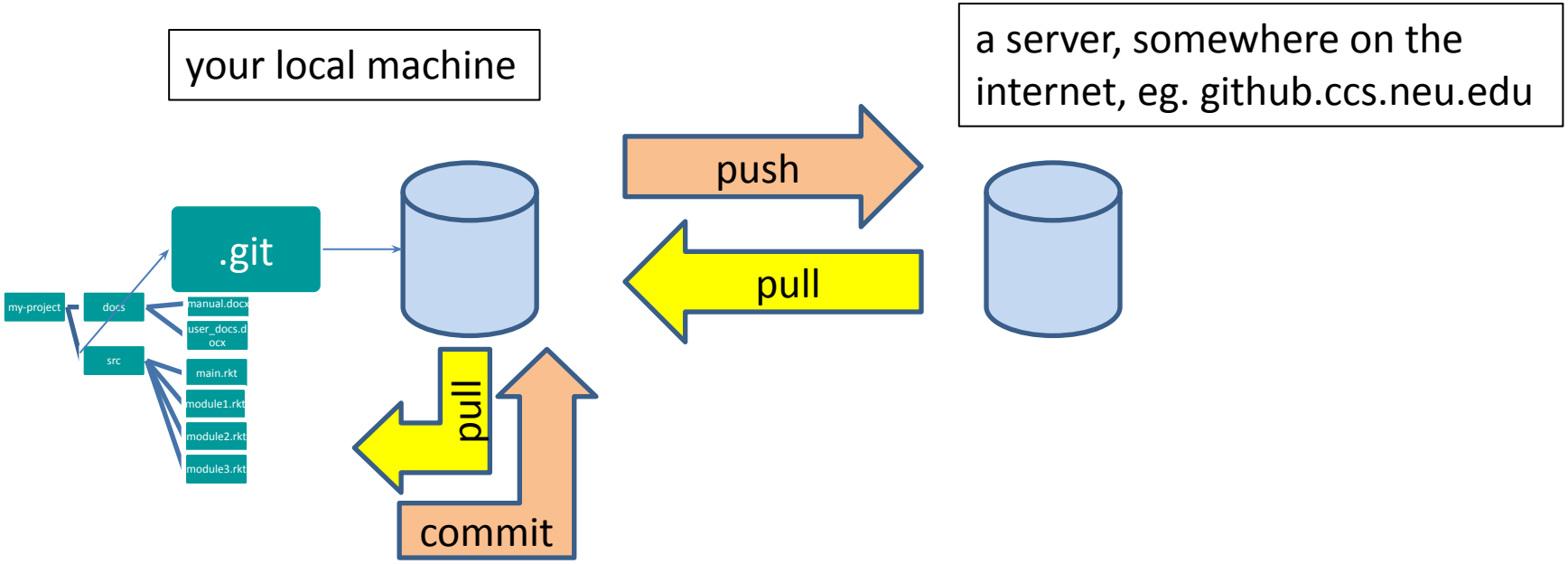
Synchronizing with the server (2)



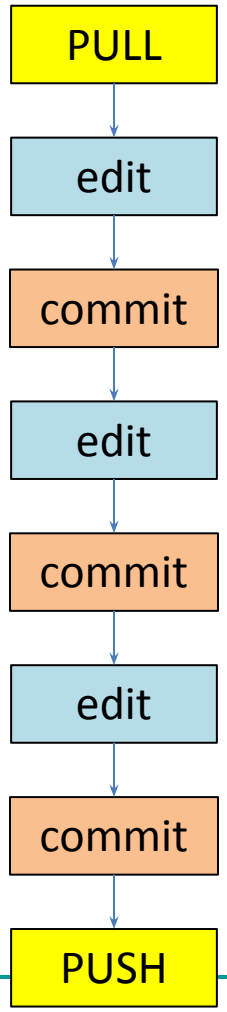
To retrieve your data from the server, you do a "pull". A "pull" takes the data from the server and puts it both in your local mini-fs and in your ordinary files.

