An introduction to git

Ryan Bignell

~

S4S Meeting
3pm June 12th 2020

Quote

"Git is a distributed version-control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed, data integrity, and support for distributed, non-linear workflows"

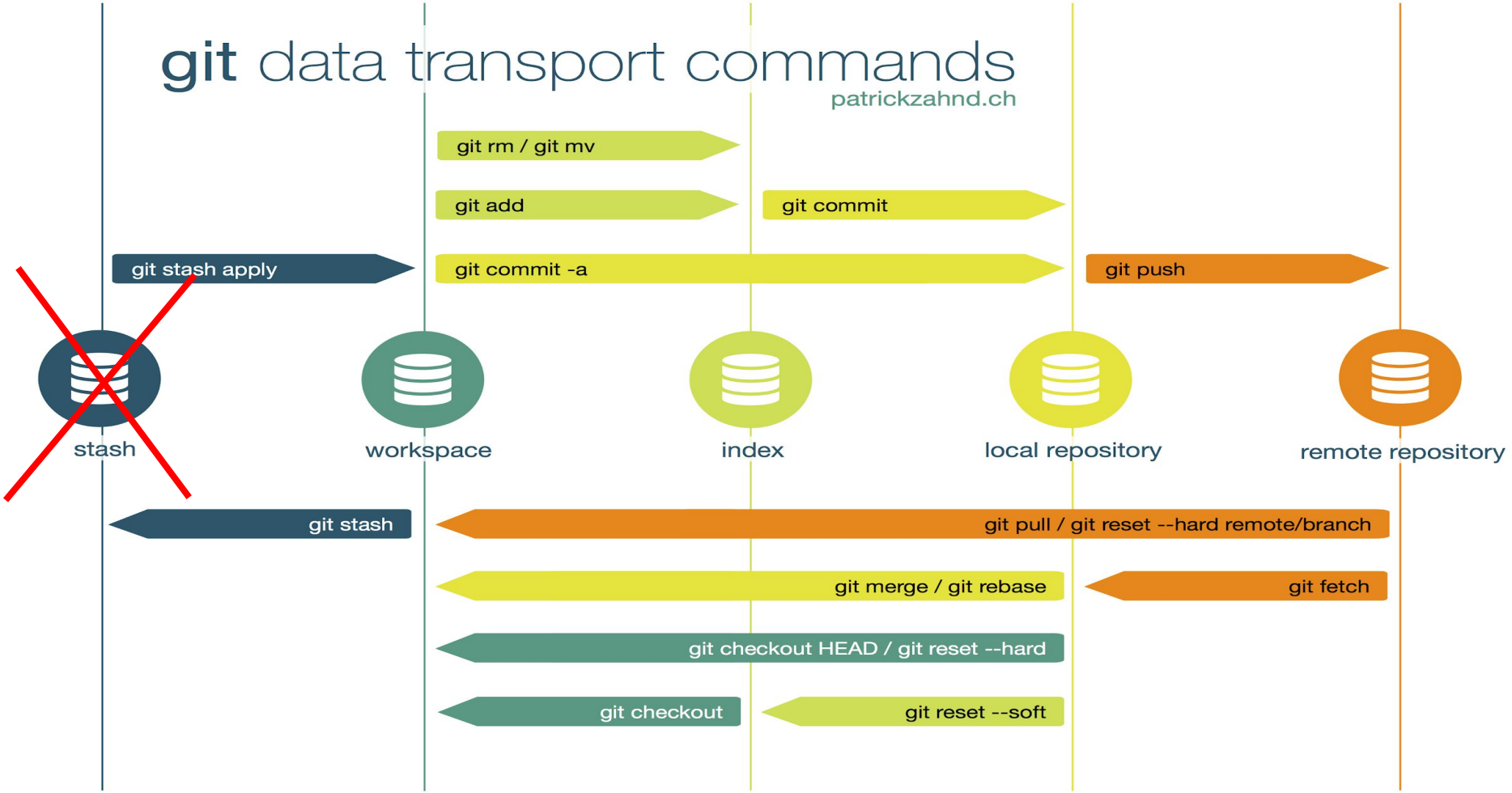
Dot points

- Version control system
- External backup solution
- Collaborative design environment

This presentation will be based on Albert's guide and also https://patrickzahnd.ch/blog.html#gittransport

