

Welcome to GitHub or die tryin'!

1. Go to datacamp.com
2. Create an account
3. Sign up for the “introduction to git for data science” class
4. Go to github.com/join and create an account

 spncrfx@gmail.com

 @foxandtheflu

 sjfox











Github or Die Tryin'

Spencer Fox
23 March 2018

PhD Candidate UT Austin (2018)
Dr. Lauren Ancel Meyers

Traditional version control issues

<input type="checkbox"/>	Name	Date modified	Type
	Rscript_4_21_2016.R	5/1/2016 3:03 PM	R File
	Rscript_4_22_2016a.R	5/1/2016 3:03 PM	R File
	Rscript_4_22_2016b.R	5/1/2016 3:03 PM	R File
	Rscript_4_24_2016.R	5/1/2016 3:03 PM	R File
	Rscript_final.R	5/1/2016 3:03 PM	R File
	Rscript_final_final.R	5/1/2016 3:03 PM	R File
	Rscript_really_final.R	5/1/2016 3:03 PM	R File
	Rscript_really_really_final_final.R	5/1/2016 3:03 PM	R File

Facilitates collaboration and prevents waiting



vs.



A public GitHub account gets you exposure that can lead to jobs and other opportunities

A public GitHub account gets you exposure that can lead to jobs and other opportunities

All publicity is good publicity.

(Proverb)

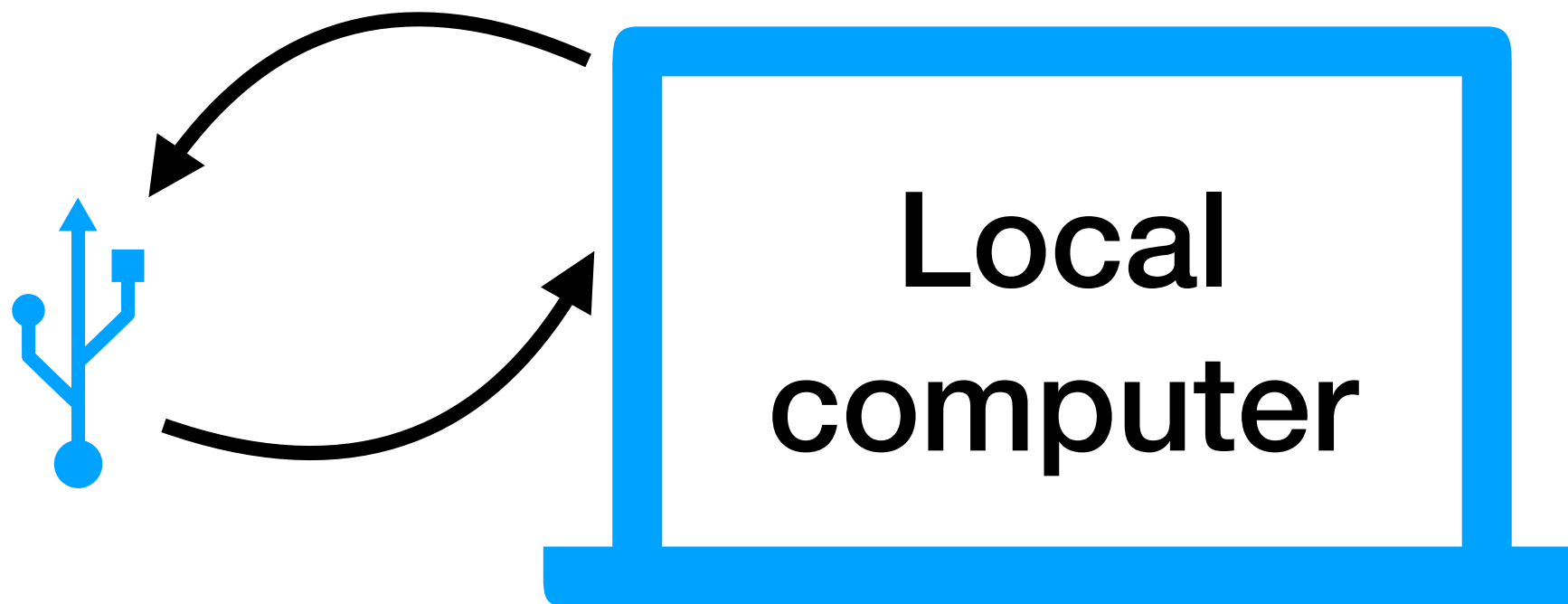
A public GitHub account gets you exposure that can lead to jobs and other opportunities

Any visible code is good code

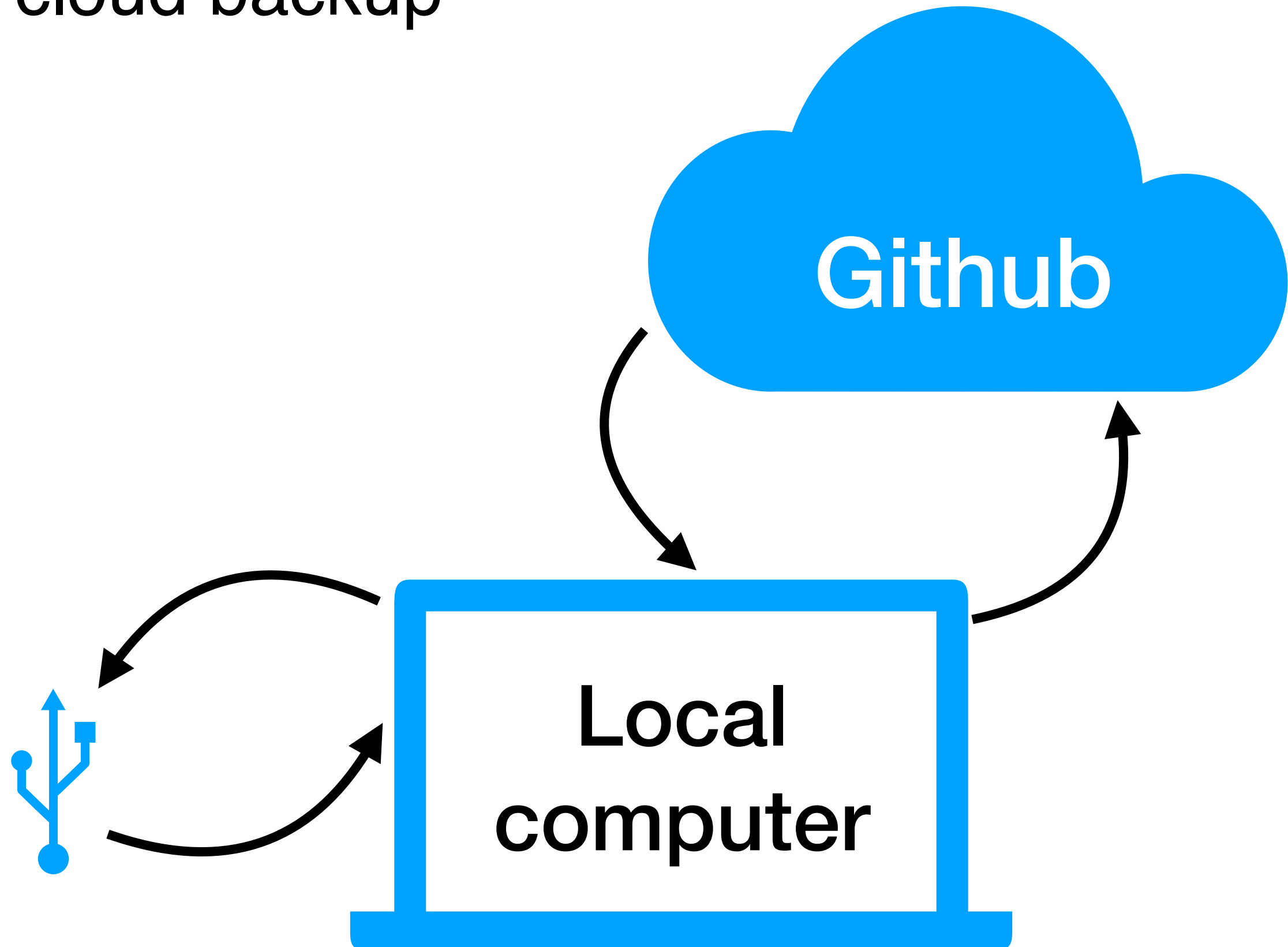
~~All publicity is good publicity.~~

(Proverb)

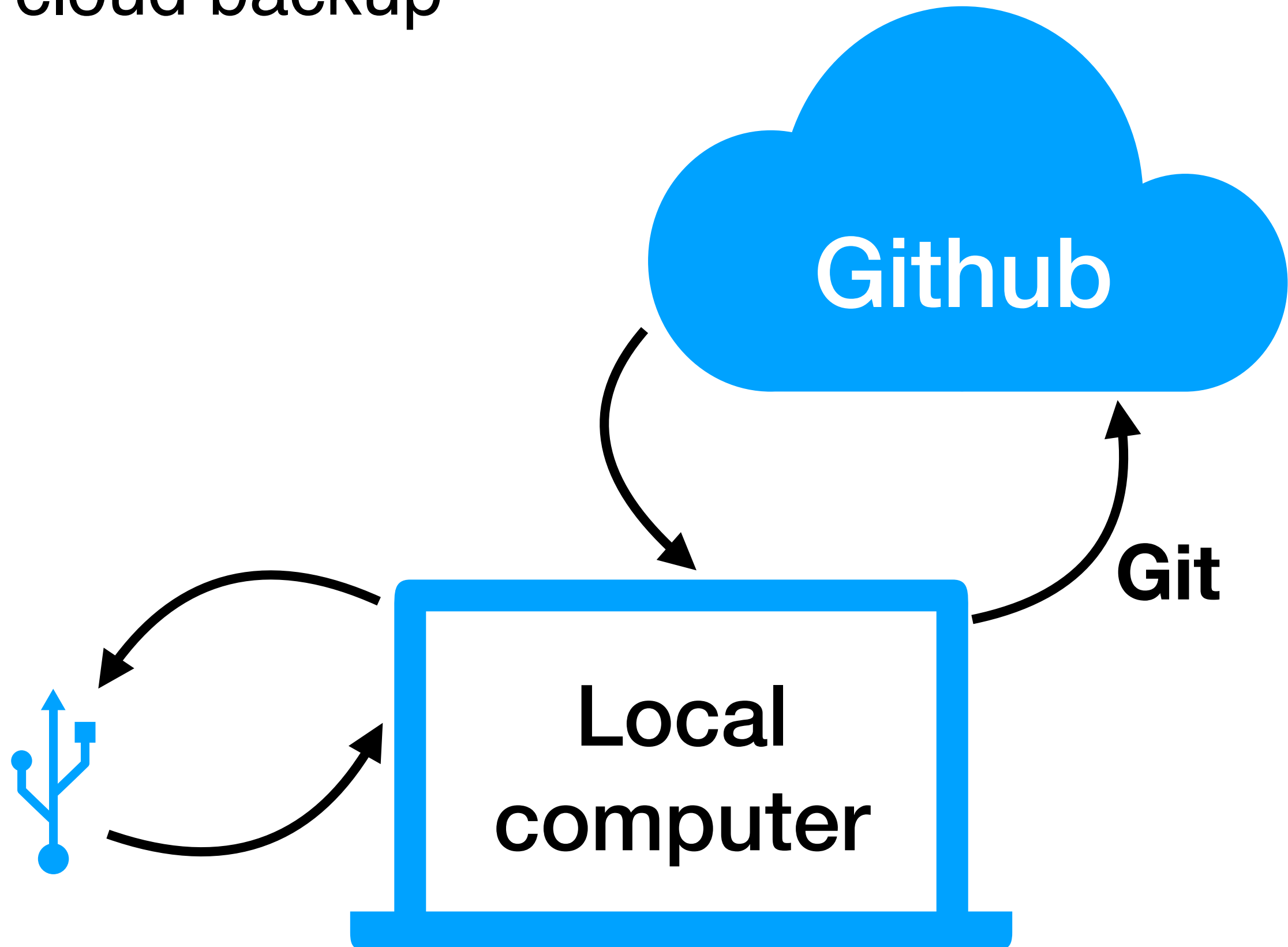
Traditional programming occurs on your own machine and (hopefully) a backup



At the most basic level, git/hub is just a cloud backup



At the most basic level, git/hub is just a cloud backup



Getting started on GitHub alone

<http://happygitwithr.com/>

Getting started on GitHub alone

1. Install Git on your machine



<http://happygitwithr.com/>

Getting started on GitHub alone

1. Install Git on your machine
2. Sign up for GitHub



<http://happygitwithr.com/>

Getting started on GitHub alone

1. Install Git on your machine
2. Sign up for GitHub
3. Create a repository for existing or new project



<http://happygitwithr.com/>

Getting started on GitHub alone

1. Install Git on your machine
2. Sign up for GitHub
3. Create a repository for existing or new project
4. Integrate GitHub backups regularly in your workflow



<http://happygitwithr.com/>

Questions?