If your local file has changed, git will merge the changes if possible. If it can't figure out how to the merge, you will get an error message.

The whole picture



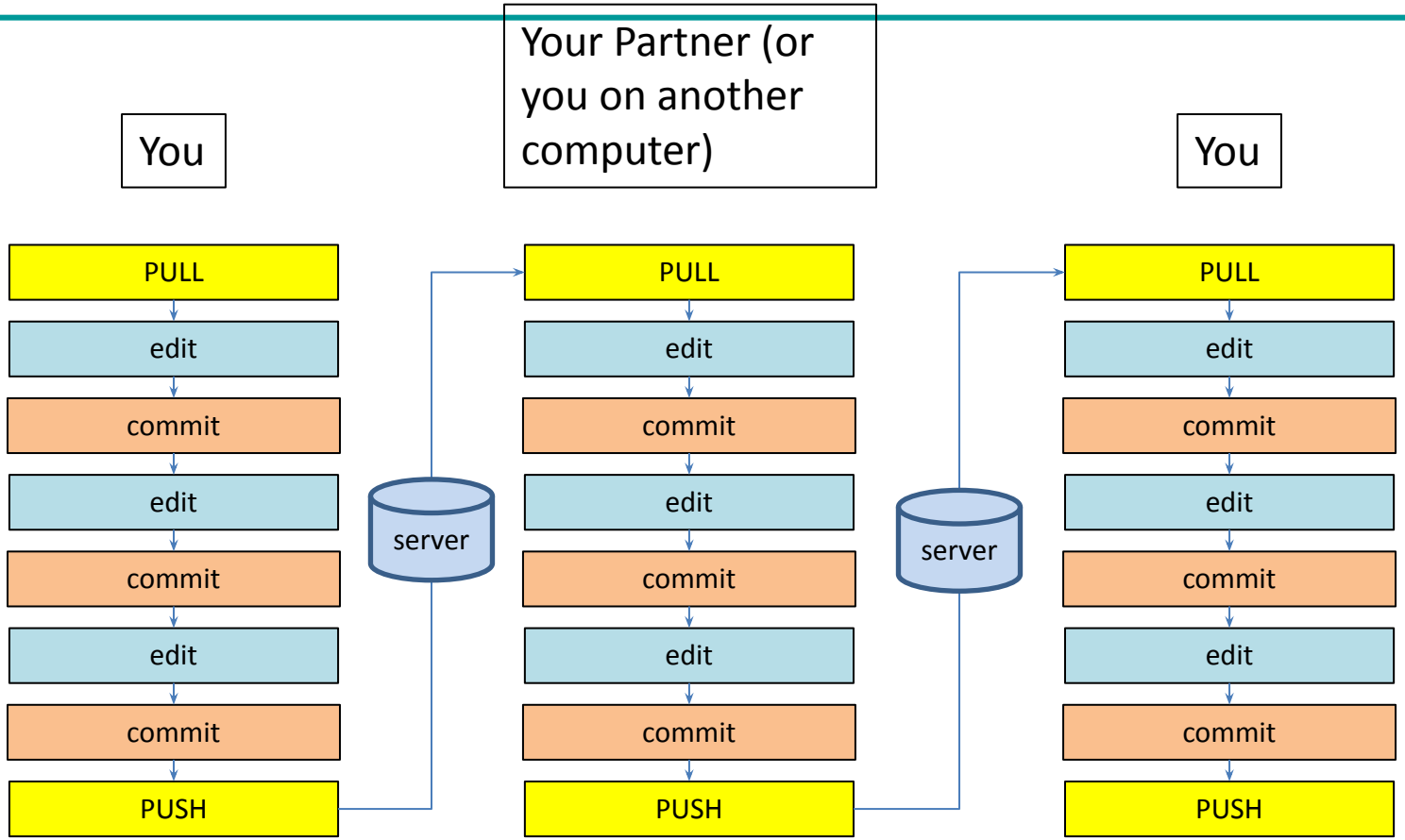
Your workflow (2)



Best practice: commit your work whenever you've gotten one part of your problem working, or before trying something that might fail.