git data transport commands

patrickzahnd.ch



Commands

Start a new repository in an empty folder using

git init

See the status of the git repository using

git status

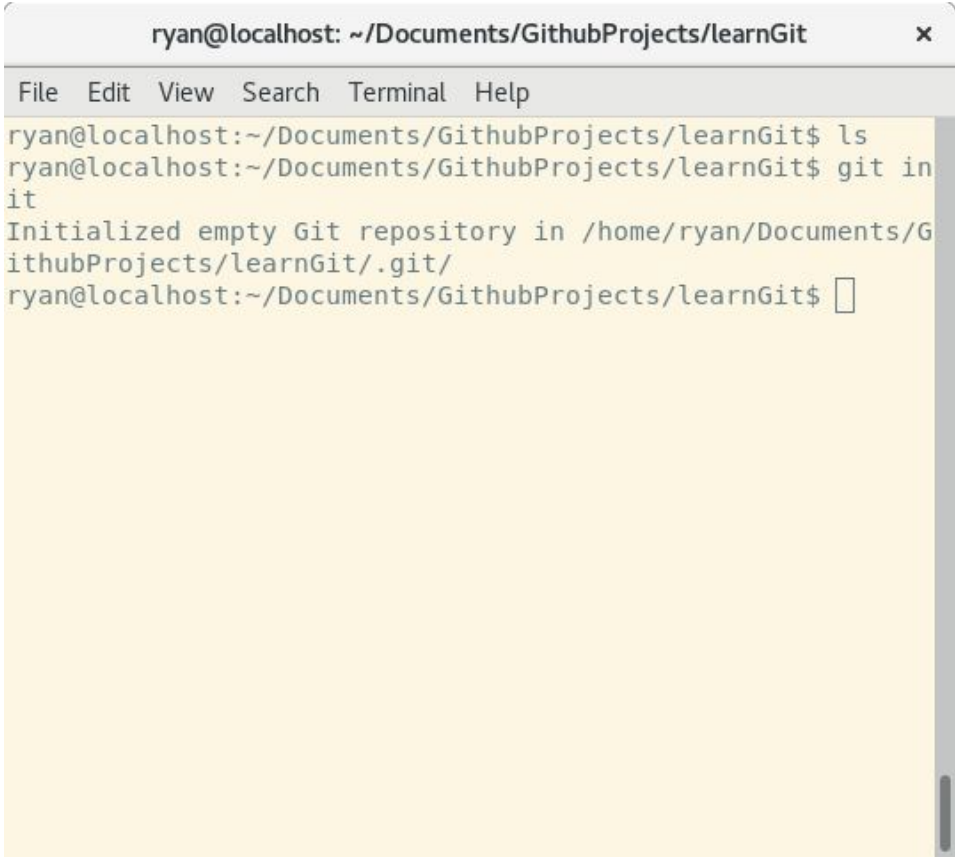
Commands

Start a new repository in an empty folder using

git init

See the status of the git repository using

git status



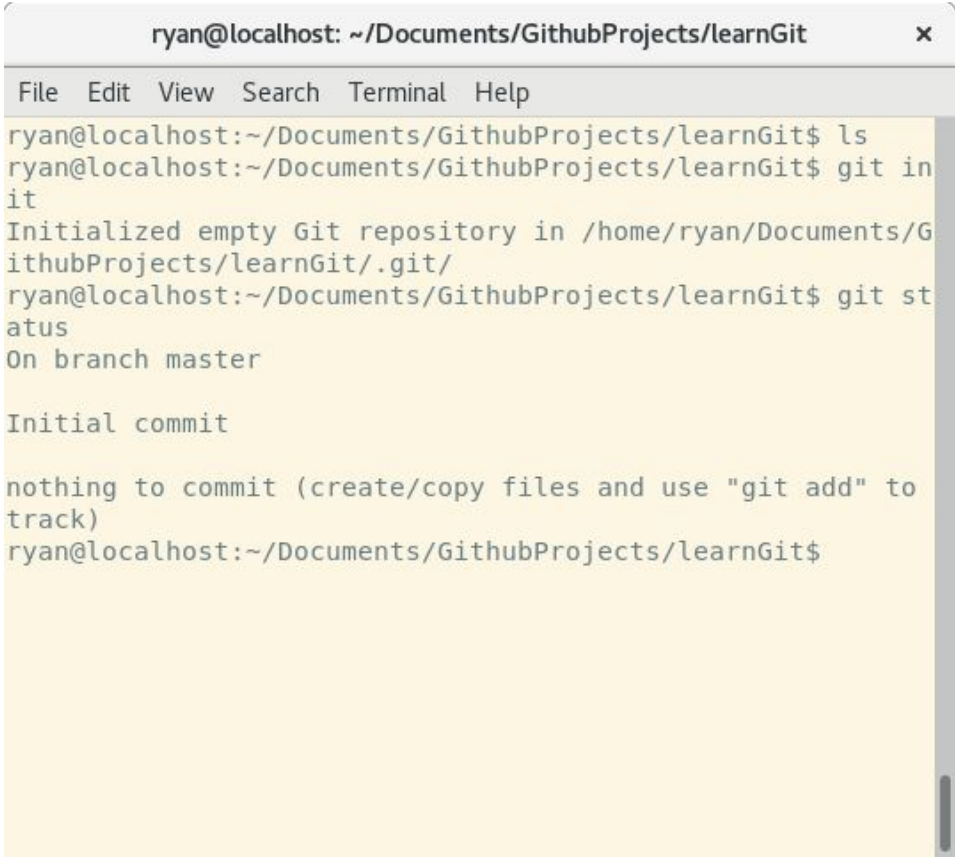
Commands

Start a new repository in an empty folder using

git init

See the status of the git repository using

git status



Commands

Copied a file into our folder

Albert-git-doc.pdf

This item is in the workspace

ryan@localhost: ~/Documents/GithubProjects/learnGit

File Edit View Search Terminal Help

```
ryan@localhost:~/Documents/GithubProjects/learnGit$ ls
Albert-git-doc.pdf
ryan@localhost:~/Documents/GithubProjects/learnGit$
```

Commands

Copied a file into our folder

Albert-git-doc.pdf

This item is in the workspace


```
ryan@localhost: ~/Documents/GithubProjects/learnGit
File  Edit  View  Search  Terminal  Help
ryan@localhost:~/Documents/GithubProjects/learnGit$ ls
Albert-git-doc.pdf
ryan@localhost:~/Documents/GithubProjects/learnGit$ git st
atus
On branch master

Initial commit

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

        Albert-git-doc.pdf

nothing added to commit but untracked files present (use "
git add" to track)
ryan@localhost:~/Documents/GithubProjects/learnGit$
```

Commands

Add the file to the index using

git add Albert-git-doc.pdf

A snapshot of this file is now in the index

ryan@localhost: ~/Documents/GithubProjects/learnGit

File Edit View Search Terminal Help

```
ryan@localhost:~/Documents/GithubProjects/learnGit$ git ad
d Albert-git-doc.pdf
ryan@localhost:~/Documents/GithubProjects/learnGit$ git st
atus
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   Albert-git-doc.pdf

ryan@localhost:~/Documents/GithubProjects/learnGit$
```

Commands

What if I change Albert-git-doc.pdf ?