Mac

Windows

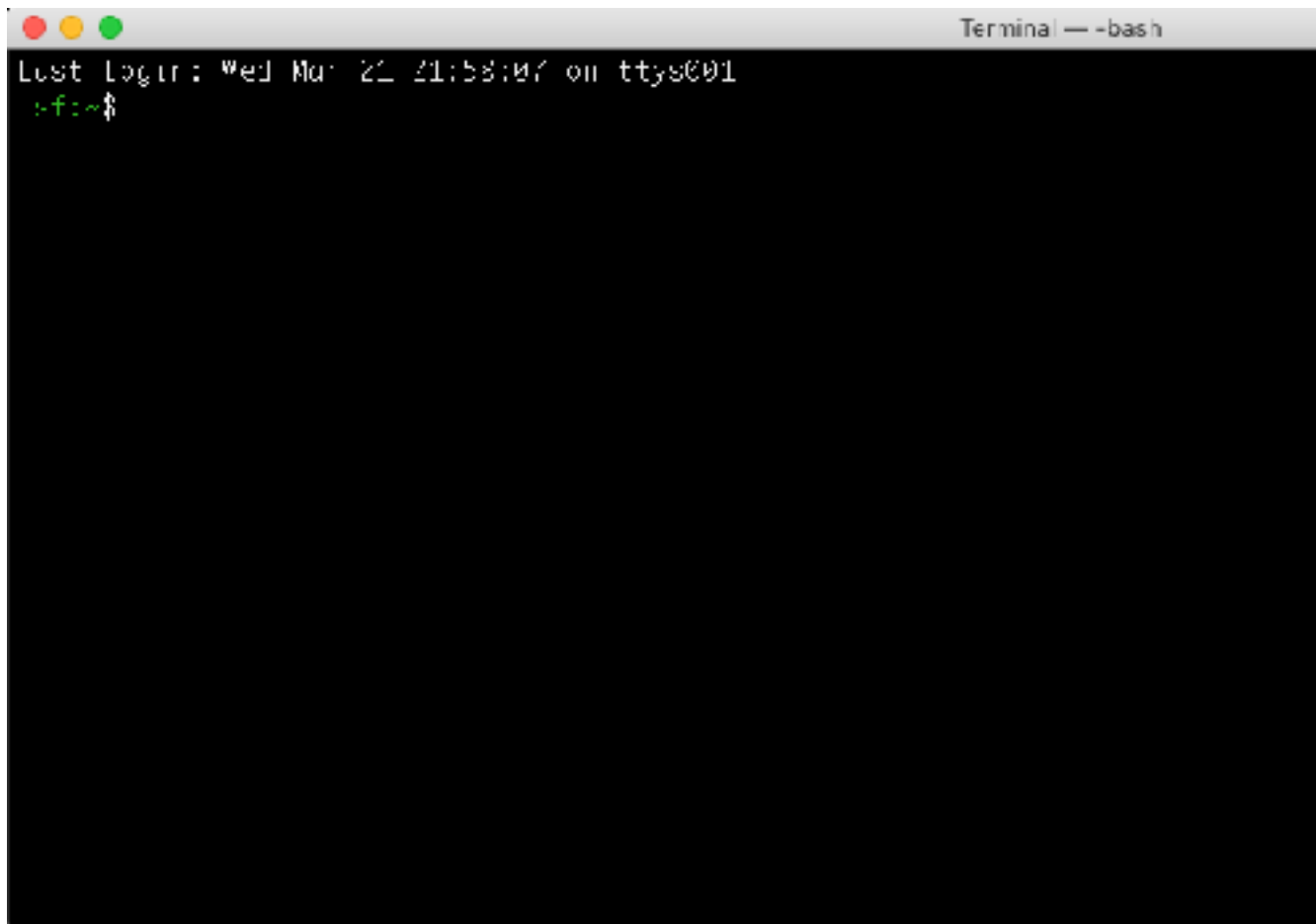
Mac

1. install git

Windows

Mac

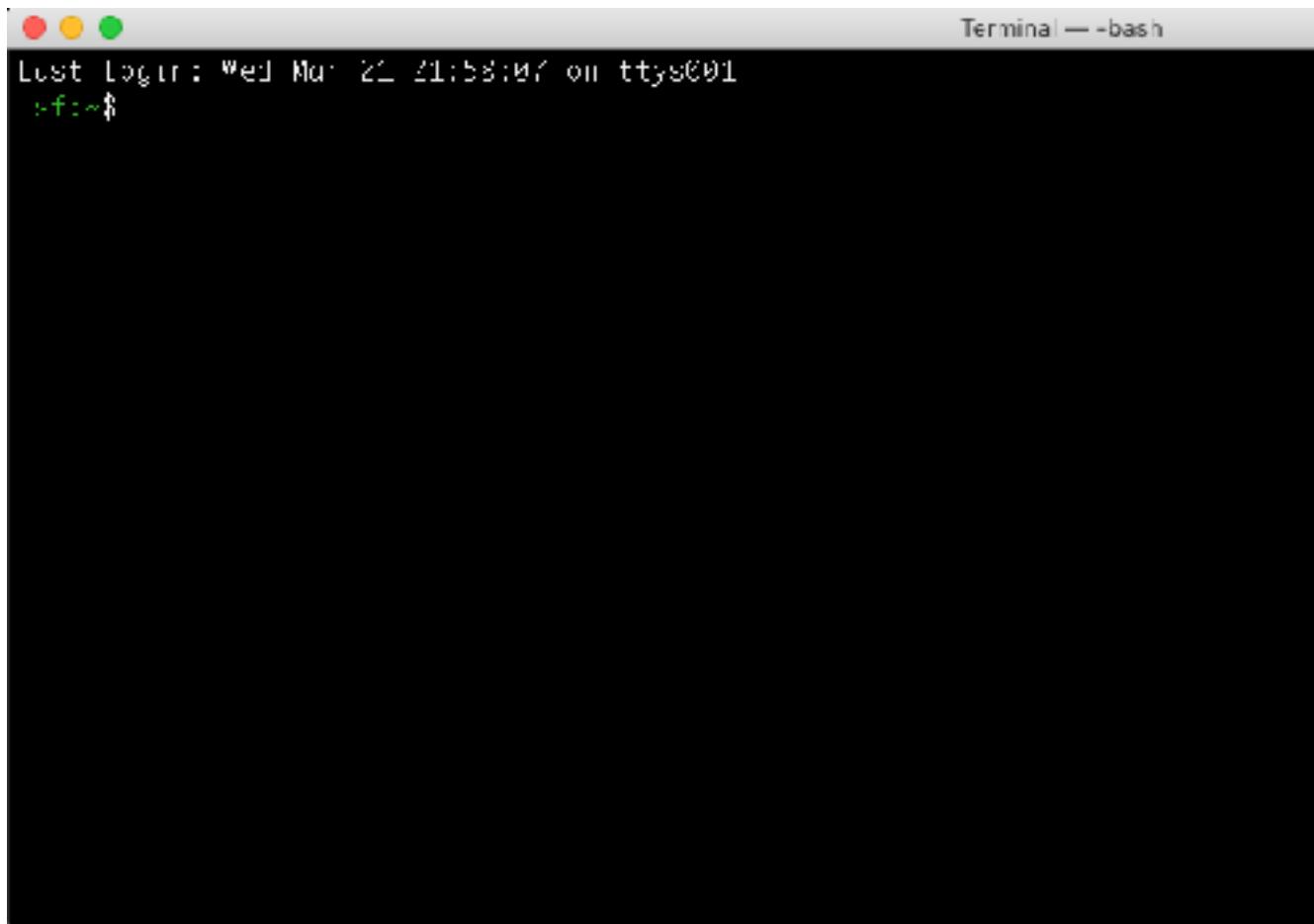
1. install git
2. open terminal window



Windows

Mac

1. install git
2. open terminal window

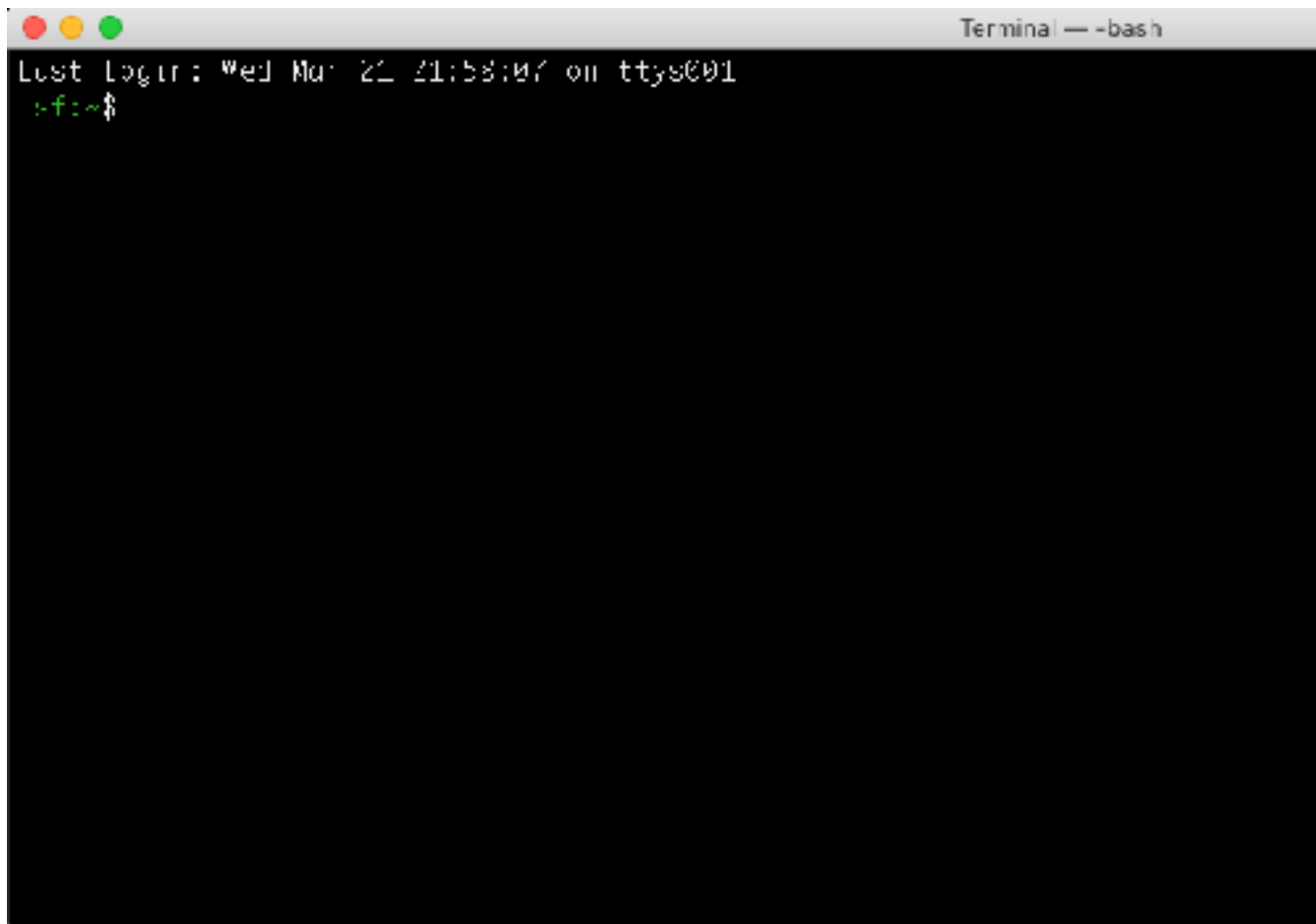


Windows

1. install git

Mac

1. install git
2. open terminal window

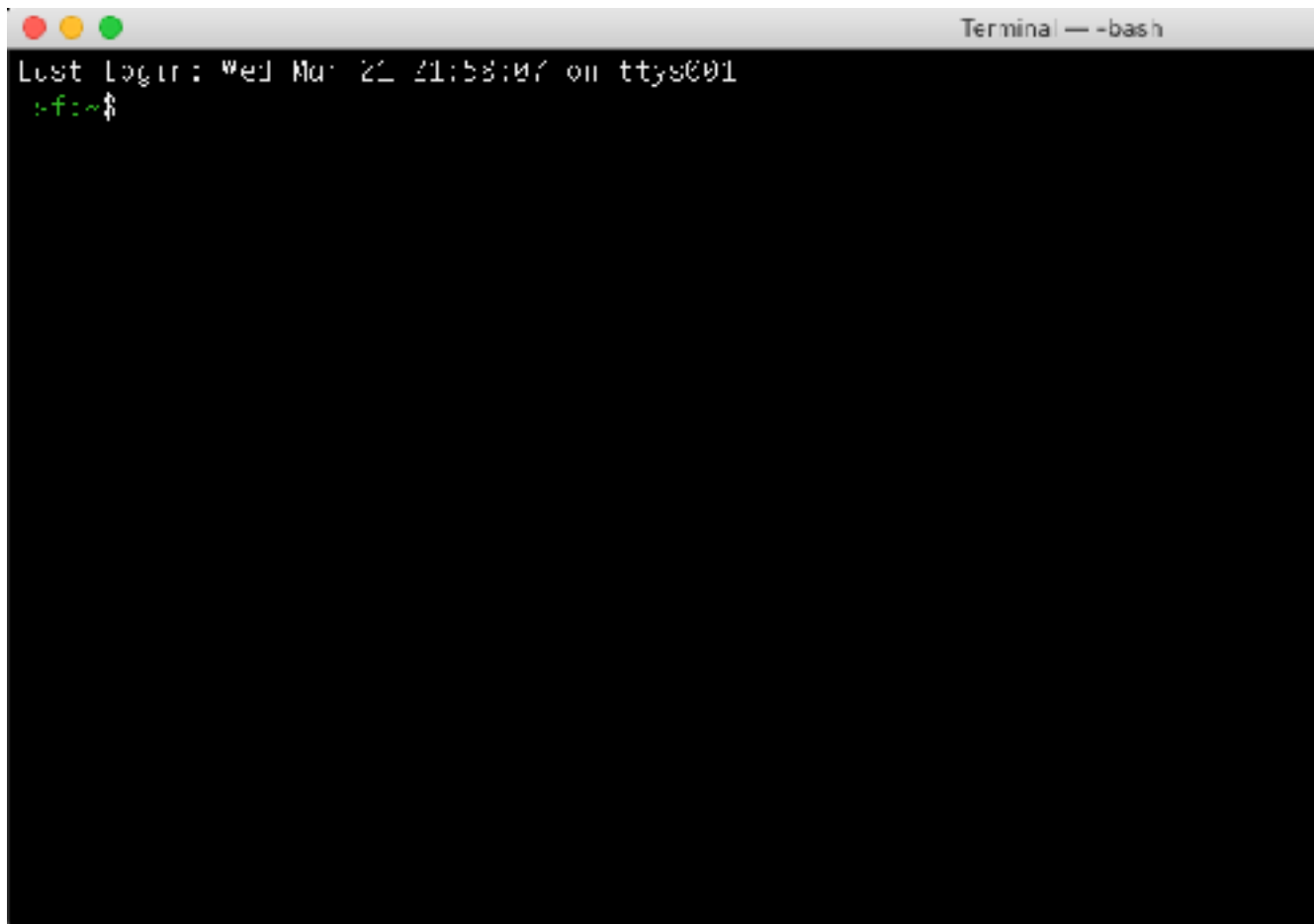


Windows

1. install git
2. install git bash

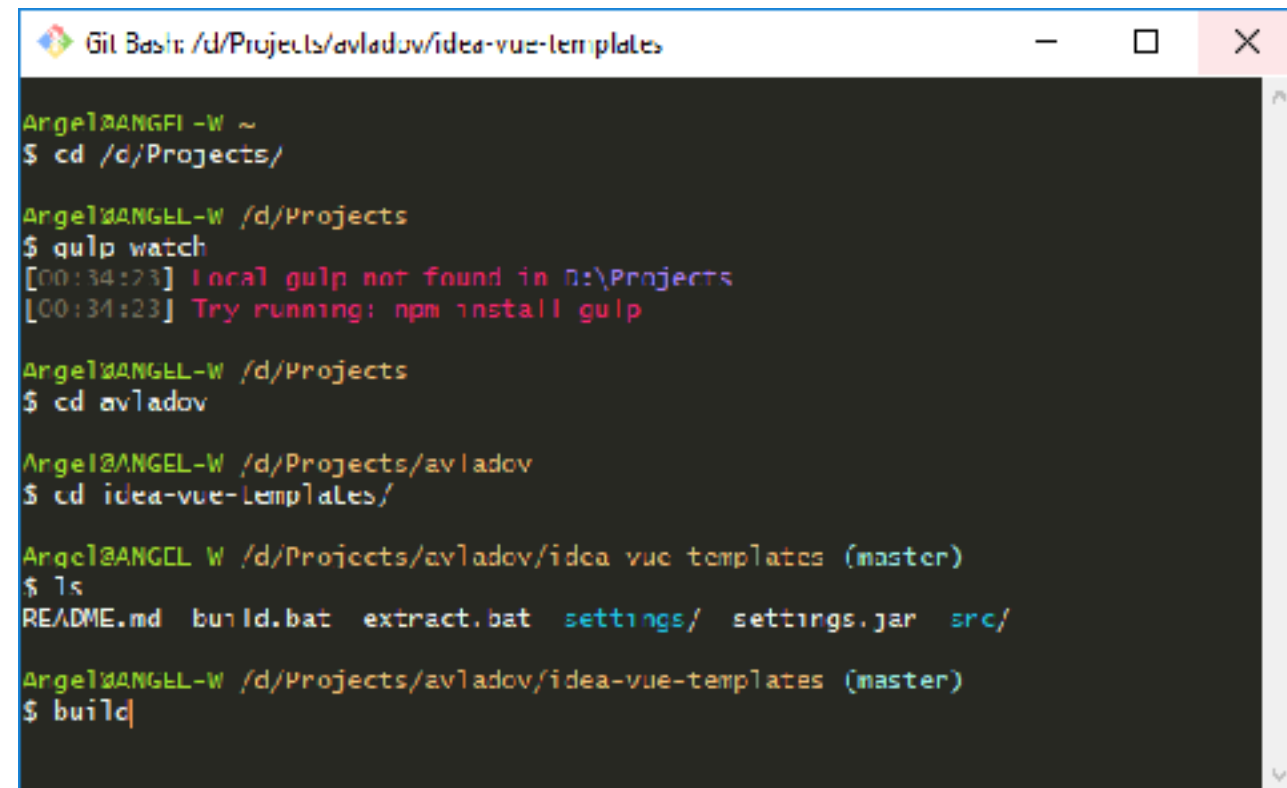
Mac

1. install git
2. open terminal window

A screenshot of a Mac Terminal window. The title bar shows 'Terminal — -bash'. The text inside the terminal reads: 'Last login: Wed Mar 22 21:58:07 on ttys001', followed by a prompt 'af:~\$'.

Windows

1. install git
2. install git bash
3. open git bash

A screenshot of a Windows Git Bash window. The title bar shows 'Git Bash: /d/Projects/avladov/idea-vue-templates'. The text inside the terminal shows a series of commands and their outputs: 'Angel@ANGEL ~', '\$ cd /d/Projects/', 'Angel@ANGEL-W /d/Projects', '\$ gulp watch', '[00:34:23] local gulp not found in D:\Projects', '[00:34:23] Try running: npm install gulp', 'Angel@ANGEL-W /d/Projects', '\$ cd avladov', 'Angel@ANGEL-W /d/Projects/avladov', '\$ cd idea-vue-templates/', 'Angel@ANGEL W /d/Projects/avladov/idea vue templates (master)', '\$ ls', 'README.md build.bat extract.bat settings/ settings.jar src/', 'Angel@ANGEL-W /d/Projects/avladov/idea-vue-templates (master)', '\$ build'.