If your new stuff is screwed up, you can always "revert" to your last good commit. (Remember: always "revert", never "roll back")

Your workflow with a partner



Your partner gets your work from the server

You get your partner's work from the server

- Enough! Let's go to the commands now.

Install Git

- Installing Git:
  - https://git-scm.com/book/en/v2/Getting-Started-Installing-Git
- Free online book:
  - http://git-scm.com/book
- Reference page for Git:
  - http://gitref.org/index.html
- Git tutorial:
  - http://schacon.github.com/git/gittutorial.html

Install Git

- Git hub account
  - https://docs.github.com/en/get-started/signing-up-for-github/signing-up-for-a-new-github-account
- Git hub token creation
  - https://docs.github.com/en/enterprise-server@3.4/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token

Tell Git who we are (configuration)

- $ git config **--global** user.name "Your Name"
  - *# Put your quote marks around your name*
- $ git config **--global** user.email yourname@yourplace.org
-

Create a GIT Repository

- cd *# Switch to your home directory.*
- $ pwd *# Print working directory (output should be /home/<username>)*
- $ mkdir paper
- $ cd paper
- git init

Add Files to GIT Repo

- git add <filename>
- git add test.txt
- git status
- git commit -m “Adding the first file”
- git push
- git config credential.helper store

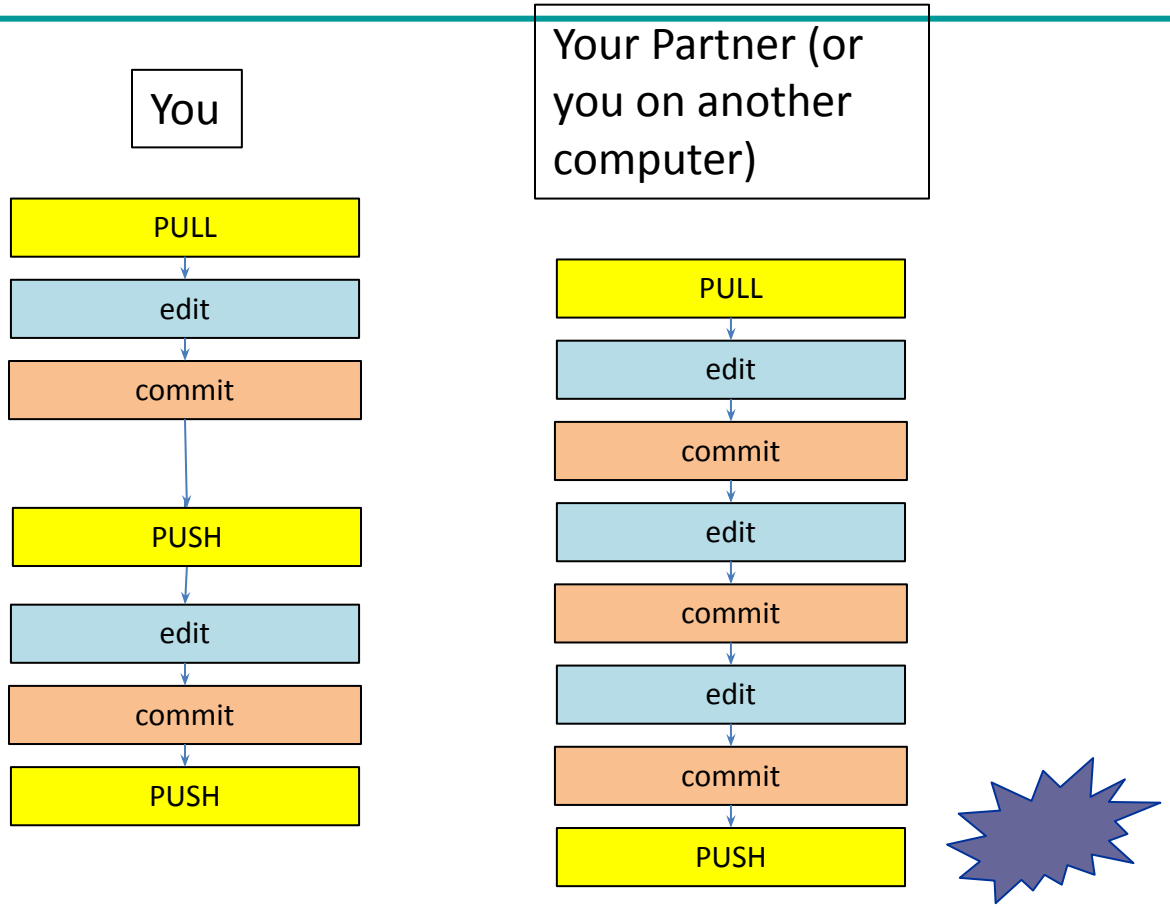
Add Files to GIT Repo

- To see what is modified but unstaged:
  - git diff
-
- To see a log of all changes in your local repo:
  - git log

Pulling Changes

- git pull
-
- To see a log of all changes in your local repo:
  - git log

Your workflow with a partner

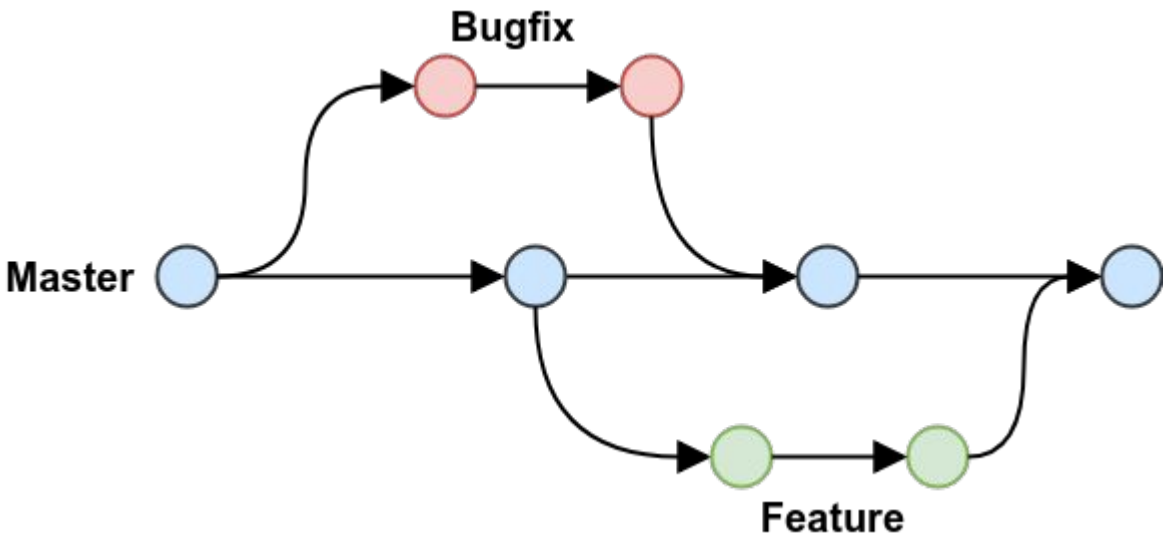


- git pull again!

Merge conflicts

```
<<<<<<< HEAD:index.html
<div id="footer">todo: message here</div>
=======
<div id="footer">
  thanks for visiting our site
</div>
>>>>>>> SpecialBranch:index.html
```

Why version control?



Add Files to GIT Repo

- To create a new local branch:
  - git branch name
- To list all local branches: (* = current branch)
  - git branch
- To switch to a given local branch:
  - git checkout branchname
- To merge changes from a branch into the local master:
  - git checkout master
  - git merge branchname

References

- Miscellaneous resource from internet

Thank you!