echo 55 > Albert-git-doc.pdf

git status

Change in workspace but the snapshot in index is unchanged


```
ryan@localhost: ~/Documents/GithubProjects/learnGit
File  Edit  View  Search  Terminal  Help
ryan@localhost:~/Documents/GithubProjects/learnGit$ echo 5
5 > Albert-git-doc.pdf
ryan@localhost:~/Documents/GithubProjects/learnGit$ git st
atus
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   Albert-git-doc.pdf

Changes not staged for commit:
  (use "git add <file>..." to update what will be committe
d)
  (use "git checkout -- <file>..." to discard changes in w
orking directory)

        modified:   Albert-git-doc.pdf

ryan@localhost:~/Documents/GithubProjects/learnGit$
```

Commands

Ok echo 55 is pretty dumb, let's get rid of that change and go back to the version in the we snapshotted

git checkout Albert-git-doc.pdf

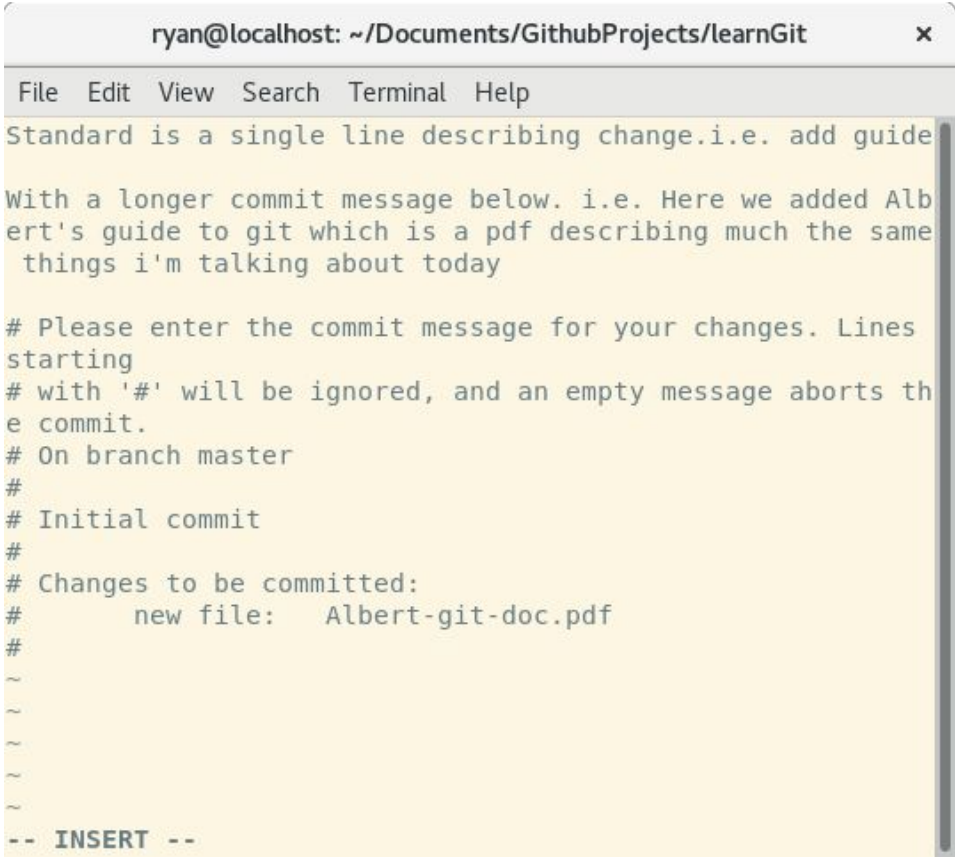
git status


```
ryan@localhost: ~/Documents/GithubProjects/learnGit
File  Edit  View  Search  Terminal  Help
ryan@localhost:~/Documents/GithubProjects/learnGit$ git ch
eckout Albert-git-doc.pdf
ryan@localhost:~/Documents/GithubProjects/learnGit$ git st
atus
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   Albert-git-doc.pdf

ryan@localhost:~/Documents/GithubProjects/learnGit$
```

Commands

Let's commit this change. I.e. We're finished what we were doing with Albert-git-doc.pdf

git commit



Commands

Let's commit this change. I.e. We're finished what we were doing with Albert-git-doc.pdf

git commit

File is now stored in the local repository.


```
ryan@localhost: ~/Documents/GithubProjects/learnGit
File  Edit  View  Search  Terminal  Help
ryan@localhost:~/Documents/GithubProjects/learnGit$ git co
mmit
[master (root-commit) 4465f19] Standard is a single line d
escribing change.i.e. add guide
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Albert-git-doc.pdf
ryan@localhost:~/Documents/GithubProjects/learnGit$
```

Commands

What have we done so far?

Created a repository

Added a file to the repository so that git knows about it.

Committed the state of that file to the repository.

git log


```
ryan@localhost: ~/Documents/GithubProjects/learnGit
File  Edit  View  Search  Terminal  Help
ryan@localhost:~/Documents/GithubProjects/learnGit$ git log
commit 4465f19363e65d6051f8326a66629d4c858dfc53
Author: Ryan Bignell <Ryan.Bignell@adelaide.edu.au>
Date:   Fri Jun 5 09:22:37 2020 +0930

    Standard is a single line describing change.i.e. add guide

    With a longer commit message below. i.e. Here we added Albert's guide to git which is a pdf describing much the same things i'm talking about today
ryan@localhost:~/Documents/GithubProjects/learnGit$
```

Commands

When you add a file to git using `git add` you tell git you care about this file and want modifications to this file to be tracked

touch blah.txt

git status

blah.txt is untracked as we haven't added it. It's still in our workspace though


```
ryan@localhost: ~/Documents/GithubProjects/learnGit
File  Edit  View  Search  Terminal  Help
ryan@localhost:~/Documents/GithubProjects/learnGit$ touch
blah.txt
ryan@localhost:~/Documents/GithubProjects/learnGit$ git st
atus
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be comm
itted)

        blah.txt

nothing added to commit but untracked files present (use "
git add" to track)
ryan@localhost:~/Documents/GithubProjects/learnGit$
```