DataCamp (datacamp.com)

DataCamp (datacamp.com)

- **DataCamp is a ~free website to learn data skills**
- **Has many tutorials**
- **Don't need to download anything**
- **Provides instantaneous feedback**

EXERCISE

How can I check the state of a repository?

When you are using Git, you will frequently want to check the **status** of your repository. To do this, run the command `git status`, which displays a list of the files that have been modified since the last time changes were saved.

You have been put in the `dental` repository. Use `git status` to discover which file(s) have been changed since the last save. Which file(s) are listed?

INSTRUCTIONS 50XP

Possible Answers

- ☐ `data/summer.csv` . press 1
- ☒ `report.txt` . press 2
- ☐ Neither of the above. press 3
- ☐ Both of the above. press 4

💡 Take Hint (-15xp)

Submit Answer

TERMINAL

```
$ cd dental
$ ls
bin  data  report.txt  results
$ pwd
/home/repl/dental
$
```

Time to learn how to hack



Time to learn how to hack



Working within a terminal

Working within a terminal

1. **\$ pwd - What folder (directory) am I in?**

Working within a terminal

- 1. \$ pwd - What folder (directory) am I in?**
- 2. \$ ls - List the files in this folder**

Working within a terminal

1. **\$ pwd - What folder (directory) am I in?**
2. **\$ ls - List the files in this folder**
3. **\$ cd - Change folders (directories)**

Working within a terminal

1. **\$ pwd - What folder (directory) am I in?**
2. **\$ ls - List the files in this folder**
3. **\$ cd - Change folders (directories)**
4. **\$ git XXXX - run the XXXX git command**

Working within a terminal

1. `$ pwd` - What folder (directory) am I in?
2. `$ ls` - List the files in this folder
3. `$ cd` - Change folders (directories)
4. `$ git XXXX` - run the XXXX git command

```
sf:projects$ pwd
/Users/spencerfox/projects
sf:projects$ ls
INoDS-model          individual_risk        vagrantfiles_dont_delete
cowplot              network              websites
dynamic_ts_modeling  rnot_updater         workshops
ebola                rtZIKVrisk           zikaEstimatorR
evolutionary_songs   test_iphone_app      zika_alarm
flu                  tsetseProject
sf:projects$ cd zikaEstimator/
sf:zikaEstimator$ ls
DESCRIPTION          ZikaEstimator.Rproj  data_produced        ms_figs
NAMESPACE            cpp                  favicon.ico           raw_data.zip
R                    data                  figs                  to_be_removed
README.md             data_for_repository  launcher
sf:zikaEstimator$
```

Atlassian

SourceTree

The screenshot displays the SourceTree application window for the 'sourcetree-website' repository. The interface includes a top toolbar with actions like Commit, Pull, Push, Branch, Merge, and Shelve. A left sidebar contains navigation options: WORKSPACE, File status, History (selected), Search, BRANCHES, BOOKMARKS, TAGS, REMOTES, SHELVED, and SUBREPOSITORIES. The main area shows a commit history table with columns for Graph, Commit, Author, Description, and Date. The table lists various commits, with the top commit being b7358c7 by Rahul Chhabra, titled 'Removing ol...'. The commit history is visualized with a graph on the left side of the table.

Graph	Commit	Author	Description	Date
	b7358c7	Rahul Chha...	master origin/master origin/HEAD Removing ol...	Mar 3, 2016, 11:...
	bdb8bef	Rahul Chhab...	Merged in update-google-verification (pull request #14)	Feb 18, 2016, 1:3...
	dfe975d	Tyler Tadej...	origin/update-google-verification Update google verificati...	Feb 11, 2016, 2:2...
	3bc3290	Tyler Tadej...	Replace outdated Atlassian logo in footer with base-64 en...	Feb 11, 2016, 2:1...
	dba47f9	Tyler Tadej...	Add gitignore	Feb 11, 2016, 1:3...
	ff67b45	Mike Minns...	Updated Mac min-spec to 10.10	Feb 15, 2016, 11:...
	72d32a8	Michael Min...	Merged in hero_images (pull request #13)	Feb 15, 2016, 10:...
	246c4ff	Joel Unger...	origin/hero_images hero_images Used TinyPNG to c...	Feb 11, 2016, 3:3...
	9d9438c	Joel Unger...	Replacing hero images with new version of SourceTree	Feb 9, 2016, 2:59...
	ce75b63	Michael Min...	Merged in bug/date-https (pull request #12)	Feb 15, 2016, 10:...
	85367bb	Patrick Tho...	origin/bug/date-https fixed date and https errors	Jan 7, 2016, 12:2...
	4f9b557	Joel Unger...	New Favicon	Feb 8, 2016, 3:55...
	384e6d5	Rahul Chhab...	origin/search-console-access search console google ver...	Feb 3, 2016, 2:09...
	6fa47a9	Mike Minns...	updated to move supported version to OSX 10.9+	Dec 15, 2015, 2:0...
	8dd87bb	Mike Minns...	remove extra , when a line is skipped due to empty server	Nov 23, 2015, 2:2...
	faa195e	Mike Minns...	Skip records with empty server/user id as gas rejects them	Nov 23, 2015, 2:1...
	0cdf96	Mike Minns...	corrected paths after merge	Nov 23, 2015, 2:0...
	051ab1b	Mike Minns...	corrected column counting	Nov 23, 2015, 1:5...
	a723bc2	Mike Minns...	Merge branch 'au2gex'	Nov 23, 2015, 1:5...
	65fd580	Mike Minns...	deal with invalid instanceids	Nov 23, 2015, 1:5...
	500a892	Michael Min...	Merged in au2gex (pull request #11)	Nov 23, 2015, 1:0...

Let's sign up for datacamp

1. Go to datacamp.com
2. Create an account
3. Sign up for the “introduction to git for data science” class
4. Feel free to start the first couple of exercises (don't worry if you don't get to this step or if it's confusing)

Back up your repository on GitHub

Back up your repository on GitHub

1. `git add`
2. `git commit`
3. `git push`

Back up your repository on GitHub

1. git add
2. git commit
3. git push

```
sf:~$ git add example_file.R
```

1. Choose which files you want to back up

Back up your repository on GitHub

1. git add
2. git commit
3. git push

```
sf:~$ git add example_file.R  
sf:~$ git commit -m "example"
```

1. Choose which files you want to back up
2. Package those files into a single “commit”

Back up your repository on GitHub

1. git add
2. git commit
3. git push

```
sf:~$ git add example_file.R  
sf:~$ git commit -m "example"  
sf:~$ git push
```

1. Choose which files you want to back up
2. Package those files into a single “commit”
3. Send package(s) to the cloud

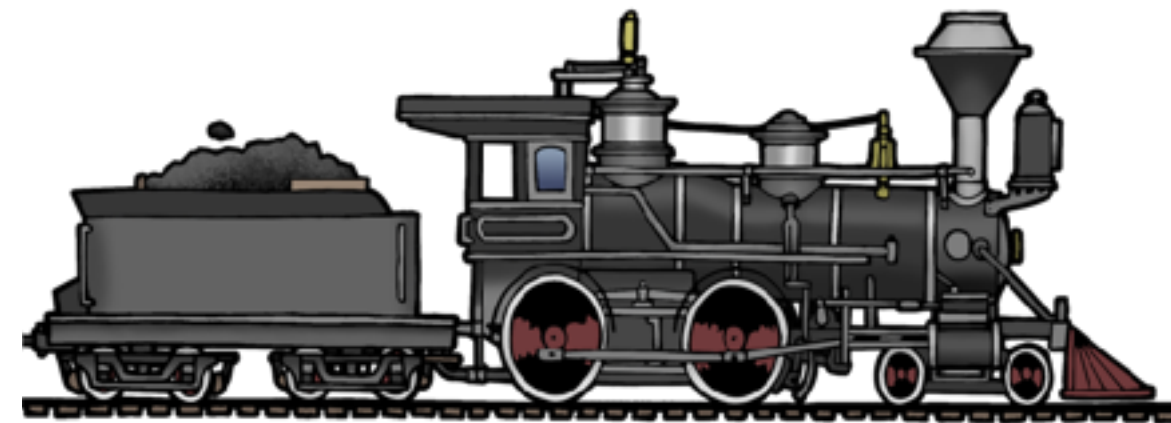
Questions?



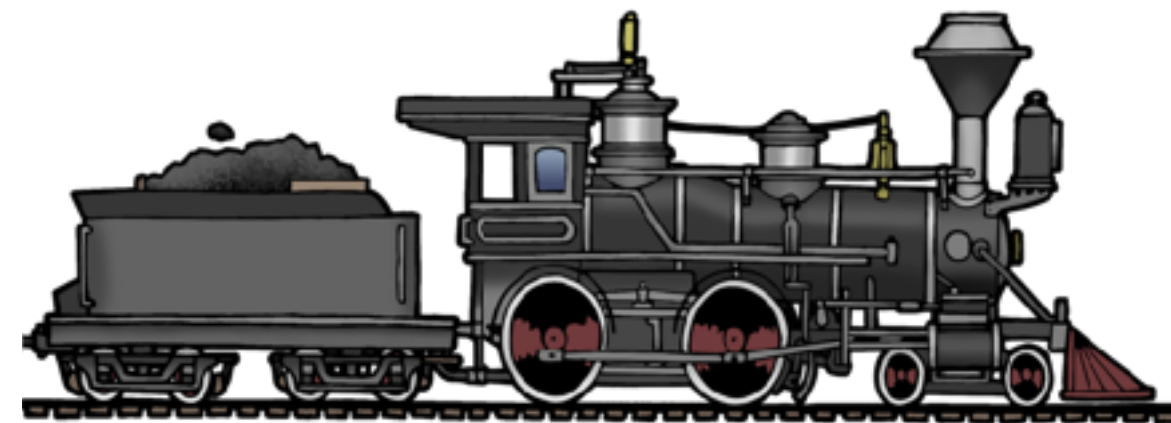
GitHub Repository

Your local machine

1. Initialize your repository

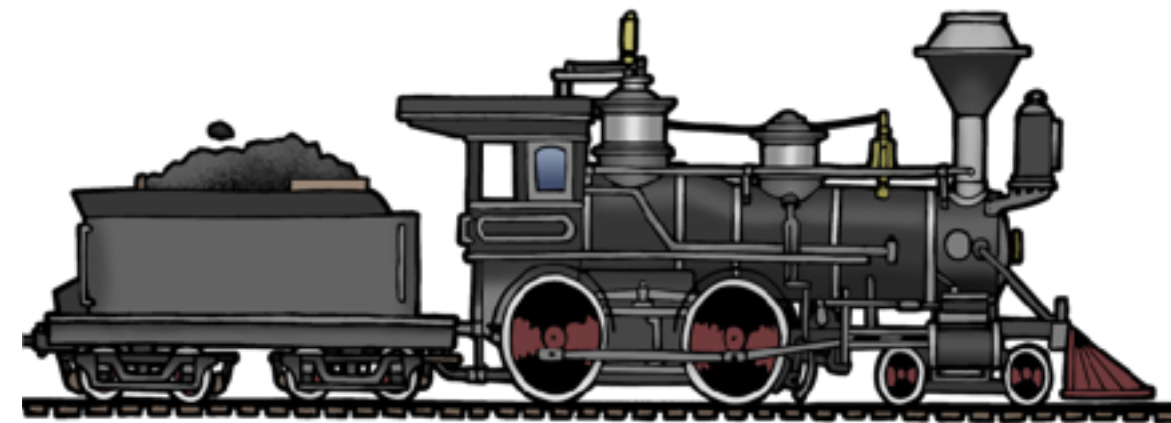


GitHub Repository

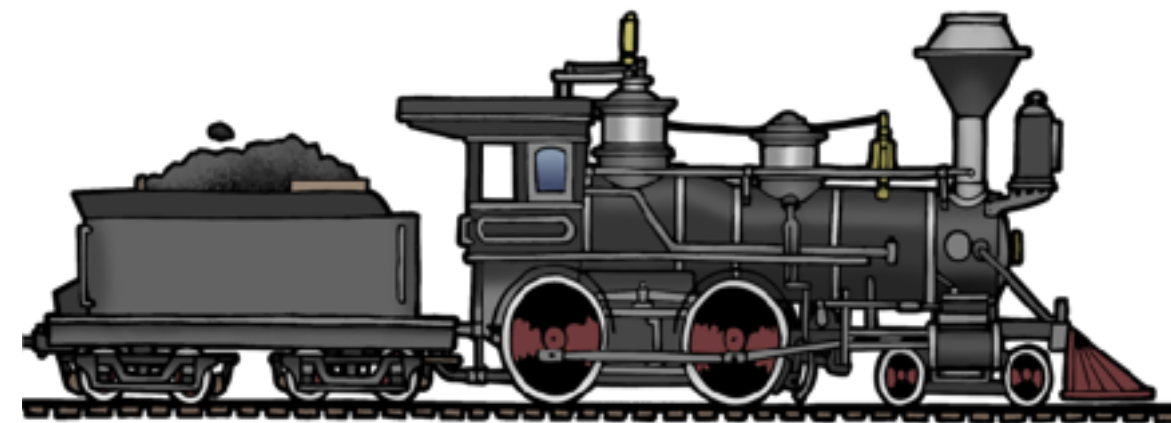
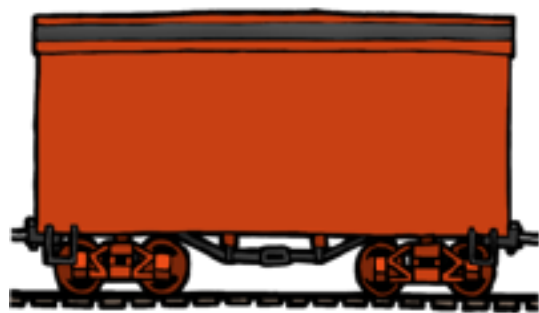


