

git

Mary-Ellen Petrich

mpetrich@stanford.edu

LinkedIn -> “Git Essential Training: The Basics”

git

- Everyone should have **git** installed
- Everyone should have a **github** account
- Github privacy settings: go into settings, select the email tab, and check the box to set email to private
- Instructions for installation:
<https://carpentries.github.io/workshop-template/#git>
- Class discussion board
<https://cryptpad.fr/pad/#/2/pad/edit/SirilCBDAuqG5NdiOl+BbpRH/>
- git materials
<https://swcarpentry.github.io/git-novice/>

What is Git?

- Git is:
 - a distributed version-control system, for tracking edits in digital files
- GitHub is:
 - a code hosting platform for version control and collaboration

xkcd



Easy to learn the basics

Hard to be an expert

90% of what you need to know

- `git status`
- `git add <file>`
- `git commit -m "My comment."`

There will be a quiz....

xkcd



Easy to learn the basics

Hard to be an expert

90% of what you need to know

- `git status`
- `git add <file>`
- `git commit -m "My comment."`

If you don't know this by the end of this class...

▪ **I have failed!!!**

Why use version control?

- Version control systems have been used for almost 40 years (1972)
- Mainly by software developers
- Now being used by researchers
 - Continuing concern about complete documentation & reproducibility
 - Increased use of software & scripting
 - Increased open access to data
 - Increased collaboration



FINAL.doc!



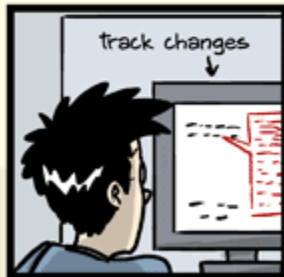
FINAL_rev.2.doc



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

This can be avoided with version control

The basic problem

- Coding is hard and time consuming
- When you get a script to work, you want to save a copy of it.
- Because, if you don't, the very next edit you make will stop it from working.
- And, when you comment out the changes, it still won't work...
- And, it will take you two hours to get the script to work again

The Situation

- You:
 - Obtain a version of open source data
 - Write scripts to generate results from the data
 - Continue to get new versions of the data and to improve your scripts over the course of time

The Question

- Your advisor or funder asks:
 - How did you generate these results last month?
- Another researcher asks:
 - I would like to build on the research in your paper published last year. How do I get a copy of that script?
- A colleague asks:
 - Can I contribute to your work by improving your scripts from two years ago? Where can I get access to them?

Answers

- Bad answer
 - I have no idea. That data and that script are long gone. I got a new laptop since then.
- Good answer
 - The data and the script are available at a publically accessible repository on GitHub.
 - The version of the repository for that data and the script are cited in my paper.

Who should use Git*?

- Anyone wanting to track edits
 - Review a history log of changes
 - View differences between versions
 - Retrieve old versions
- Anyone needing to share changes with collaborators
- Anyone who can use command line tools

*Invented in 2005

The bad news



Version control features

- Unlimited un-do
- Digital lab notebook
 - Automatically tracking changes: what, when, who
- Best for text materials
 - scripts, csv files, web pages, perl, python, C++
- Not as powerful for tracking non-text files
 - powerpoint, excel, movies, images, pdfs
- Tool for sharing and collaboration

Section2: Setting up git (5 min)

- Commands
 - › `git config --list`
 - › `git config --global user.name "Vlad Dracula"`
 - › `git config --global user.email "vlad@tran.sylvan.ia"`
 - › `git config --global color.ui "auto"`
 - › `git config --global core.editor "nano -w"`
 - › `git config --global core.editor "vim"`
 - › `git config -h` #quick reference
 - › `git config --help` #manual
- When done (name, email, editor),
put up a green stickie ✓
- Learn more: See section 2 in the etherpad

Help with git

- google.com
- stackoverflow.com

The screenshot shows the Stack Overflow website interface. The browser's address bar displays the URL `https://stackoverflow.com/questions/tagged/git`. The page title is "Questions tagged [git]". A search bar at the top contains the text "[git]". The left sidebar shows the "Stack Overflow" logo and navigation links for "Home", "PUBLIC", "Stack Overflow", "Tags", "Users", and "Jobs". Below the sidebar, there is a "Teams" section with a "Learn More" button. The main content area displays the description of the "git" tag: "Git is an open-source distributed version control system (DVCS). Use this tag for questions related to Git usage and workflows. Do not use this tag for general programming questions that happen to involve a Git repository." Below the description, there are links for "Learn more...", "Top users", and "Synonyms". A summary bar indicates "103,828 questions" and provides filters for "Info", "Newest", "Featured" (4), "Frequent", "Votes", "Active", and "Unanswered". The first question listed is "How can I undo my last commit in Git and stash those changes instead?", which has 0 votes, 1 answer, and 10 views. The second question is "Sync local project folder from git", which also has 0 votes and 1 answer.