Commands

Let's add blah.txt

git add blah.txt; git status; git commit -m 'Here I demonstrate the short syntax for commit messages'

We've now added blah.txt to our repository


```
ryan@localhost: ~/Documents/GithubProjects/learnGit
File  Edit  View  Search  Terminal  Help
ryan@localhost:~/Documents/GithubProjects/learnGit$ git ad
d blah.txt
ryan@localhost:~/Documents/GithubProjects/learnGit$ git st
atus
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   blah.txt

ryan@localhost:~/Documents/GithubProjects/learnGit$ git co
mmit -m 'Here I demonstrate the short syntax for commit me
ssages'
[master 2b4b83a] Here I demonstrate the short syntax for c
ommit messages
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 blah.txt
ryan@localhost:~/Documents/GithubProjects/learnGit$
```

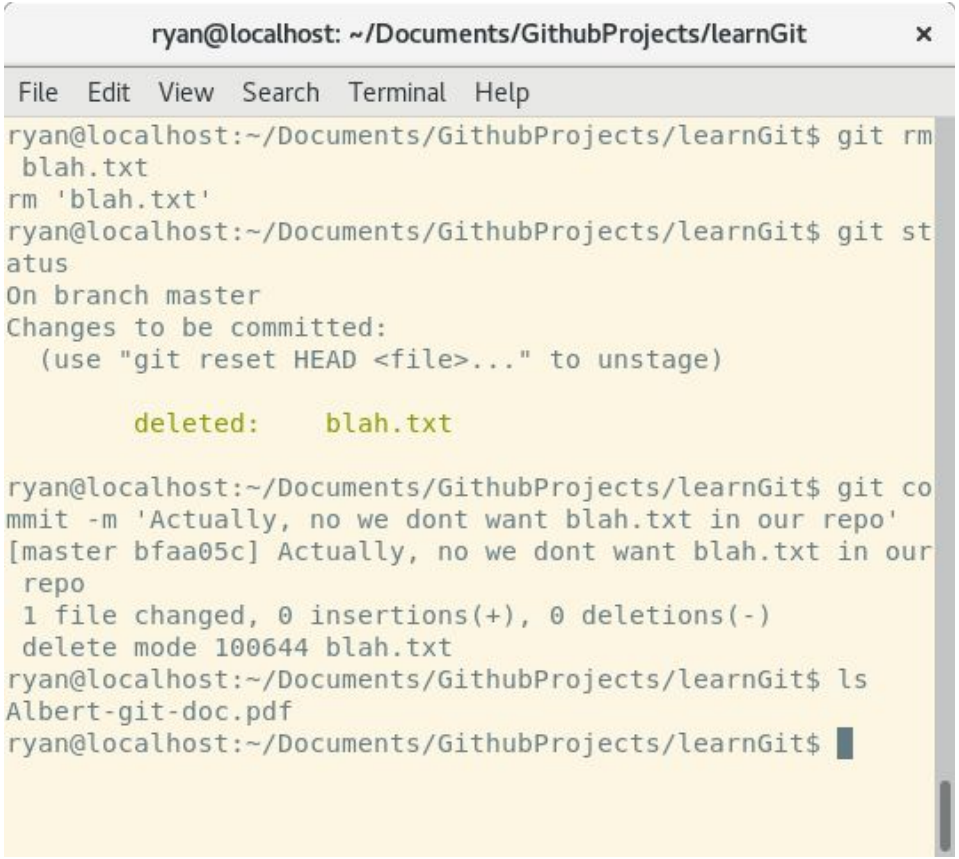
Commands

Let's remove blah.txt from repository

git rm blah.txt

Removes the file - both from repository tracking and file system.

git status; git commit -m 'Actually, no we dont want blah.txt in our repo'; ls



Commands

Let's add (stage) a new file to be committed.

Touch newFile.txt; git add newFile.txt; git status


```
ryan@localhost: ~/Documents/GithubProjects/learnGit
File  Edit  View  Search  Terminal  Help
ryan@localhost:~/Documents/GithubProjects/learnGit$ gitun
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   newFile.txt

Untracked files not listed (use -u option to show untracke
d files)
ryan@localhost:~/Documents/GithubProjects/learnGit$
```

Commands

We haven't finished editing yet though. It is not ready to be committed.

So unstage so we can do another snapshot to index later

git reset HEAD newFile.txt


```
ryan@localhost: ~/Documents/GithubProjects/learnGit
File  Edit  View  Search  Terminal  Help
ryan@localhost:~/Documents/GithubProjects/learnGit$ gitun
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   newFile.txt

Untracked files not listed (use -u option to show untracke
d files)
ryan@localhost:~/Documents/GithubProjects/learnGit$ git re
set HEAD newFile.txt
ryan@localhost:~/Documents/GithubProjects/learnGit$ git st
atus
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be comm
itted)

        newFile.txt

nothing added to commit but untracked files present (use "
git add" to track)
ryan@localhost:~/Documents/GithubProjects/learnGit$
```

Aside: Remote Repository

Search or jump to... / Pull requests Issues Marketplace Explore

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner: RJaBi / Repository name *: learnGit

Great repository names are short and memorable. Need inspiration? How about fantastic-robot?

Description (optional)

A repository for a slideshow on how to learn git.