Your local machine

1. Initialize your repository

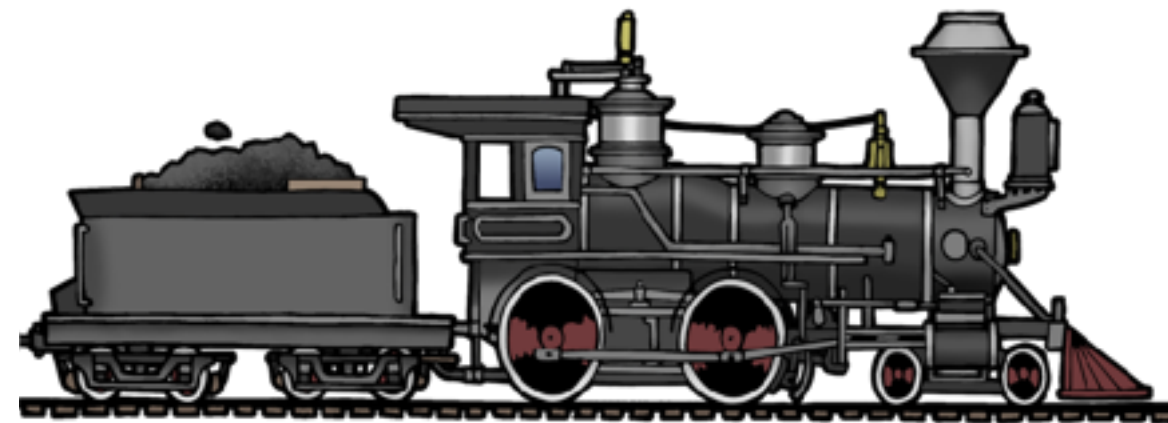


GitHub Repository

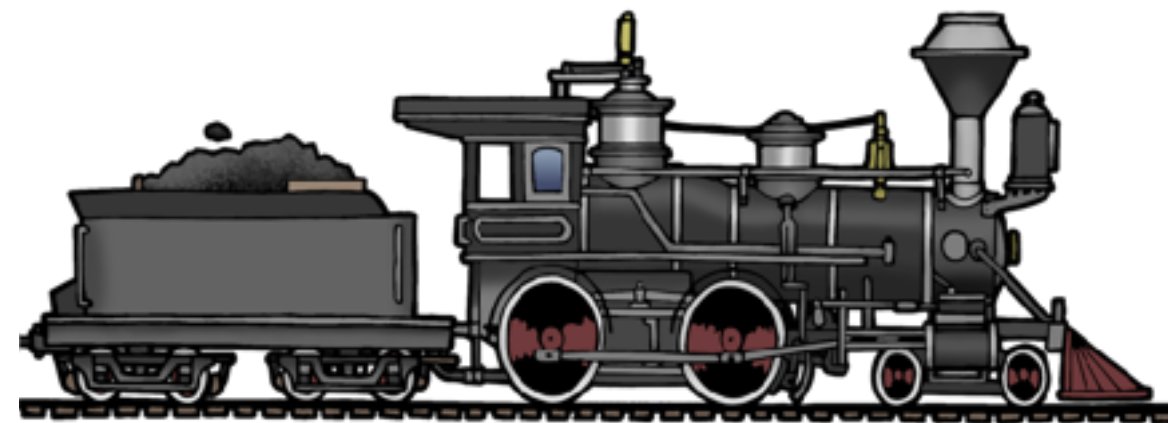
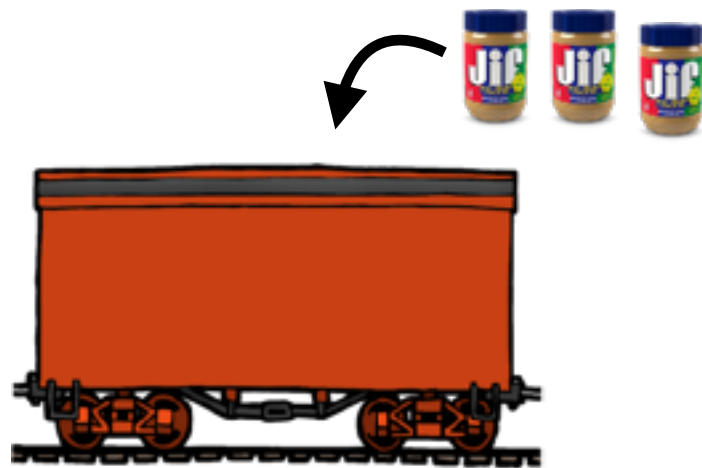


Your local machine

1. Initialize your repository
2. \$ git add

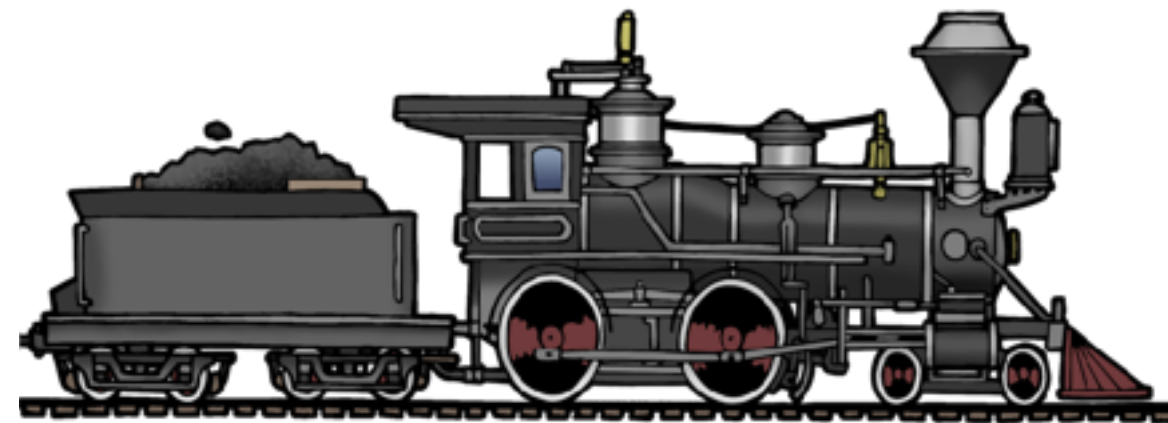


GitHub Repository

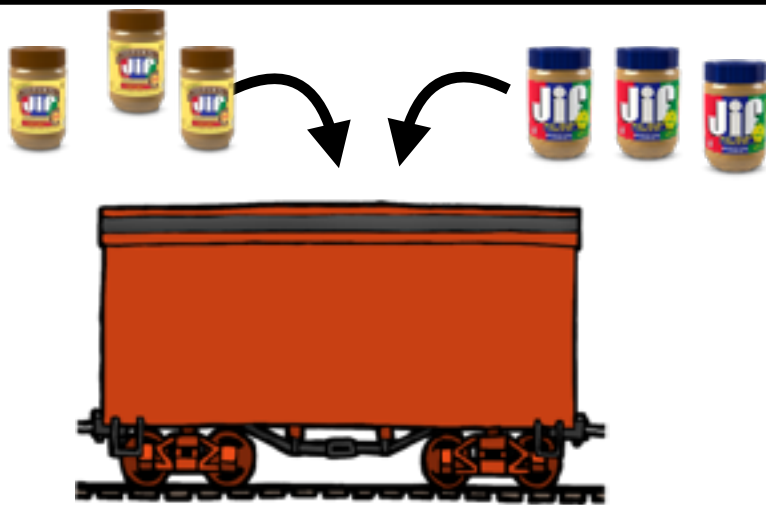


Your local machine

1. Initialize your repository
2. \$ git add

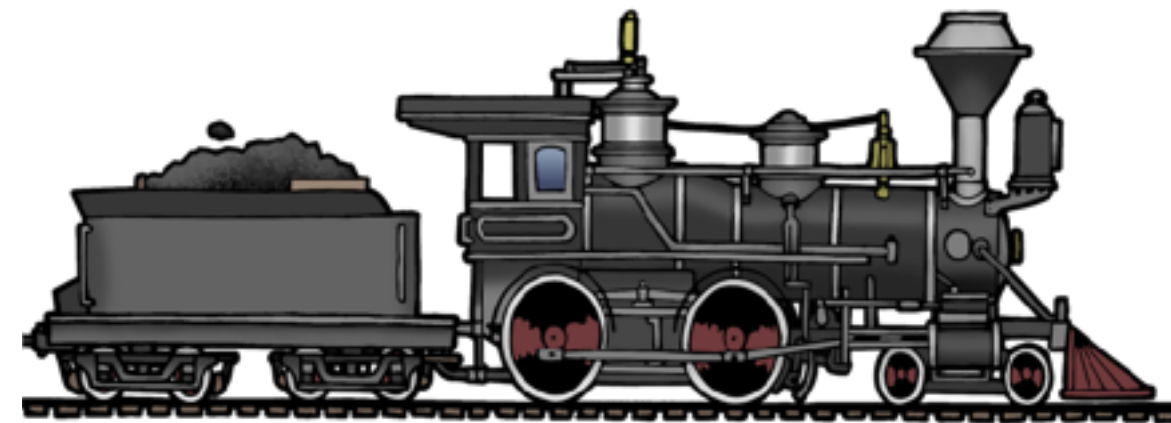


GitHub Repository



Your local machine

1. Initialize your repository
2. \$ git add
3. \$ git commit



GitHub Repository



Your local machine

1. Initialize your repository
2. \$ git add
3. \$ git commit
4. \$ git push



GitHub Repository

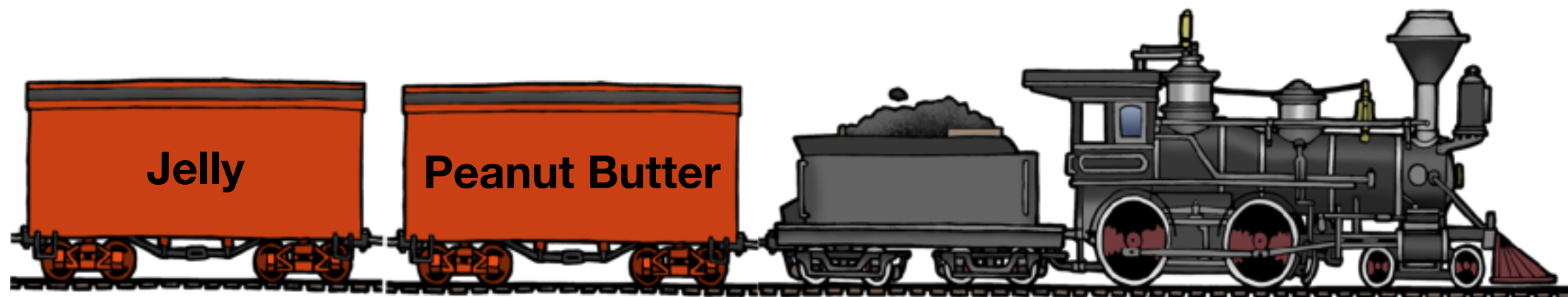


Your local machine

1. Initialize your repository
2. \$ git add
3. \$ git commit
4. \$ git push

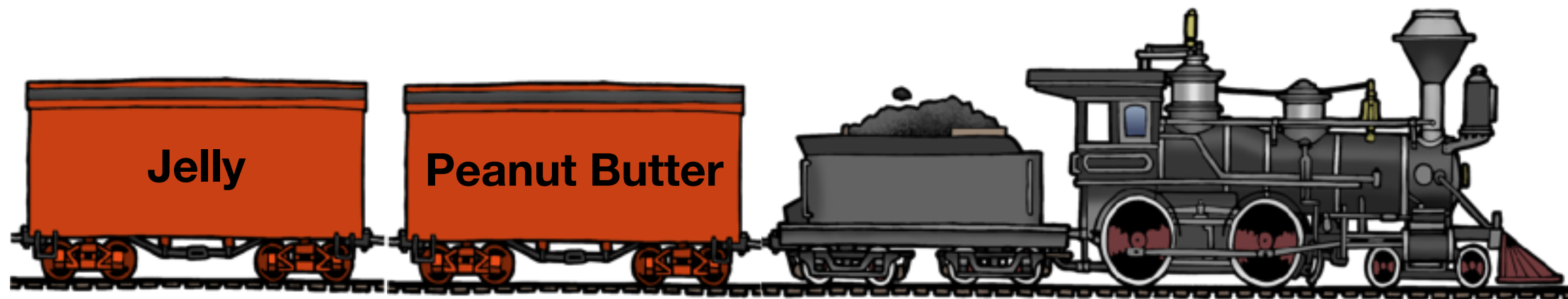


GitHub Repository

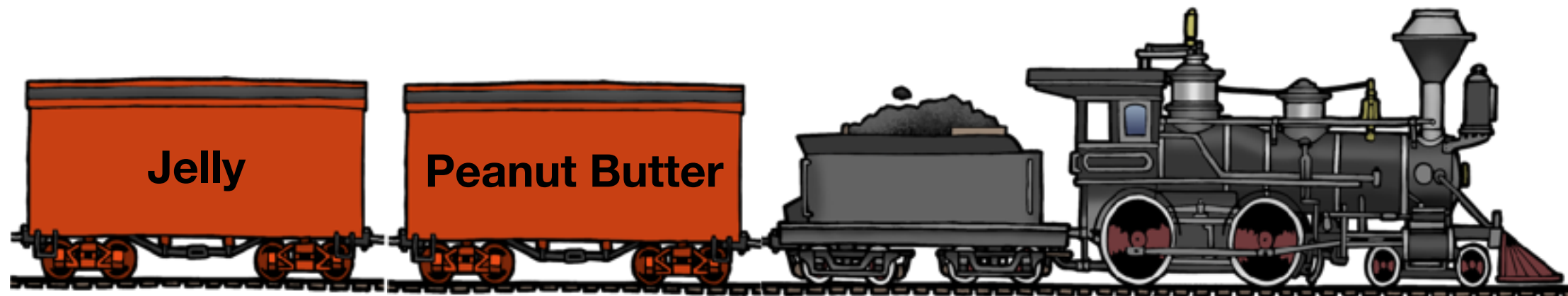


Your local machine

1. Initialize your repository
2. \$ git add
3. \$ git commit
4. \$ git push



GitHub Repository



Your local machine

Compartmentalized structure allows you to revert back to previous versions



GitHub Repository

most recent

oldest



Your local machine

Compartmentalized structure allows you to revert back to previous versions



GitHub Repository



Your local machine

Compartmentalized structure allows you to revert back to previous versions



GitHub Repository

most recent

oldest



Your local machine

“\$ git clone” makes it easy to change machines or share projects

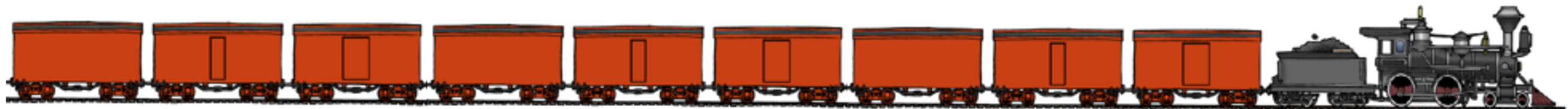


GitHub Repository

“\$ git clone” makes it easy to change machines or share projects



GitHub Repository



Your new computer



Your friends computer



AWS, TACC, other HPC

“\$ git pull” lets you update the code on one machine from another (different from cloning)



Your computer

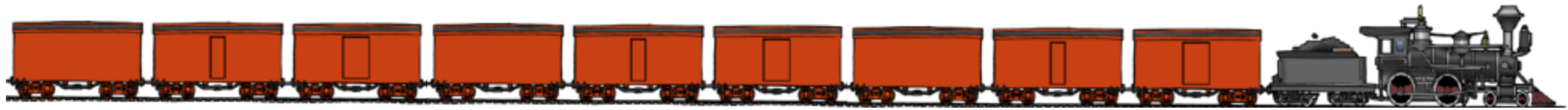


GitHub Repository



Your friend's computer

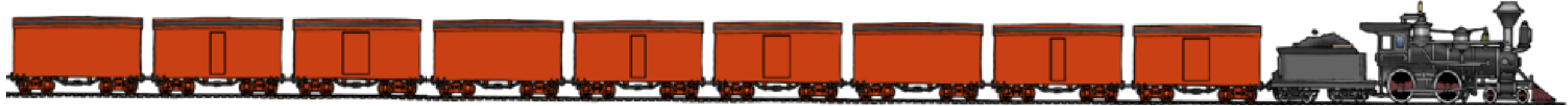
“\$ git pull” lets you update the code on one machine from another (different from cloning)



Your computer



git push

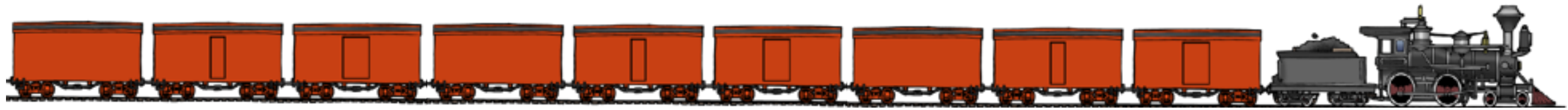


GitHub Repository



Your friend's computer

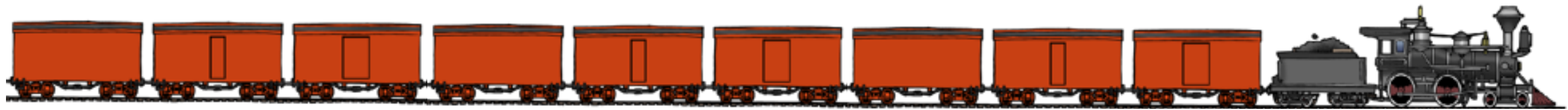
“\$ git pull” lets you update the code on one machine from another (different from cloning)