← → ↻ 🏠 `https://stackoverflow.com/questions/tagged/git` 🔍 linux path for a lir →

MEP_Quick Latest Headlines ⚙️ Most Visited 📌 Getting Started 🌐 Save Page to Waybac... 🌐 Getting Started

stackoverflow [git]

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

Teams
Q&A for work

[Learn More](#)

Questions tagged [git]

[Ask Question](#)

Git is an open-source distributed version control system (DVCS). Use this tag for questions related to Git usage and workflows. Do not use this tag for general programming questions that happen to involve a Git repository.

[Learn more...](#) [Top users](#) [Synonyms](#)

103,828 questions

Info	Newest	4 Featured	Frequent	Votes	Active	Unanswered
------	--------	------------	----------	-------	--------	------------

0 votes

How can I undo my last commit in Git and stash those changes instead?

I have a branch called feat-a and I essentially want to "undo" my last local commit and just have those files staged. Then when they are all staged I want to stash those changes. Now what I'm doing ...

[git](#)

asked 12 mins ago

coltonidle 1 • 2

1 answer

10 views

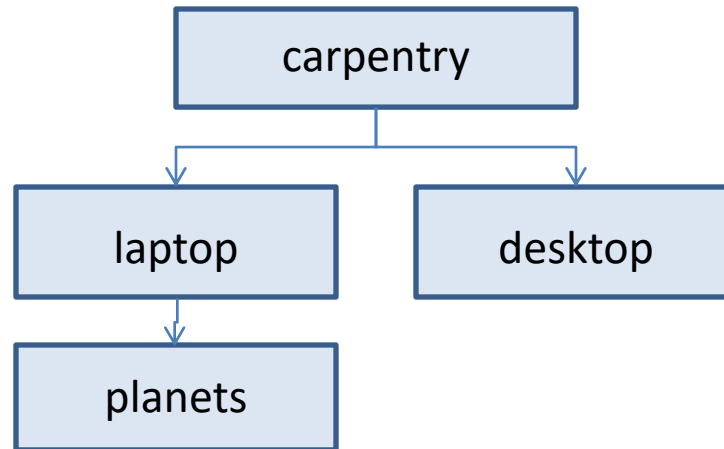
0 votes

Sync local project folder from git

I've created a git repository where, my colleague are pushing codes for front-end section of application as contributors and I'm pushing codes backend section. Is it possible to sync our project ...


1 answer

Creating a repository (10 min)