Public
Anyone can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: None | Add a license: None

Create repository

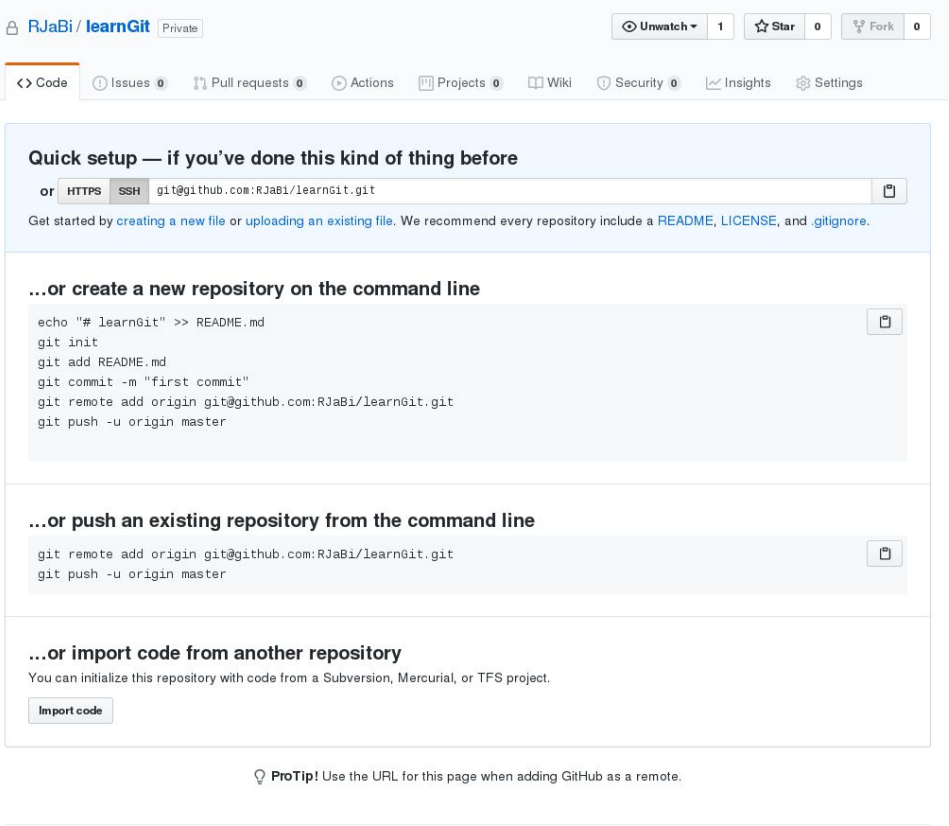
Github

Github is one of many online git repository systems. I use it purely because I have an account there.

Alternatives include

- bitbucket
- Gitlab

Self hosted and enterprise solutions are also available.



Commands

I use the ssh interface, https also available

git remote add origin
git@github.com:RJaBi/learnGit.git

Check remote worked using

git remote -v

Push to remote repository using

git push -u origin master

Origin is address of remote repo, master is the branch we're pushing

```
ryanbignell@coepp24: ~/OneDrive/Folder/University/Phd/2020/learnGit
File  Edit  View  Search  Terminal  Help
ryanbignell@coepp24:learnGit$ git remote add origin git@githu
b.com:RJaBi/learnGit.git
ryanbignell@coepp24:learnGit$ git remote -v
origin  git@github.com:RJaBi/learnGit.git (fetch)
origin  git@github.com:RJaBi/learnGit.git (push)
ryanbignell@coepp24:learnGit$ git push -u origin master
Counting objects: 7, done.
Delta compression using up to 16 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 126.93 KiB | 0 bytes/s, done.
Total 7 (delta 0), reused 0 (delta 0)
To git@github.com:RJaBi/learnGit.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origi
n.
ryanbignell@coepp24:learnGit$
```

Aside: Remote Repository

RJaBi / learnGit Private

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Actions | Projects 0 | Wiki | Security 0 | Insights | Settings

A repository for a slideshow on how to learn git.

Edit

Manage topics

3 commits | 1 branch | 0 packages | 0 releases | 0 contributors

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

Ryan Bignell and Ryan Bignell Actually, no we dont want blah.txt in our repo — Latest commit bfaa05c 4 days ago

Albert-git-doc.pdf — Standard is a single line describing change.i.e. add guide — 4 days ago

Add a README with an overview of your project.

Add a README

remote repository

Commands

Add a file to tell it we don't care about .txt files

git status; echo *.txt > .gitingore; git status

Can see that have a new file .gitignore and that git status is no longer telling us about .txt files.

Intended for use with i.e. *.pyc, *aux files (working files)

This Github Repo has examples for different languages.


```
ryanbignell@coepp24: ~/OneDrive/Folder/University/Phd/2020/learnGit
File  Edit  View  Search  Terminal  Help
ryanbignell@coepp24:learnGit$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Untracked files:
  (use "git add <file>..." to include in what will be committ
ed)

        newFile.txt

nothing added to commit but untracked files present (use "git
 add" to track)
ryanbignell@coepp24:learnGit$ echo *.txt > .gitignore
ryanbignell@coepp24:learnGit$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Untracked files:
  (use "git add <file>..." to include in what will be committ
ed)

        .gitignore

nothing added to commit but untracked files present (use "git
 add" to track)
ryanbignell@coepp24:learnGit$
```

Commands

Let's just add this file to our repo

git add .gitignore; git status; git commit -m 'Added a .gitignore file'

git status

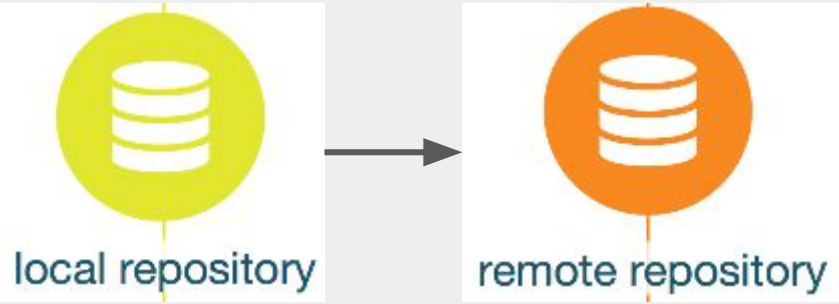
Now see that our local repository has more commits than the remote.


```
ryanbignell@coepp24: ~/OneDrive/Folder/University/Phd/2020/learnGit
File  Edit  View  Search  Terminal  Help
ryanbignell@coepp24:learnGit$ git add .gitignore
ryanbignell@coepp24:learnGit$ 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)

        new file:   .gitignore

ryanbignell@coepp24:learnGit$ git commit -m 'Added a .gitigno
re file'
[master 2dcd3d3] Added a .gitignore file
 1 file changed, 1 insertion(+)
 create mode 100644 .gitignore
ryanbignell@coepp24:learnGit$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
nothing to commit, working directory clean
ryanbignell@coepp24:learnGit$
```