Your computer



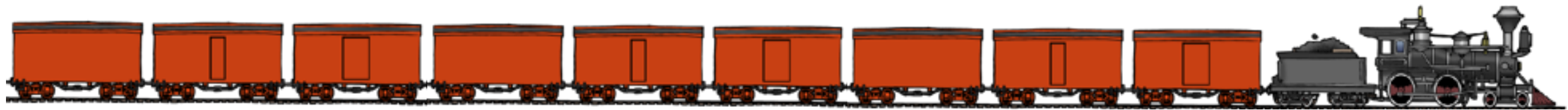
git push



GitHub Repository



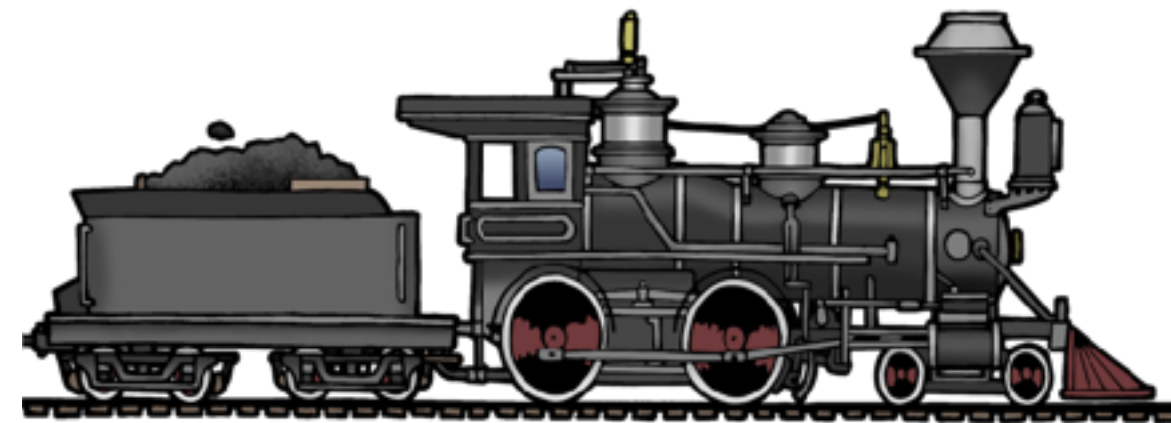
git pull



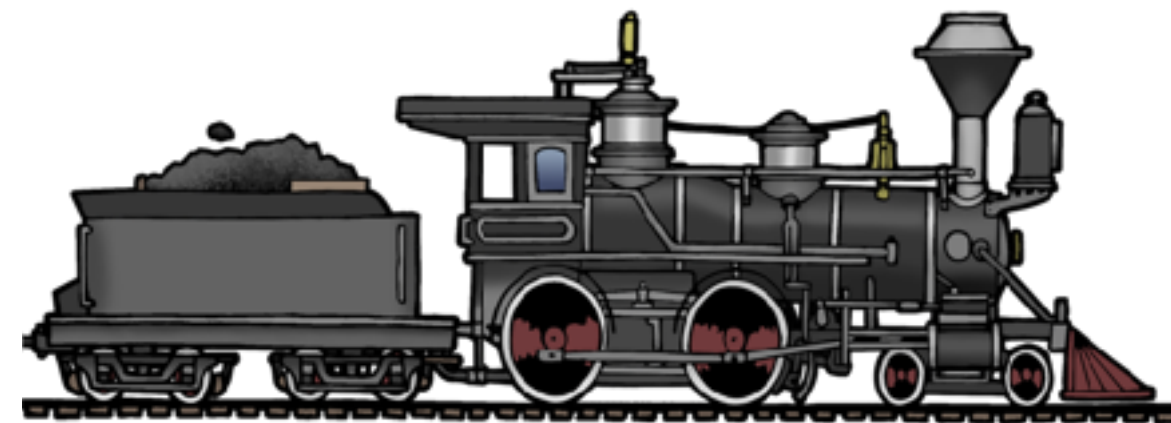
Your friend's computer

Questions?

\$ git status checks the state of a repository

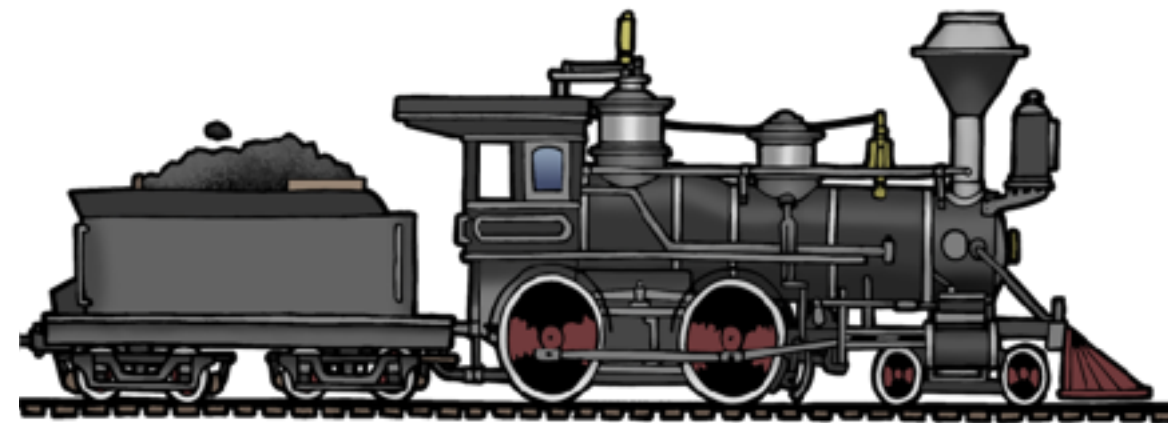


GitHub Repository

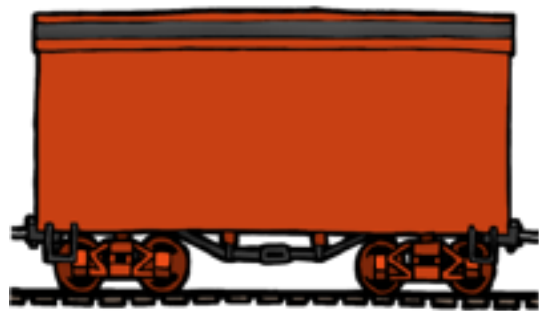


Your local machine

\$ git status checks the state of a repository

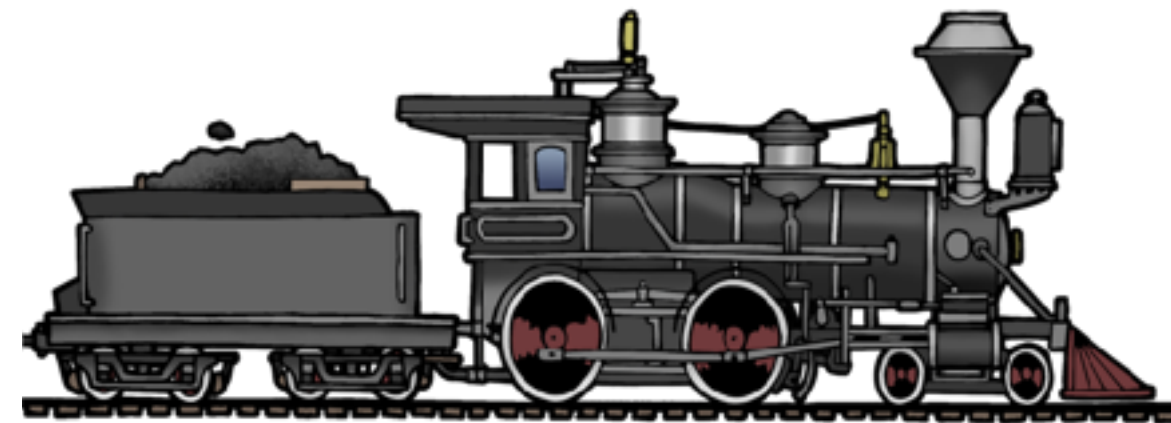


GitHub Repository

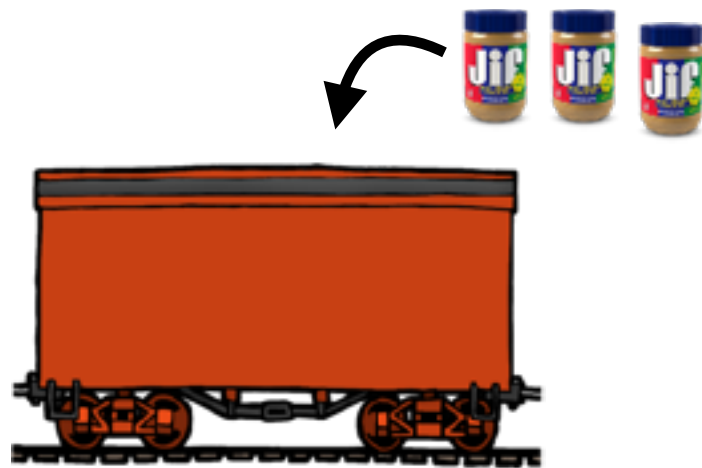


Your local machine

\$ git status checks the state of a repository

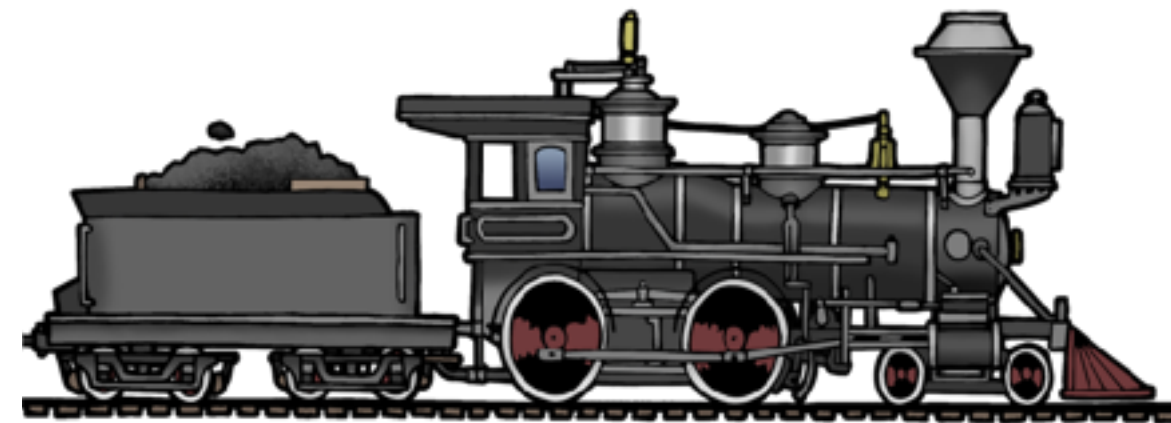


GitHub Repository

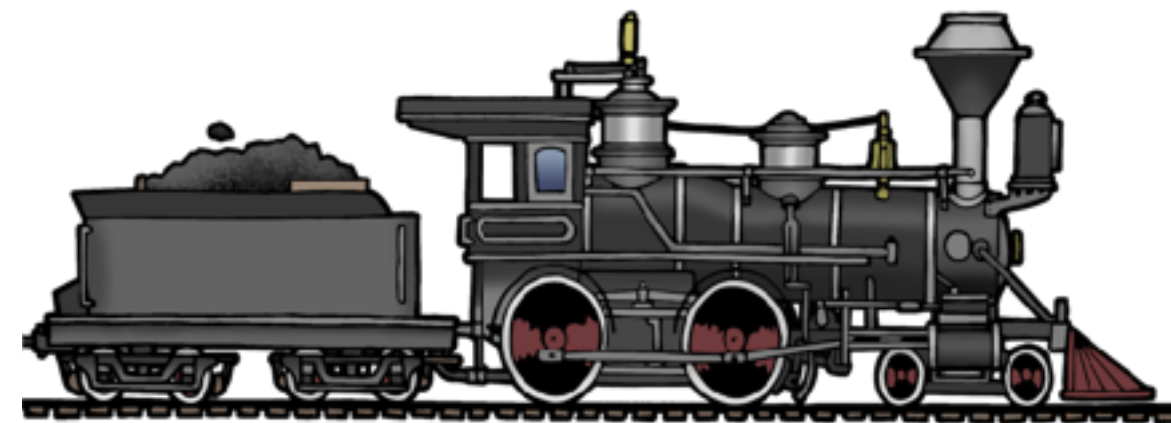
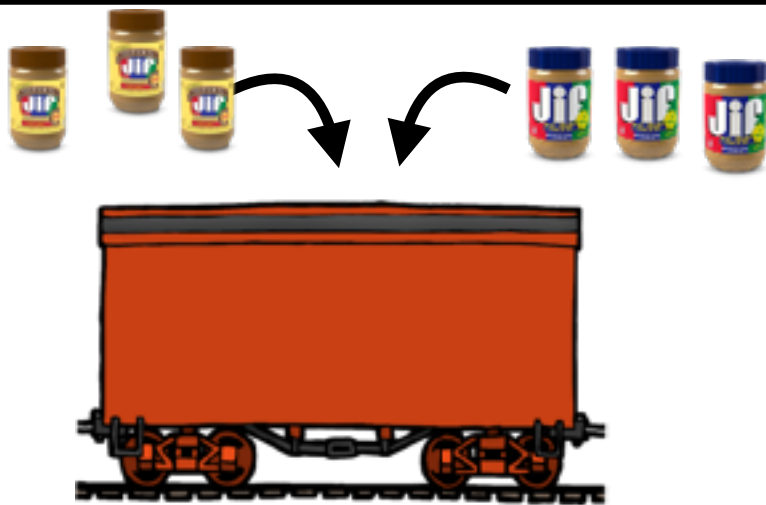