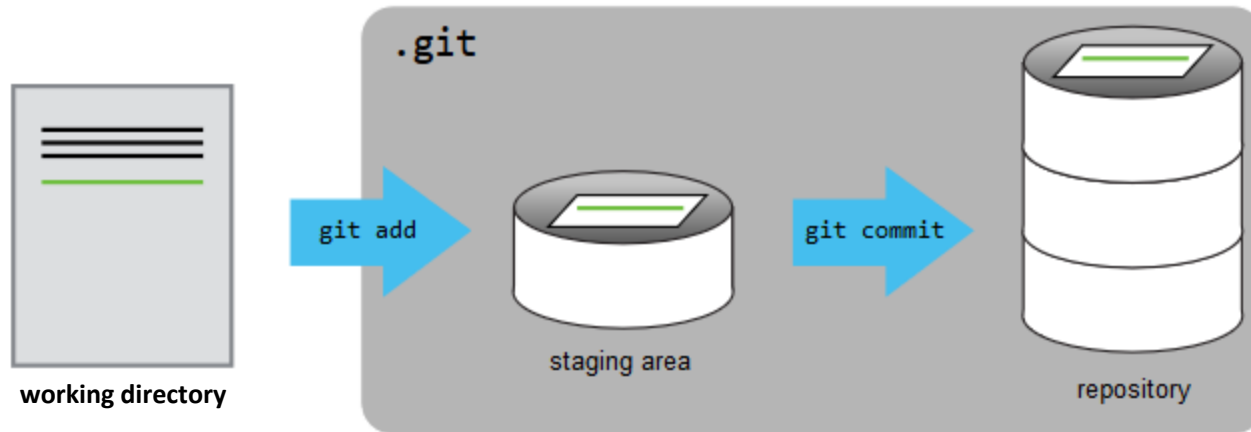


- % [go to your work space and make the folders above]
- % [navigate to the planets directory]
- % [List all file/dir names; you should see 2]
- % git init
- % [List all file/dir names; you should see 3]!
- % git status
- % [In the planets directory, create a file called mars.txt. Add a line of text to the file.]
- % git status
- % When done, put up the green stickie ✓

.git

- The directory where all the backups are stored
- Not human-readable
- Accessible through the git program
- There should never be two in the same hierarchy of files
- Don't want a git repository? Delete the .git directory.
- `rm -rf .git` 

add vs commit



- git requires that we *add* files to the staging area
- This allows us to create change-sets with specific changes rather than committing all changes.

How to

- Add and commit with one command:
 - `git commit -am "Update my poem."`
- Add multiple files:
 - `git add mars.txt; git add venus.txt`
 - `git add *.txt` #adds all .txt files
 - `git add mars.txt venus.txt`
 - `git add .` #adds all files

git diff

- 3 of these is the same, which ones?
- `git diff mars.txt`
- `git diff --staged mars.txt`
- `git diff HEAD mars.txt`
- `git diff HEAD~0 mars.txt`
- `git diff HEAD~1 mars.txt`