Commands

So push the changes to remote repository

git push -u origin master

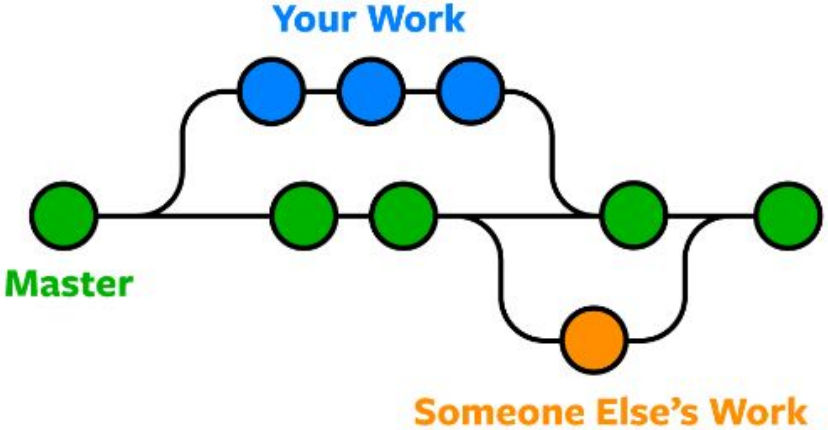



```
ryanbignell@coepp24: ~/OneDrive/Folder/University/Phd/2020/learnGit
File  Edit  View  Search  Terminal  Help
ryanbignell@coepp24:learnGit$ git push -u origin master
Counting objects: 3, done.
Delta compression using up to 16 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 306 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To git@github.com:RJaBi/learnGit.git
   bfaa05c..2dcd3d3  master -> master
Branch master set up to track remote branch master from origi
n.
ryanbignell@coepp24:learnGit$
```

Quotes

"The diagram right visualizes a repository with two isolated lines of development, one for a little feature, and one for a longer-running feature. By developing them in branches, it's not only possible to work on both of them in parallel, but it also keeps the main master branch free from questionable code."

"Git branches are effectively a pointer to a snapshot of your changes."



Good guides are available from Atlassian and Noble Desktop.

(I actually used Atlassian git guides a lot when first learning git. They're very good).

Words

Use the example of calculating average as an example. Write the opposite python script. The mean value here is 2.5.

ryan@localhost: ~/Documents/GithubProjects/learnGit

File Edit View Search Terminal Help

File Edit Options Buffers Tools Python Help

```python
import numpy as np

def mean(array):
    count=0
    sum = 0
    for x in array:
        sum = sum + x
        count = count + 1
    return float(sum)/float(count)
def main():
    data = np.array([1.0, 2.0, 3.0, 4.0])
    ave = mean(data)
    print(data, 'ave:',ave)

if __name__ == '__main__':
    main()
```

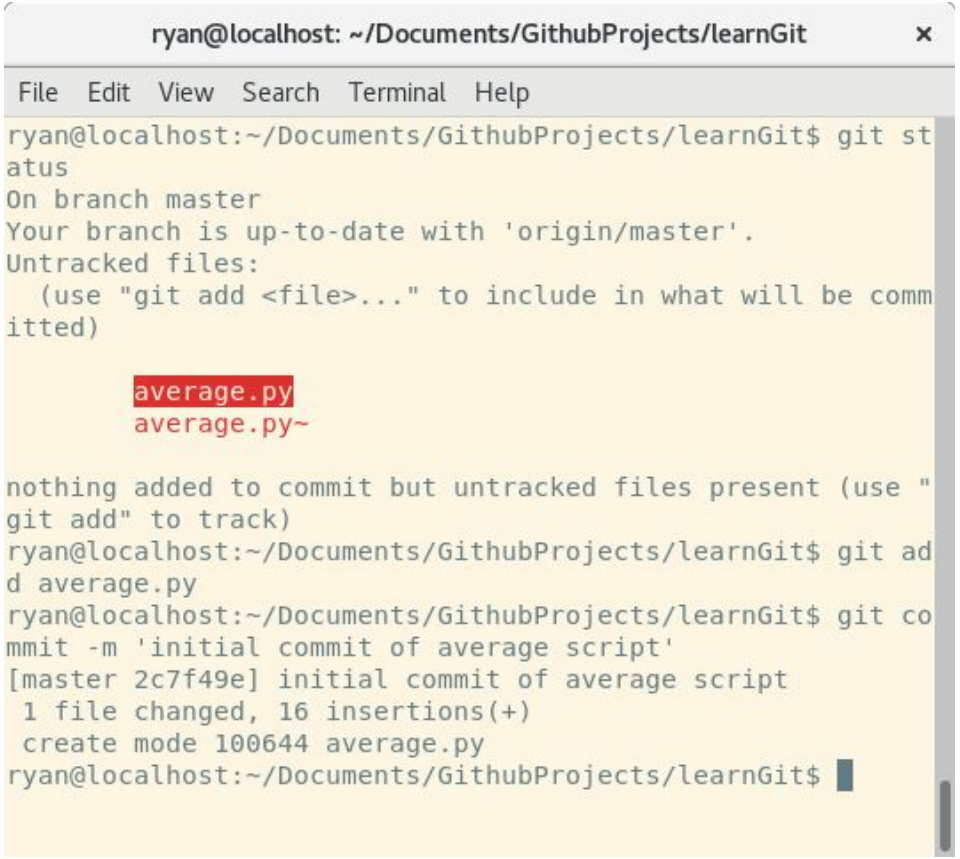
-UUU:----F1 **average.py** All L10 (Python) ----------

master

Use the example of calculating average as an example. Write the opposite python script. The mean value here is 2.5.