Your local machine

\$ git status checks the state of a repository

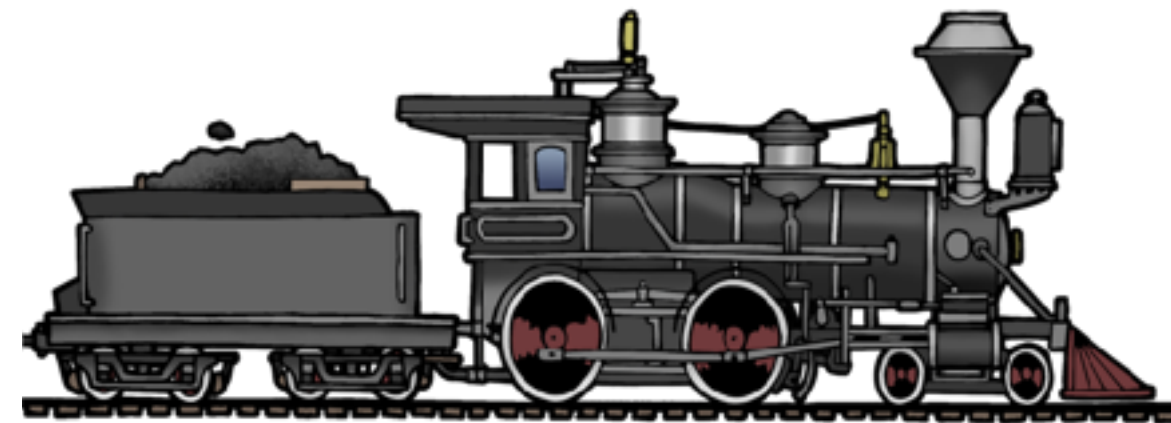


GitHub Repository



Your local machine

\$ git status checks the state of a repository



GitHub Repository



Your local machine

\$ git status checks the state of a repository



GitHub Repository

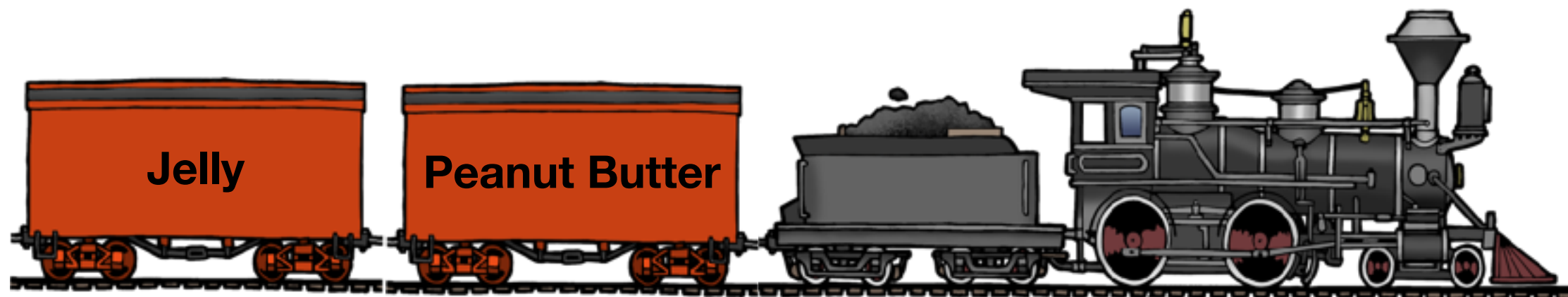


Your local machine

\$ git status checks the state of a repository

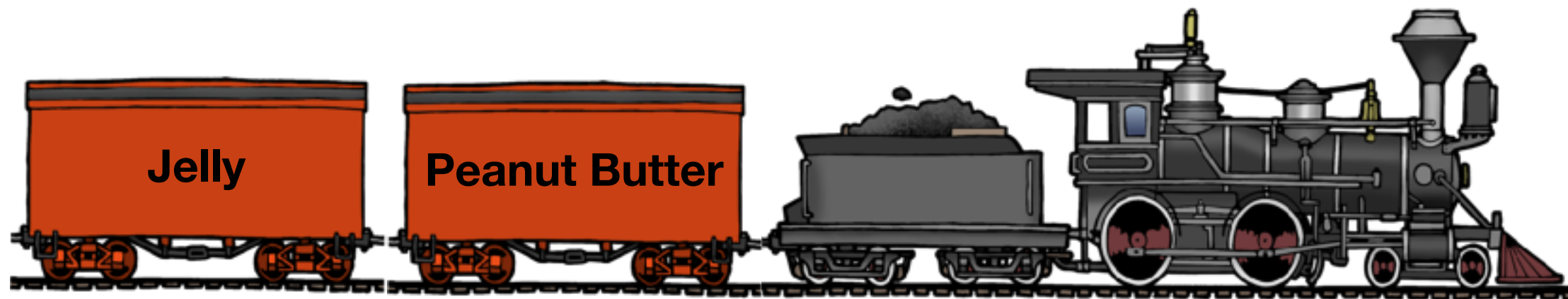


GitHub Repository

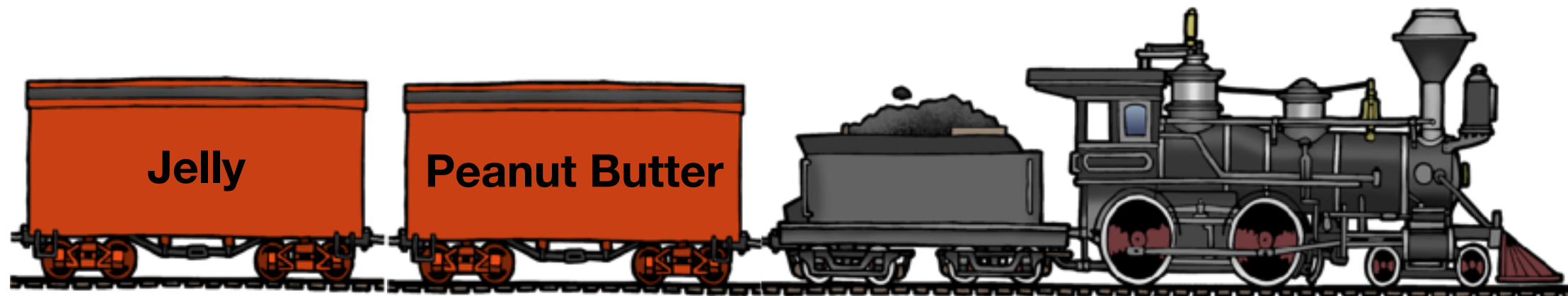


Your local machine

\$ git status checks the state of a repository



GitHub Repository



Your local machine

```
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   data/northern.csv
        modified:   report.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   data/northern.csv
        modified:   report.txt

no changes added to commit (use "git add" and/or "git commit -a")
$ git add report.txt
$ git status
```



```
$ git status
```

```
On branch master
```

```
Changes not staged for commit:
```

```
  (use "git add <file>..." to update what will be committed)
```

```
  (use "git checkout -- <file>..." to discard changes in working directory)
```

```
    modified:   data/northern.csv
```

```
    modified:   report.txt
```

```
no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git add report.txt
```

```
$ git status
```

```
On branch master
```

```
Changes to be committed:
```

```
  (use "git reset HEAD <file>..." to unstage)
```

```
    modified:   report.txt
```

```
Changes not staged for commit:
```

```
  (use "git add <file>..." to update what will be committed)
```

```
  (use "git checkout -- <file>..." to discard changes in working directory)
```

```
    modified:   data/northern.csv
```

Repo status

1. Go to Chapter 1, “How can I check the state of a repository?”
2. Spencer Demo (using terminal, .., and tab completion)
3. Work with your neighbors and follow through the exercise

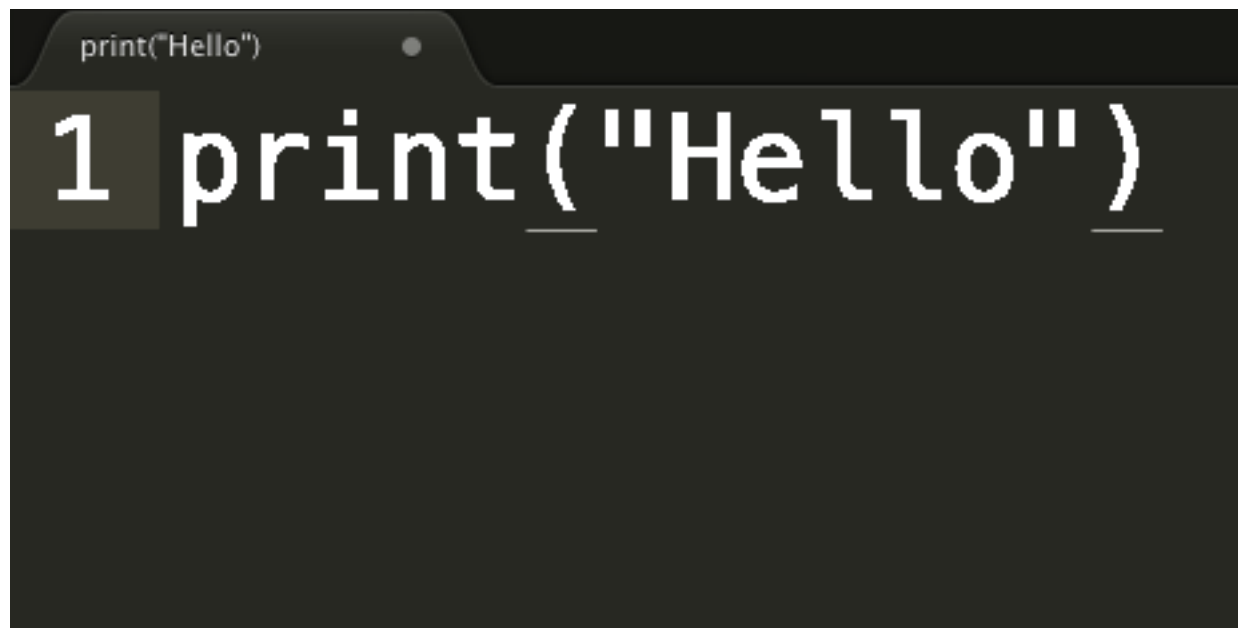
\$ git diff tells you what has changed

testfile.txt

\$ git diff tells you what has changed

testfile.txt

Previous State

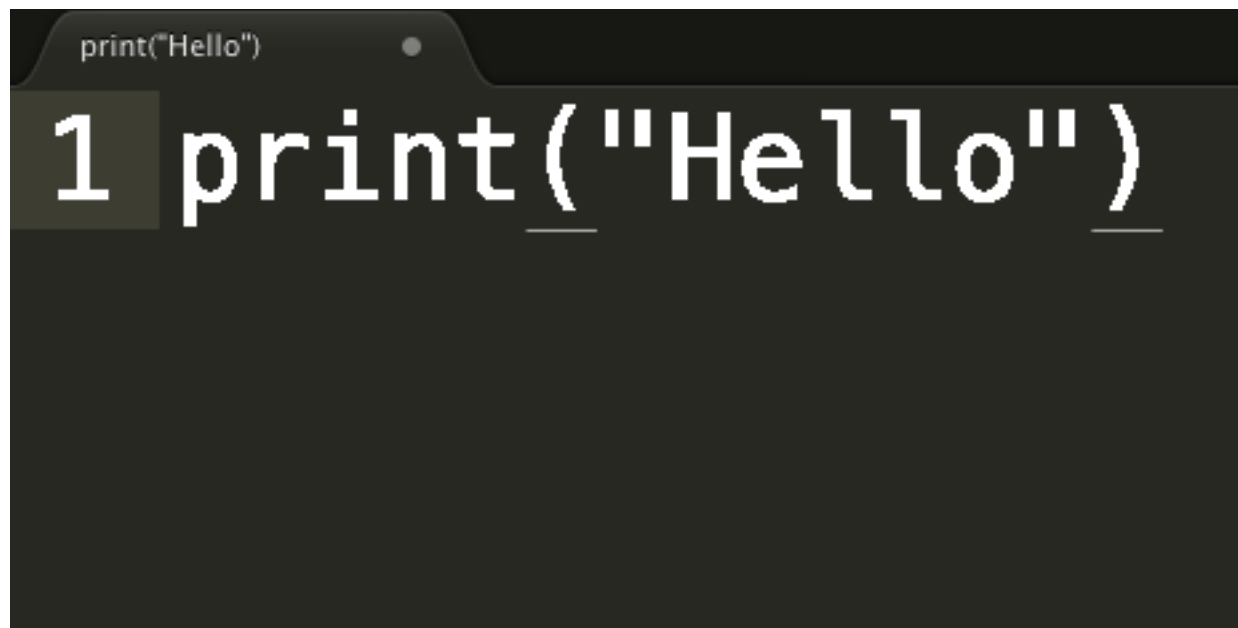


A screenshot of a code editor window. The title bar at the top reads "print('Hello')". The editor area shows a single line of Python code: `1 print("Hello")`. The line number "1" is in a light gray box on the left. The code is in a light blue monospace font on a dark background. There are small white underlines under the opening and closing quotes of the string "Hello".

\$ git diff tells you what has changed

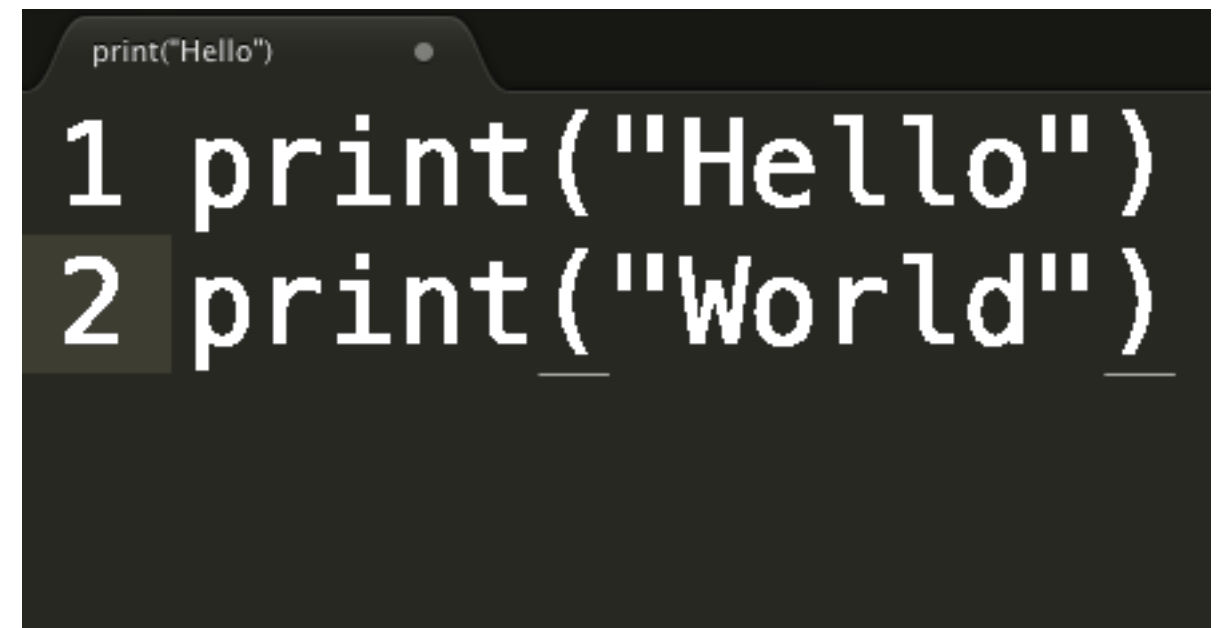
testfile.txt

Previous State



```
print("Hello")
1 print("Hello")
```

Current State

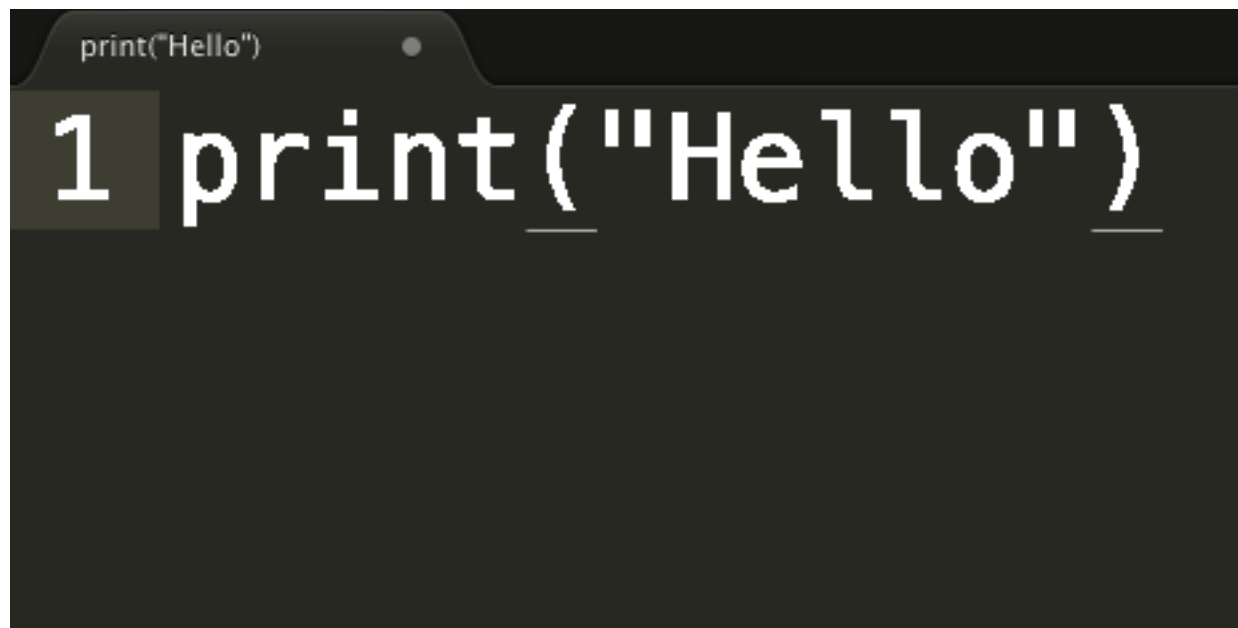


```
print("Hello")
1 print("Hello")
2 print("World")
```