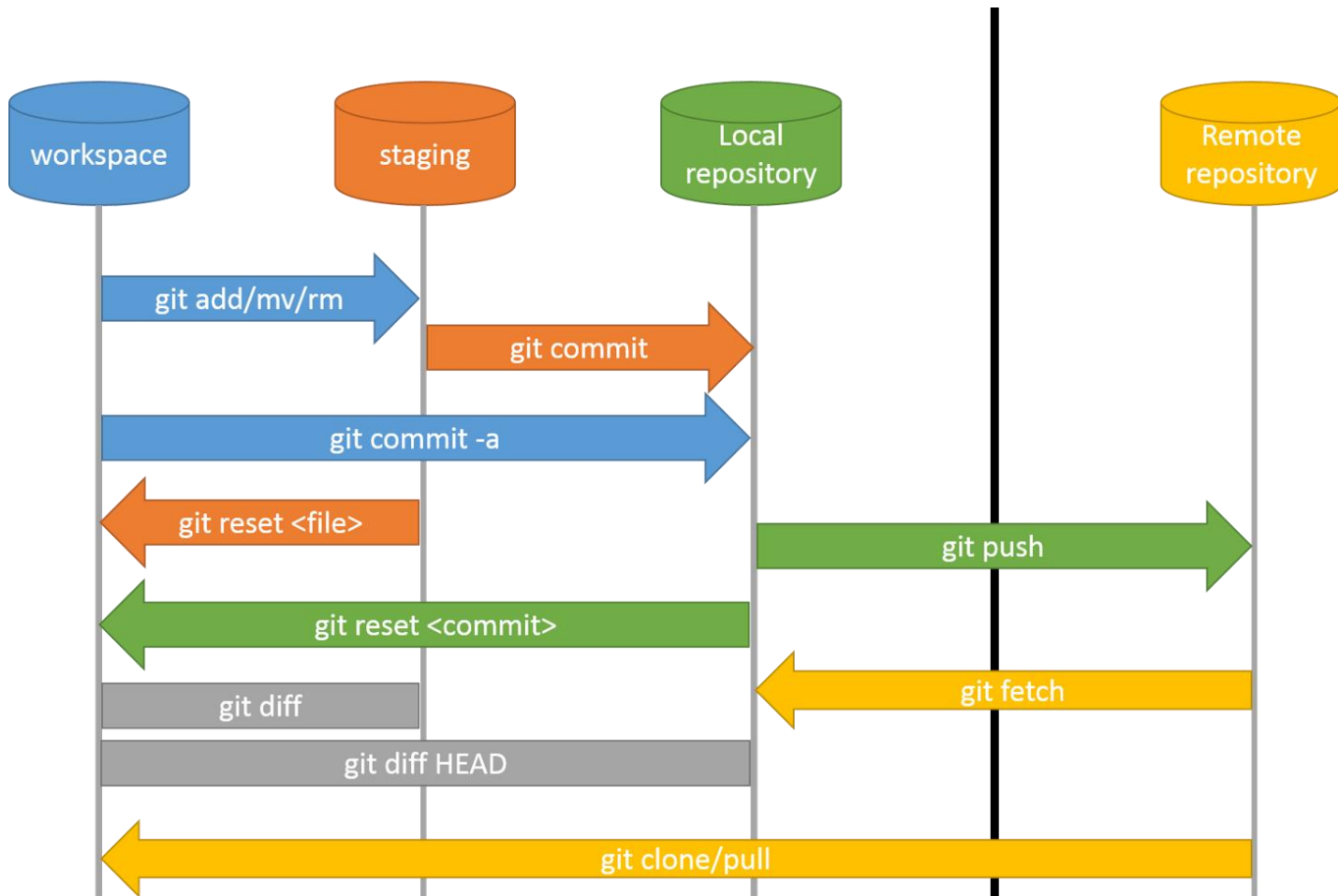
git checkout

- `echo "We will need potatoes on Mars." > mars.txt`
- `cat mars.txt`
- `git status`
- `git diff`
- Which will correct the problem:
 - `git checkout mars.txt`
 - `git checkout HEAD mars.txt`
 - `git checkout HEAD~0 mars.txt`
 - `git checkout HEAD~1 mars.txt`

Section 7: Remotes on Gitub

- Log into GitHub
- Upper-right corner + to 'create repository'
- Type **planets** in for the repository name
- Click button **Create repository**
- Make sure **HTTPS** is selected
- Check out the helpful instructions. Which set should you follow?
- Copy the two instructions from the web page to the root of your repository to push the content to GitHub
- To confirm, try `git remote -v`
- View your commits on GitHub. Explore
- Look at Qs for section 7 in the etherpad.

Git commands



git

Mary-Ellen Petrich

mpetrich@stanford.edu

LinkedIn -> “Git Essential Training: The Basics”

xkcd on commit messages



	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAAANDS	2 HOURS AGO

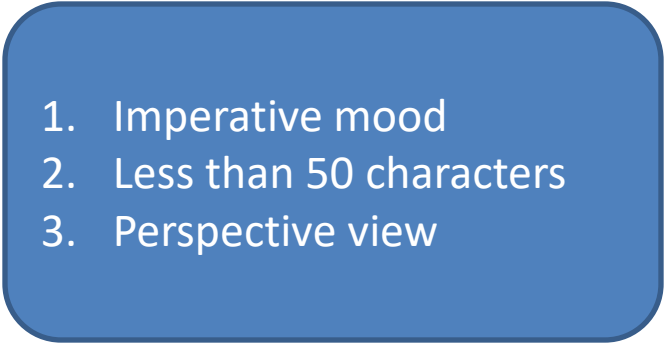
AS A PROJECT DRAGS ON, MY GIT COMMIT
MESSAGES GET LESS AND LESS INFORMATIVE.

Good Commit Messages

```
% git commit -m "Initialize project"
```



This is a
commit
message.

- 
1. Imperative mood
 2. Less than 50 characters
 3. Perspective view

Section 6: Ignoring Files

- [make sure you are in the folder planets]
- for num in {1..5}; do touch file\$num.dat; done
- [create a folder *results*, add 2 files: a.out & b.out]
- go back to the planets directory!
- git status [what do you see?]
- [create and edit a file called .gitignore]
 - *.dat
 - results/
- git status
- [add and commit the file that is untracked]
- git add file1.dat [what happened?]
- git status --ignored [what do you see?]
- [Look at Qs for section 6 in the etherpad]

Avoiding Conflicts

- Have small single purpose files
- Pull more frequently
- Make smaller commits
- Clarify who is responsible for what sections
- Discuss what order work will be done in
- Establish coding style norms