Let's commit it as usual.

git status; git add average.py; git commit -m 'initial commit of average script'



improveMeanCalc

Now switch to a new branch. We're going to make the calculation of the mean better.

git status; git checkout - b improveMeanCalc

improveMeanCalc is the name of the branch

git status; git branch

git branch shows the branches in our local repository. Clearly improveMeanCalc is the active branch in the workspace/index

ryan@localhost: ~/Documents/GithubProjects/learnGit

File Edit View Search Terminal Tabs Help

ryan@localhost: ~/Doc... ryan@localhost: ~/Doc...

```
ryan@localhost:~/Documents/GithubProjects/learnGit$ git st
atus
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
nothing to commit, working directory clean
ryan@localhost:~/Documents/GithubProjects/learnGit$ git ch
eckout -b improveMeanCalc
Switched to a new branch 'improveMeanCalc'
ryan@localhost:~/Documents/GithubProjects/learnGit$ git st
atus
On branch improveMeanCalc
nothing to commit, working directory clean
ryan@localhost:~/Documents/GithubProjects/learnGit$ git br
anch
* improveMeanCalc
  master
ryan@localhost:~/Documents/GithubProjects/learnGit$
```

improveMeanCalc

So i've just replaced the loop to calculate the mean with numpy's vectorised form.

ryan@localhost: ~/Documents/GithubProjects/learnGit

File Edit View Search Terminal Help

```
File Edit Options Buffers Tools Python Help
import numpy as np

def mean(array):
    return np.mean(array)
def main():
    data = np.array([1.0, 2.0, 3.0, 4.0])
    ave = mean(data)
    print(data, 'ave:',ave)

if __name__ == '__main__':
    main()
```

```
-UU-:----F1  average.py      All L4    Git:improveMeanCalc
(No changes need to be saved)
```

improveMeanCalc

So i've just replaced the loop to calculate the mean with numpy's vectorised form. Then commit to repo

Git status; git add average.py; git commit -m 'replaced loop mean with vector mean'


```
ryan@localhost: ~/Documents/GithubProjects/learnGit
File  Edit  View  Search  Terminal  Help
ryan@localhost:~/Documents/GithubProjects/learnGit$ git st
atus
On branch improveMeanCalc
Changes not staged for commit:
  (use "git add <file>..." to update what will be committe
d)
  (use "git checkout -- <file>..." to discard changes in w
orking directory)

        modified:   average.py

no changes added to commit (use "git add" and/or "git comm
it -a")
ryan@localhost:~/Documents/GithubProjects/learnGit$ git ad
d average.py
ryan@localhost:~/Documents/GithubProjects/learnGit$ git co
mmit -m 'replaced loop mean with vector mean'
[improveMeanCalc 3a4dda9] replaced loop mean with vector m
ean
 1 file changed, 1 insertion(+), 6 deletions(-)
ryan@localhost:~/Documents/GithubProjects/learnGit$
```

improveMeanCalc -> master

Let's double check that master hasn't been changed.

git checkout master; head average.py

It hasn't changed! Still the loop calc.

We used git checkout earlier to 'reset' a changed file to the version in the index. Here we use it to switch what branch we're on

git branch

```
ryan@localhost: ~/Documents/GithubProjects/learnGit
File  Edit  View  Search  Terminal  Help
ryan@localhost:~/Documents/GithubProjects/learnGit$ git ch
eckout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
ryan@localhost:~/Documents/GithubProjects/learnGit$ head a
verage.py
import numpy as np

def mean(array):
    count=0
    sum = 0
    for x in array:
        sum = sum + x
        count = count + 1
    return float(sum)/float(count)
def main():
ryan@localhost:~/Documents/GithubProjects/learnGit$ git br
anch
  improveMeanCalc
* master
ryan@localhost:~/Documents/GithubProjects/learnGit$
```

master

While we're here. Let's add some new functionality to average.py - the std. Dev. Here i've just made that change, printed the python file to the terminal and run it once.

ryan@localhost: ~/Documents/GithubProjects/learnGit

File Edit View Search Terminal Help

```
ryan@localhost:~/Documents/GithubProjects/learnGit$ cat av
erage.py
import numpy as np
def mean(array):
    count=0
    sum = 0
    for x in array:
        sum = sum + x
        count = count + 1
    return float(sum)/float(count)
def main():
    data = np.array([1.0, 2.0, 3.0, 4.0])
    ave = mean(data)
    stdDev = np.std(data)
    print(data, 'ave:',ave, 'std.dev:',stdDev)
if __name__ == '__main__':
    main()
ryan@localhost:~/Documents/GithubProjects/learnGit$ python
 average.py
(array([ 1.,  2.,  3.,  4.]), 'ave:', 2.5, 'std.dev:', 1.1
180339887498949)
ryan@localhost:~/Documents/GithubProjects/learnGit$
```

master

Let's commit changes

git status; git add average.py; git commit -m 'added std.dev calc'


```
ryan@localhost: ~/Documents/GithubProjects/learnGit
File  Edit  View  Search  Terminal  Help
ryan@localhost:~/Documents/GithubProjects/learnGit$ git st
atus
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
Changes not staged for commit:
  (use "git add <file>..." to update what will be committe
d)
  (use "git checkout -- <file>..." to discard changes in w
orking directory)

        modified:   average.py

no changes added to commit (use "git add" and/or "git comm
it -a")
ryan@localhost:~/Documents/GithubProjects/learnGit$ git ad
d average.py
ryan@localhost:~/Documents/GithubProjects/learnGit$ git co
mmit -m 'added std.dev calc'
[master 7f8486b] added std.dev calc
 1 file changed, 2 insertions(+), 3 deletions(-)
ryan@localhost:~/Documents/GithubProjects/learnGit$
```

master

Let's commit changes

git status; git add average.py; git commit -m 'added std.dev calc'