\$ git diff tells you what has changed

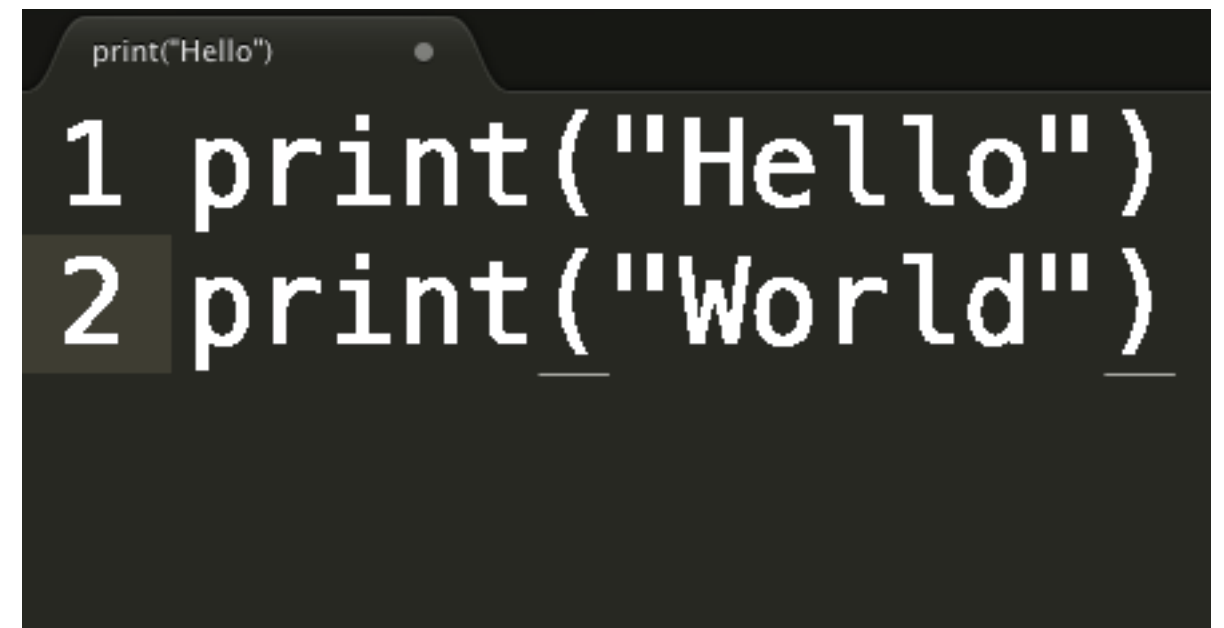
testfile.txt

Previous State



```
print("Hello")
1 print("Hello")
```

Current State



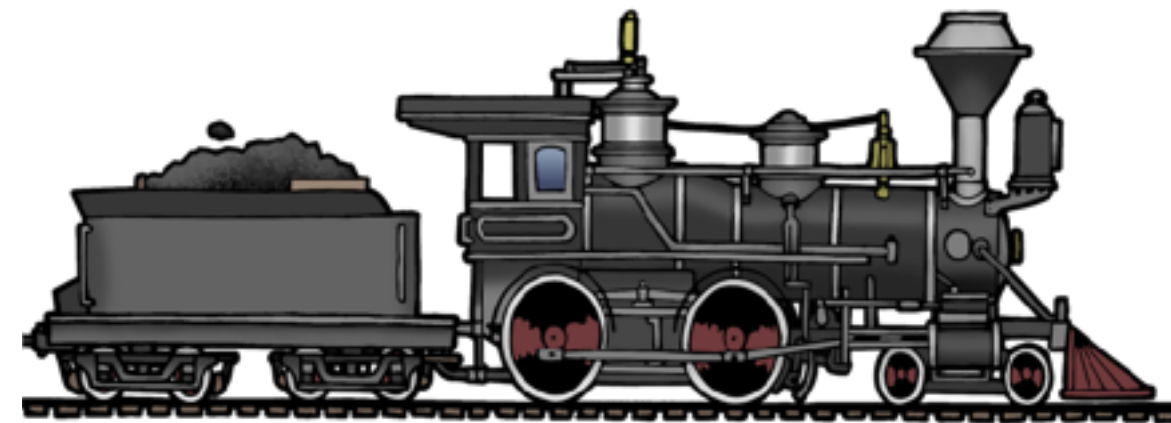
```
print("Hello")
1 print("Hello")
2 print("World")
```

What changed?

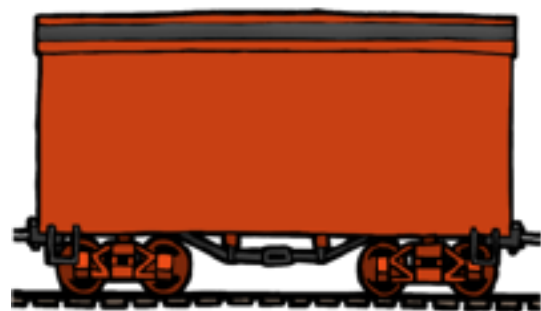
Understanding what changed

1. Go to Chapter 1, “How can I tell what I have changed?”
2. Work with your neighbors and follow through the exercise
3. If you have extra time, feel free to move on to the next exercise “What is in a diff?” to learn more

\$ git add gets files ready for a commit and allows tracking

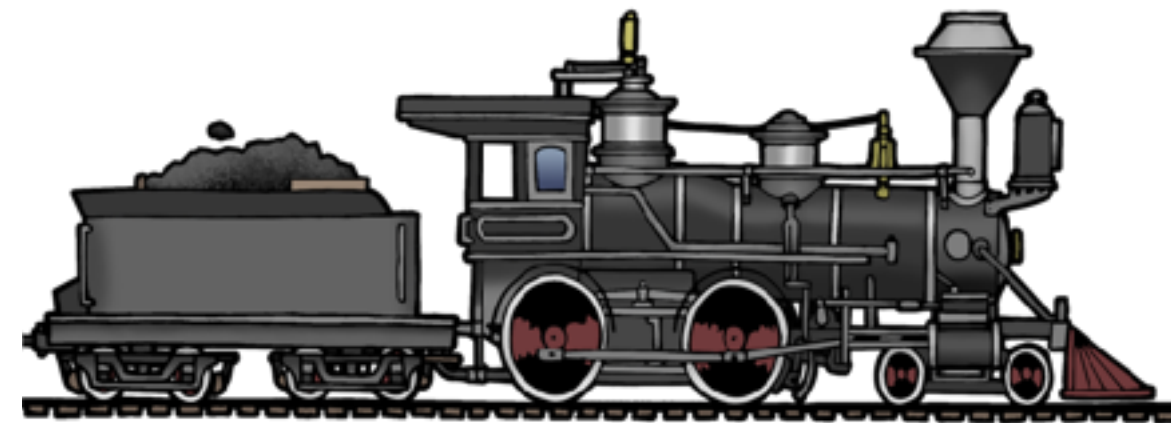


GitHub Repository

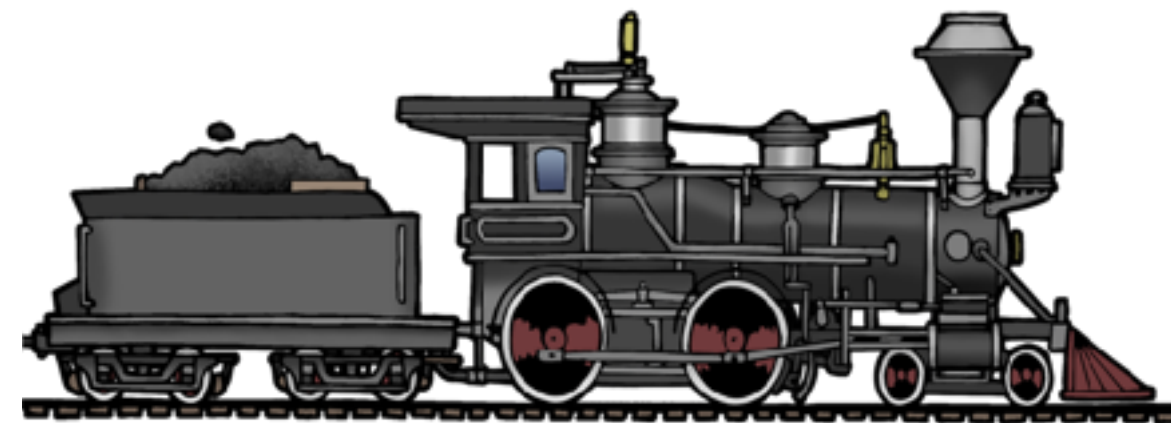
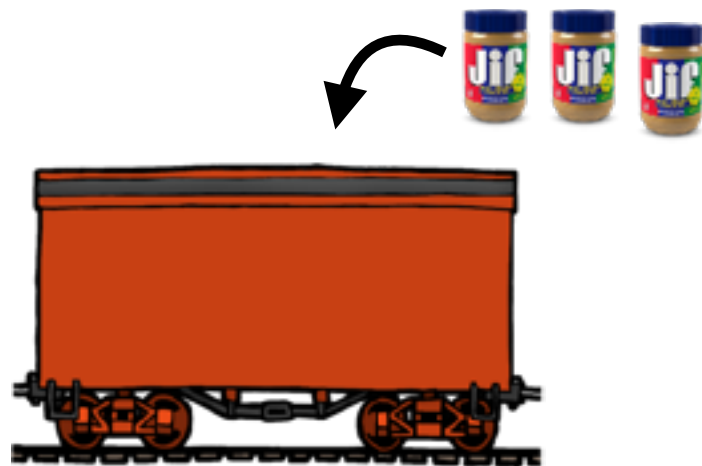


Your local machine

\$ git add gets files ready for a commit and allows tracking

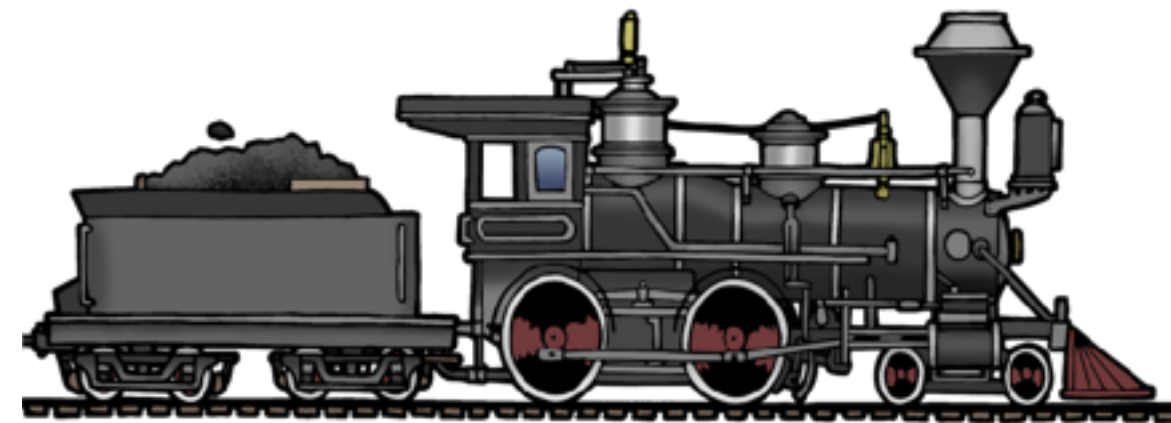


GitHub Repository

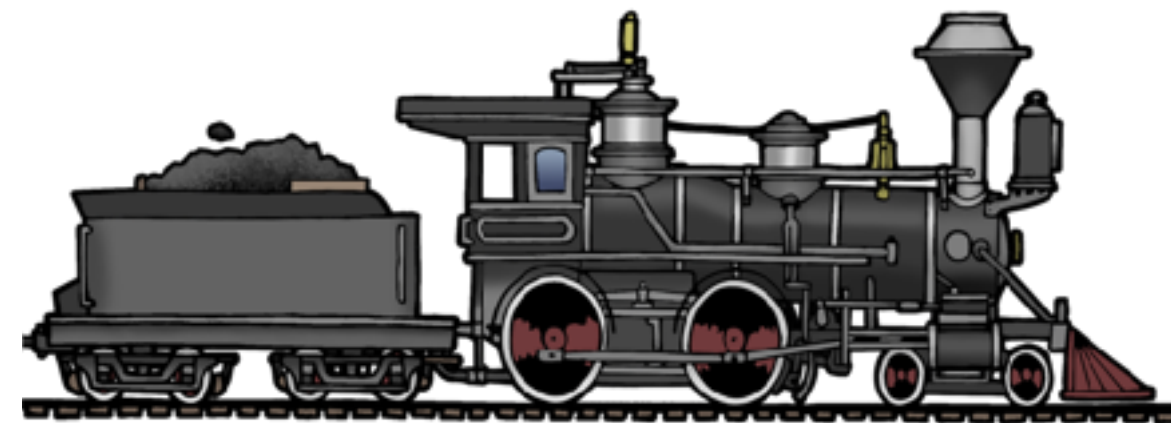
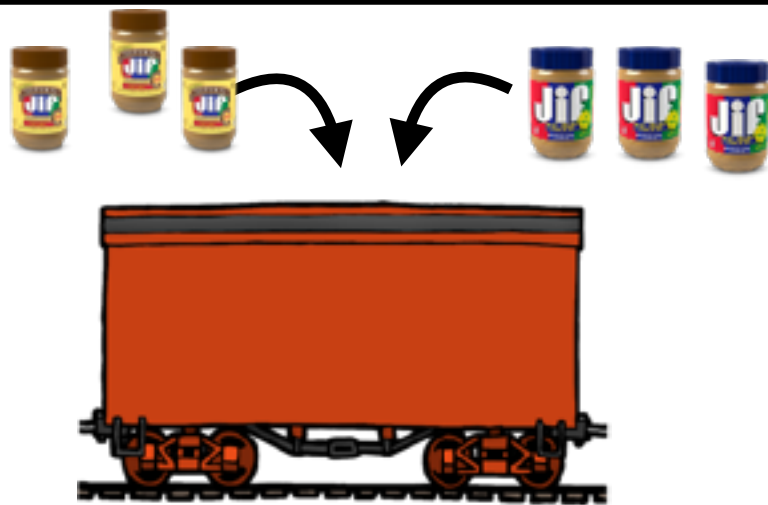


Your local machine

\$ git add gets files ready for a commit and allows tracking



GitHub Repository



Your local machine

\$ git add allows git to track the file

```
sf:zikaEstimatorR$ git status
On branch master
Your branch is up-to-date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   R/ms_fig_creator.R

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        deleted:    R/calc_monthly_rnts.R
        deleted:    R/r0_calc_fxns.R
        modified:   R/sim_test_likelihood_fxn.R

Untracked files:
  (use "git add <file>..." to include in what will be committed)

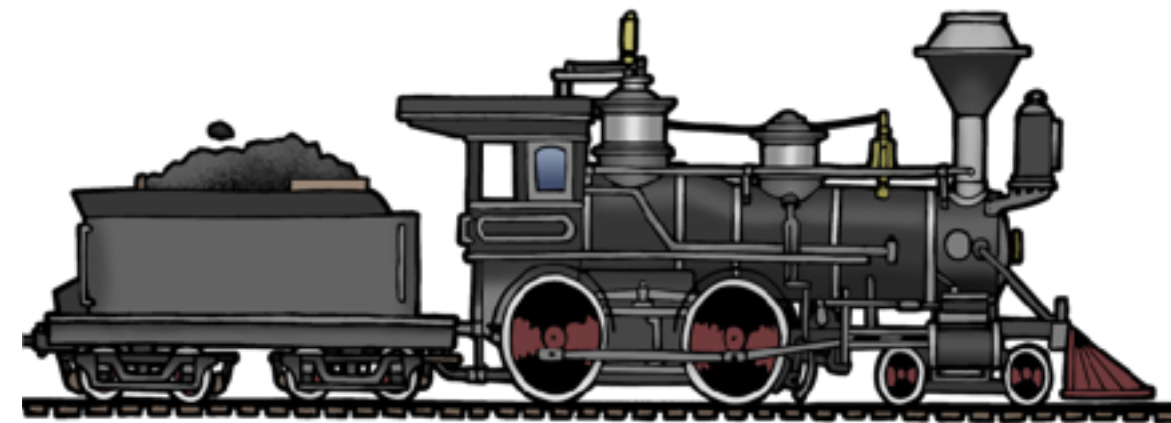
        R/lanl_figs.R
        R/plotting_fxns.R
```

Getting files ready to commit

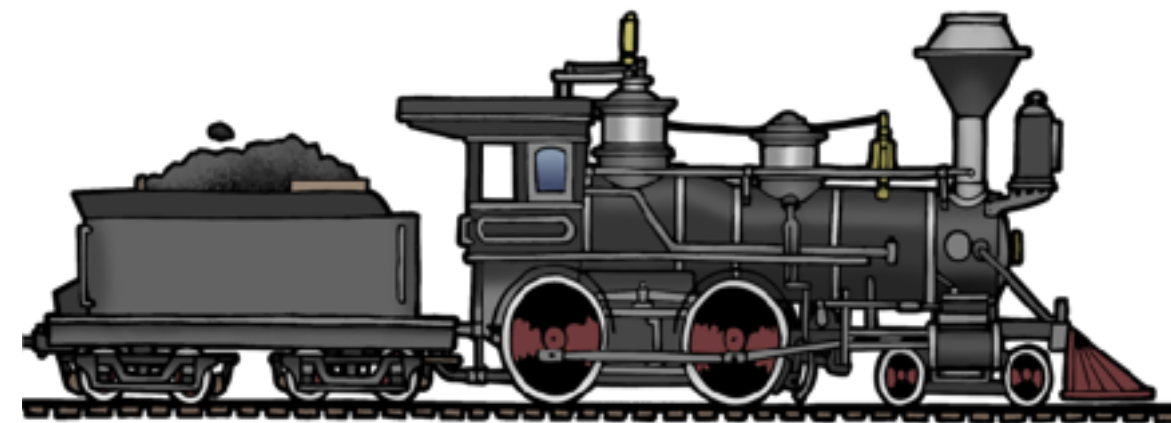
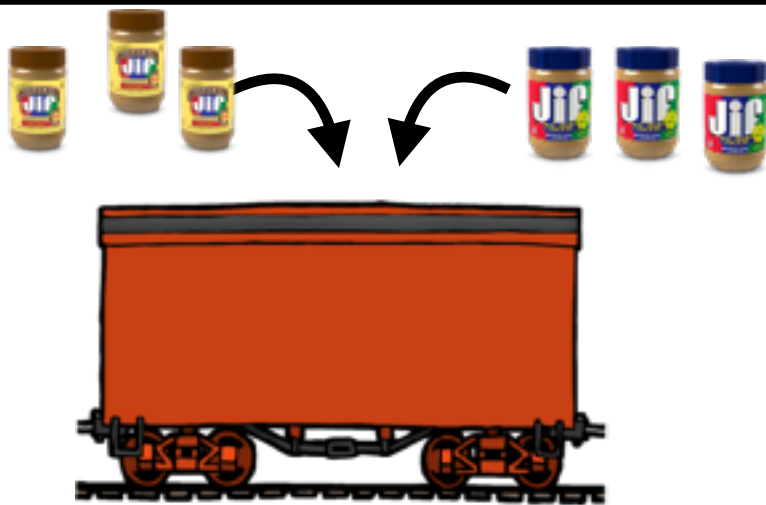
1. Go to Chapter 1, “What’s the first step in saving changes?”
2. Work with your neighbors and follow through the exercise
3. Move on to the next exercise if you have time

Hint: I usually start by using “\$ git status” to see what’s changed before adding

\$ git commit commits the added files to local repository

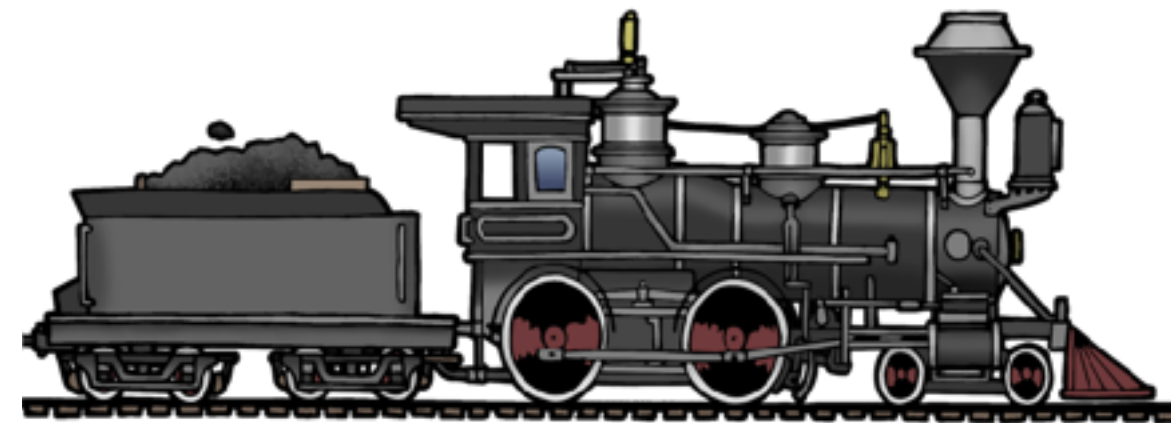


GitHub Repository



Your local machine

\$ git commit commits the added files to local repository



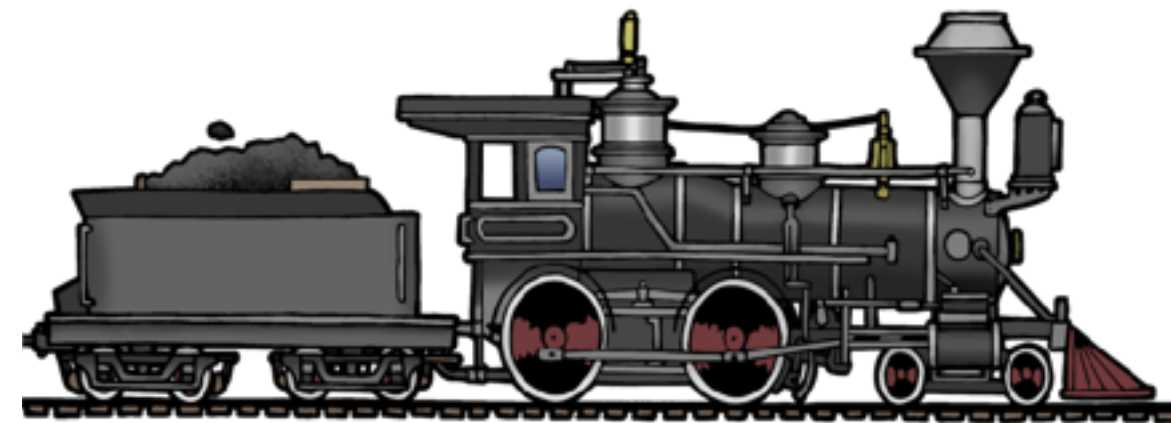
GitHub Repository



Your local machine

\$ git commit commits the added files to local repository

\$ git commit -m “Peanut Butter”



GitHub Repository



Your local machine

Committing changes to the repository

1. Go to Chapter 1, “How do I commit changes?”
2. Work with your neighbors and follow through the exercise
3. Check out the “Interlude: how can I edit a file” if you have time still

Questions?

Putting it all together

1. Go to Chapter 2, “How do I add new files?”
2. Work with your neighbors and follow through the exercise
3. Try Chapter 5, “How can I push my changes to a remote repository” if you have time - I haven’t taught everything needed for this though

Cloning (downloading) a repository

Cloning (downloading) a repository

- 1. Find repository online**
- 2. Copy the cloning link**
- 3. Navigate to where you want folder to download**
- 4. git clone**

Cloning (downloading) a repository

1. Find repository online
2. Copy the cloning link
3. Navigate to where you want folder to download
4. git clone

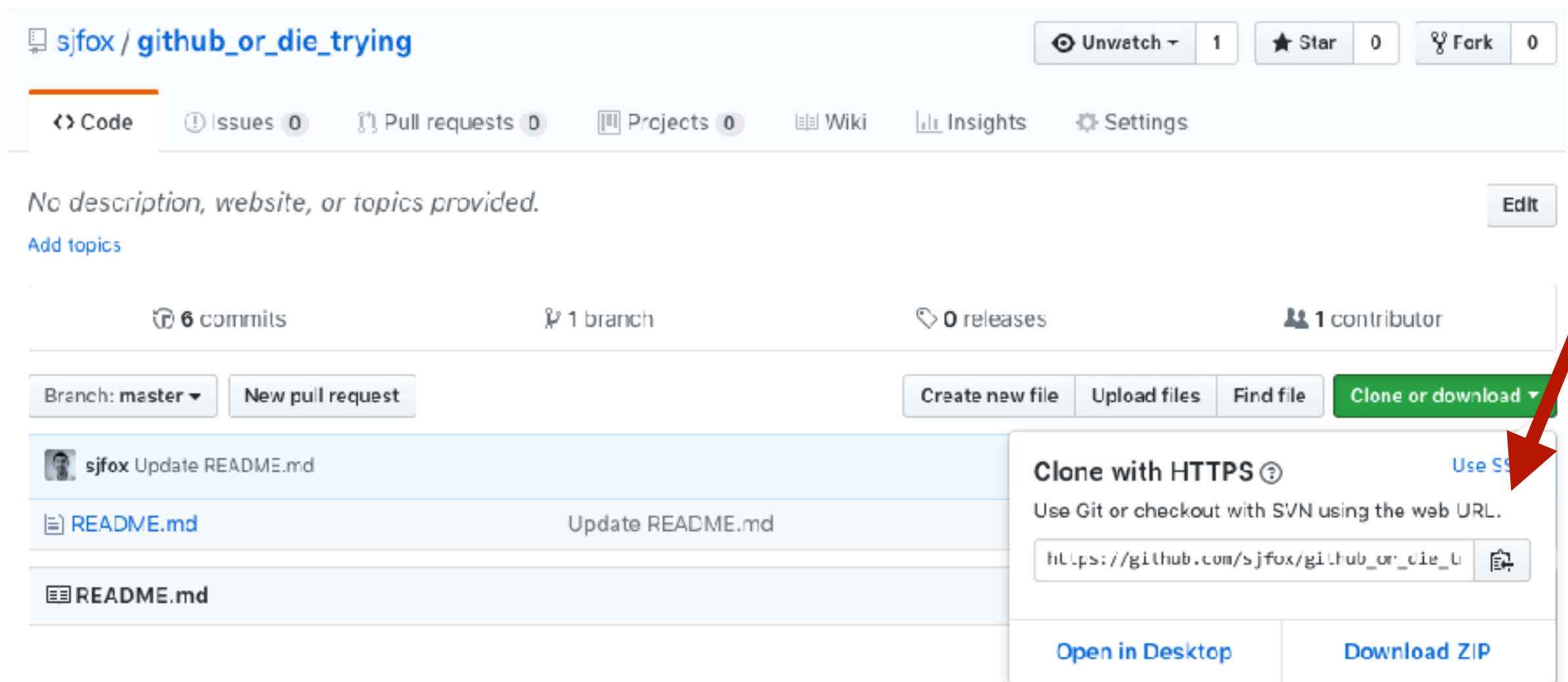
The screenshot shows the GitHub interface for the repository 'sjfox / github_or_die_trying'. At the top, there are buttons for 'Unwatch' (1), 'Star' (0), and 'Fork' (0). Below this is a navigation bar with links for 'Code', 'Issues' (0), 'Pull requests' (0), 'Projects' (0), 'Wiki', 'Insights', and 'Settings'. The repository description states 'No description, website, or topics provided.' with an 'Edit' button. Below the description, it shows '6 commits', '1 branch', '0 releases', and '1 contributor'. A bar at the bottom contains a 'Branch: master' dropdown, a 'New pull request' button, and buttons for 'Create new file', 'Upload files', 'Find file', and a green 'Clone or download' button. The 'Clone or download' button is open, showing a dropdown menu with the following options: 'Clone with HTTPS' (with a help icon), 'Use SSH', a text input field containing the URL 'https://github.com/sjfox/github_or_die_t...' with a copy icon, 'Open in Desktop', and 'Download ZIP'. The repository file list shows 'Update README.md' by 'sjfox' and 'README.md'.

Github or Die Tryin'

This is a repository that holds the materials from the introduction to github class on March 23, 2016. The session was presented as part of the [data and donuts seminar series](#), and contains the slides from the talk.

Cloning (downloading) a repository

1. Find repository online
2. Copy the cloning link
3. Navigate to where you want folder to download
4. git clone



The screenshot shows the GitHub interface for the repository 'sjfox / github_or_die_trying'. At the top, there are buttons for 'Unwatch' (1), 'Star' (0), and 'Fork' (0). Below these are tabs for 'Code', 'Issues' (0), 'Pull requests' (0), 'Projects' (0), 'Wiki', 'Insights', and 'Settings'. A message states 'No description, website, or topics provided.' with an 'Edit' button. Below this, statistics show '6 commits', '1 branch', '0 releases', and '1 contributor'. A row of buttons includes 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and a green 'Clone or download' button. A red arrow points to this button, which has a dropdown menu open. The dropdown menu has the title 'Clone with HTTPS' and the text 'Use Git or checkout with SVN using the web URL.' It contains a text input field with the URL 'https://github.com/sjfox/github_or_die_trying' and a copy icon. At the bottom of the dropdown are two buttons: 'Open in Desktop' and 'Download ZIP'.

sjfox / github_or_die_trying

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

No description, website, or topics provided. Edit

Add topics

6 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

Clone with HTTPS ⓘ Use SSH

Use Git or checkout with SVN using the web URL.

https://github.com/sjfox/github_or_die_trying

Open in Desktop Download ZIP

Github or Die Tryin'

This is a repository that holds the materials from the introduction to github class on March 23, 2016. The session was presented as part of the [data and donuts seminar series](#), and contains the slides from the talk.

Cloning (downloading) a repository

1. Find repository online
2. Copy the cloning link
3. Navigate to where you want folder to download
4. git clone

```
sf:projects$ git clone https://github.com/sjfox/github_or_die_trying.git
Cloning into 'github_or_die_trying'...
remote: Counting objects: 24, done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 24 (delta 6), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (24/24), done.
sf:projects$ █
```

Cloning a repository

1. Go to Chapter 5, “How can I create a copy of an existing repository?”
2. First - ignore the instructions - and try to clone my GitHub repository into the terminal (https://github.com/sjfox/github_or_die_trying/)
3. Try to follow their instructions if you have time - work with your neighbors!

Finally, git provides a systematic way to resolve conflicts during collaboration



Your collaborator



GitHub Repository



Your computer

Finally, git provides a systematic way to resolve conflicts during collaboration

Ketchup



Your collaborator



GitHub Repository

Hot dogs



Your computer

Finally, git provides a systematic way to resolve conflicts during collaboration

Ketchup



Your collaborator

git push



GitHub Repository

Hot dogs



Your computer

Finally, git provides a systematic way to resolve conflicts during collaboration

Ketchup



Your collaborator

git push



GitHub Repository

git push

Hot dogs



Your computer

Finally, git provides a systematic way to resolve conflicts during collaboration

Ketchup



Your collaborator

git push



GitHub Repository

~~git push~~

Hot dogs



Your computer

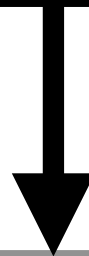
Finally, git provides a systematic way to resolve conflicts during collaboration

Ketchup



Your collaborator

git push



GitHub Repository

~~git push~~



Hot dogs w/ mustard



Your computer

Finally, git provides a systematic way to resolve conflicts during collaboration

Ketchup



Your collaborator

git push

Hot dogs w/ mustard



GitHub Repository

git push

Hot dogs w/ mustard



Your computer

Finally, git provides a systematic way to resolve conflicts during collaboration

Hot dogs w/ mustard



Your collaborator



git push



git pull

Hot dogs w/ mustard



GitHub Repository



git push

Hot dogs w/ mustard



Your computer

Summary

Summary

- Hopefully you've learned a bit about command line and git/hub!

Summary

- Hopefully you've learned a bit about command line and git/hub!
- Start your next project in GitHub - even if it's solo

Summary

- Hopefully you've learned a bit about command line and git/hub!
- Start your next project in GitHub - even if it's solo
- Three basic commands are all that's necessary to start
 - git add
 - git commit
 - git push

Resources

- Jenny Bryan is always an amazing resource:
 - <http://happygitwithr.com/>
- datacamp
- Pro Git book available online
 - <https://git-scm.com/book/en/v2>
- GitHub guides are also really helpful
 - <https://guides.github.com/>

 spncrfx@gmail.com

 @foxandtheflu

Resources

- Jenny Bryan is always an amazing resource:
 - <http://happygitwithr.com/>
- datacamp
- Pro Git book available online
 - <https://git-scm.com/book/en/v2>
- GitHub guides are also really helpful
 - <https://guides.github.com/>



 spncrfx@gmail.com

 @foxandtheflu