Also push changes on master branch to remote repo


```
ryan@localhost: ~/Documents/GithubProjects/learnGit
File  Edit  View  Search  Terminal  Help
ryan@localhost:~/Documents/GithubProjects/learnGit$ git st
atus
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)
nothing to commit, working directory clean
ryan@localhost:~/Documents/GithubProjects/learnGit$ git pu
sh
Warning: Permanently added the RSA host key for IP address
 '140.82.112.3' to the list of known hosts.
Total 0 (delta 0), reused 0 (delta 0)
To git@github.com:RJaBi/learnGit.git
   2dcd3d3..7f8486b  master -> master
ryan@localhost:~/Documents/GithubProjects/learnGit$
```

master

Let's grab the changes from the improveMeanCalc branch now.

git status; git merge improveMeanCalc

This bring changes from improveMeanCalc into master


```
ryan@localhost: ~/Documents/GithubProjects/learnGit
File  Edit  View  Search  Terminal  Help
ryan@localhost:~/Documents/GithubProjects/learnGit$ git st
atus
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean
ryan@localhost:~/Documents/GithubProjects/learnGit$ git me
rge improveMeanCalc
Auto-merging average.py
Merge made by the 'recursive' strategy.
 average.py | 7 +------
 1 file changed, 1 insertion(+), 6 deletions(-)
ryan@localhost:~/Documents/GithubProjects/learnGit$
```

master

Let's grab the changes from the improveMeanCalc branch now.

git status; git merge improveMeanCalc

This bring changes from improveMeanCalc into master

ryan@localhost: ~/Documents/GithubProjects/learnGit

File Edit View Search Terminal Help

```
Merge branch 'improveMeanCalc'

# Please enter a commit message to explain why this merge
is necessary,
# especially if it merges an updated upstream into a topic
 branch.
#
# Lines starting with '#' will be ignored, and an empty me
ssage aborts
# the commit.
~
~
~
~
~
~
~
~
~
~
~
<cuments/GithubProjects/learnGit/.git/MERGE_MSG" 7L, 258C
```

master

Push to remote repo

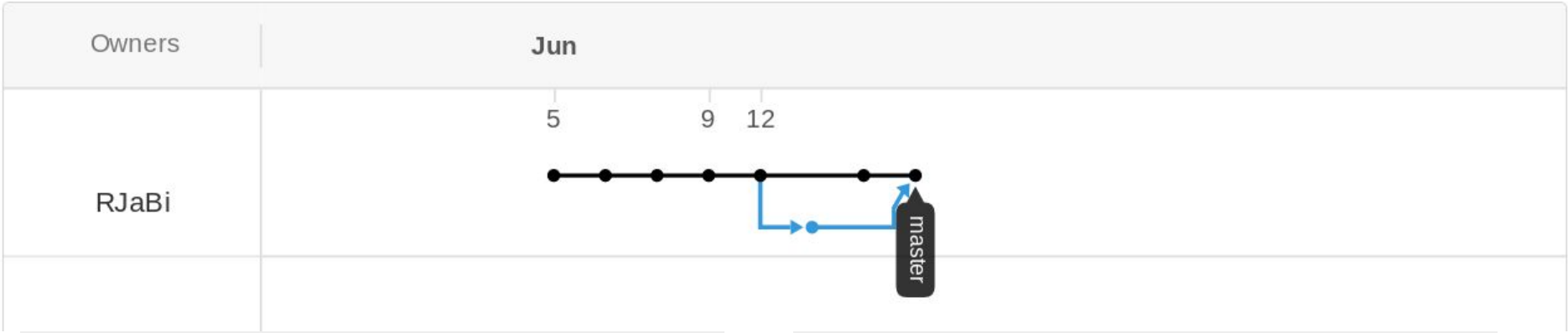
git status; git push -u origin master

Can see that now 2 commits ahead of remote repo. This is as the merge process is in itself a commit.


```
ryan@localhost: ~/Documents/GithubProjects/learnGit
File  Edit  View  Search  Terminal  Help
ryan@localhost:~/Documents/GithubProjects/learnGit$ git st
atus
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)
nothing to commit, working directory clean
ryan@localhost:~/Documents/GithubProjects/learnGit$ git pu
sh -u origin master
Warning: Permanently added the RSA host key for IP address
 '140.82.113.3' to the list of known hosts.
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 350 bytes | 0 bytes/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 loc
al objects.
To git@github.com:RJaBi/learnGit.git
   7f8486b..5399509  master -> master
Branch master set up to track remote branch master from or
igin.
ryan@localhost:~/Documents/GithubProjects/learnGit$
```

Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.



A convenient way to visualise your branch structure is on Github.

Here we can see where we branched off for the improveMeanCalc commit (in blue)

Conclusions?

Hopefully this slideshow is useful as a guide to some basic git usage.

I haven't covered features such as

- The stash
- Pull requests
- git log
- git diff

Resources

- Albert's repo of git tricks: https://github.cs.adelaide.edu.au/a1687112/git-gear
- `git status -uno` is my standard git status command. It doesn't show untracked files
- Atlassian and Noble Desktop.
- https://patrickzahnd.ch/blog.html#gittransport

- This repository and slideshow: https://github.com/RJaBi/learnGit