

Yet Another Git(Hub) Tutorial

Diya Das

06.26.2017

for the CDIPS Data Science Workshop

What is version control and why should I use it?

- Process of **keeping track of changes** between different versions of a file
- What if **different versions of your scripts produce different results?**
 - This can be difficult to resolve without knowing versions!

Version control

Ok, very common systems

- Saving a new version of your file with suffixes:
 - _MMDDYY
 - _[Initials]



Much better systems

- a unique identifier AND
- a **message that tells you what changed** between versions
- Also save space by only saving **diffs** (differences) between versions
 - e.g. incremental computer backup

What is Git and why should I use it?

- A distributed version control system that works with almost any file type, commonly used in software development
- All of the features described above:
 - Unique IDs for each version of a file
 - In logs, allows you to save a descriptive message about your changes
 - Saves diffs (for text files, these can be browsed)
- Each project in Git is a repository: a directory of files with additional version tracking features

What is GitHub, then?

- A website that hosts Git repositories (repos)
- Helps facilitate group projects / working from multiple machines because your work is hosted on their servers

(You do have to sync your work, but it's worth it.)

Setting up Git/GitHub on your computer (and assigning authorship)

- Sign up for an account on [GitHub](#)
 - For later: look into [educational discount](#)
- [Download and install Git](#) (hopefully, you did this already!)
- Open Terminal
- `git config --global user.name "Mona Lisa"`
- `git config --global user.email "mona@berkeley.edu"`
 - Make sure this email is associated with your GitHub account:
<https://github.com/settings/emails>

THIS IS GIT. IT TRACKS COLLABORATIVE WORK
ON PROJECTS THROUGH A BEAUTIFUL
DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE 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 DOWNLOAD A FRESH COPY.



XKCD: Git (1597)

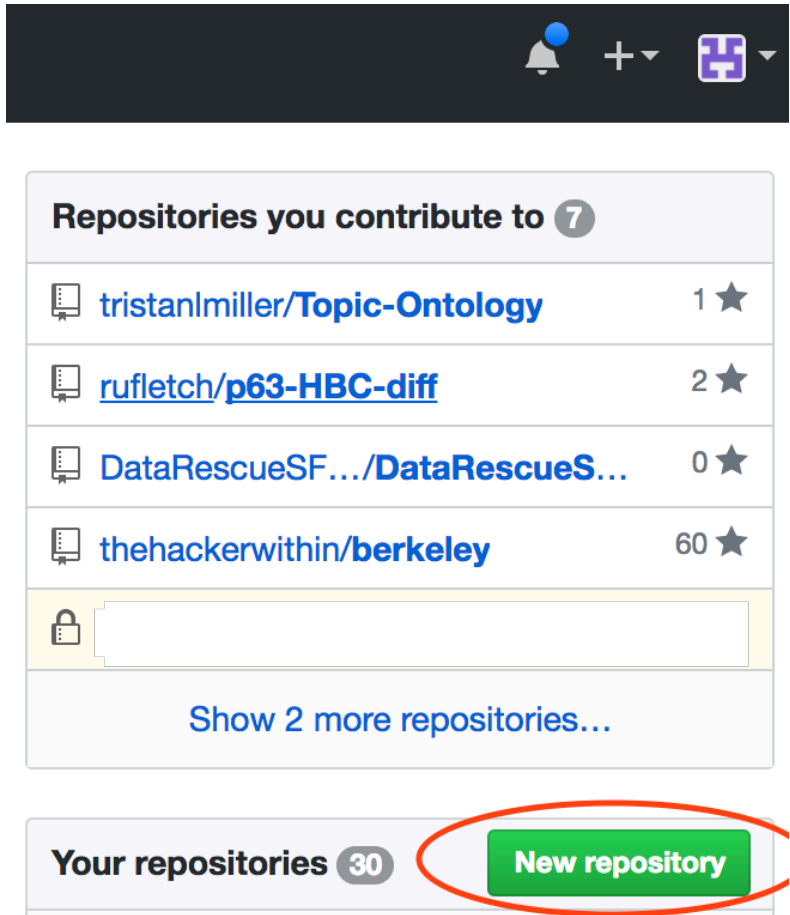
The goal is to *avoid* this.

The rest of this tutorial is focused on walking you through some commands.

What's the basic workflow?






- (Initialize a repo. Easiest on GitHub.)
- (**Clone** your repo – or someone else's – onto your computer.)
- Terminal: `cd myfirstrepo`
- **Pull** changes from the server.
- Edit a file, make a change. Nothing to do with Git.
- **Add** your file to the queue to be staged. x N
- **Commit** your changes with a message that explains them.
- **Push** your changes to the server.

Initialize a repo on GitHub



The image shows a screenshot of the GitHub interface. At the top, there is a dark navigation bar with a bell icon, a plus sign, and a GitHub logo. Below this, a section titled "Repositories you contribute to 7" lists four repositories: `tristanmiller/Topic-Ontology` (1 star), `rufletch/p63-HBC-diff` (2 stars), `DataRescueSF.../DataRescueS...` (0 stars), and `thehackerwithin/berkeley` (60 stars). A fifth row shows a lock icon and a redacted repository name. A link "Show 2 more repositories..." is below the list. At the bottom, a section titled "Your repositories 30" contains a green "New repository" button, which is circled in red.

Repositories you contribute to 7

 <code>tristanmiller/Topic-Ontology</code>	1 ★
 <code>rufletch/p63-HBC-diff</code>	2 ★
 <code>DataRescueSF.../DataRescueS...</code>	0 ★
 <code>thehackerwithin/berkeley</code>	60 ★
 [Redacted]	

[Show 2 more repositories...](#)

Your repositories 30 [New repository](#)

Initialize a repo on GitHub

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner



diyadas ▾

/

Repository name

myfirstrepo



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

Description (optional)

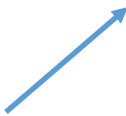
this is a test

☒ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

Add a license: **None** ▾



My first repo

diyadas / myfirstrepo

Unwatch ▾

1

★ Star

0

Fork

0

<> Code

! Issues 0

🔗 Pull requests 0

📁 Projects 0

📖 Wiki

⚙ Settings

🔍 Insights ▾

this is a test

Edit

Add topics

🕒 1 commit

🌿 1 branch

📦 0 releases

👤 1 contributor

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾

diyadas Initial commit

Latest commit ca8f3eb just now

📄 README.md

Initial commit


just now

📖 README.md

myfirstrepo

this is a test

Detour: add a collaborator

 diyadas / myfirstrepo

Unwatch ▼1

★ Star0

🍴 Fork0

<> Code

🔔 Issues0

🔗 Pull requests0

📁 Projects0

📖 Wiki

⚙️ Settings

Insights ▼

Options

Collaborators

Branches

Webhooks

Integrations & services

Deploy keys

Collaborators

Push access to the repository

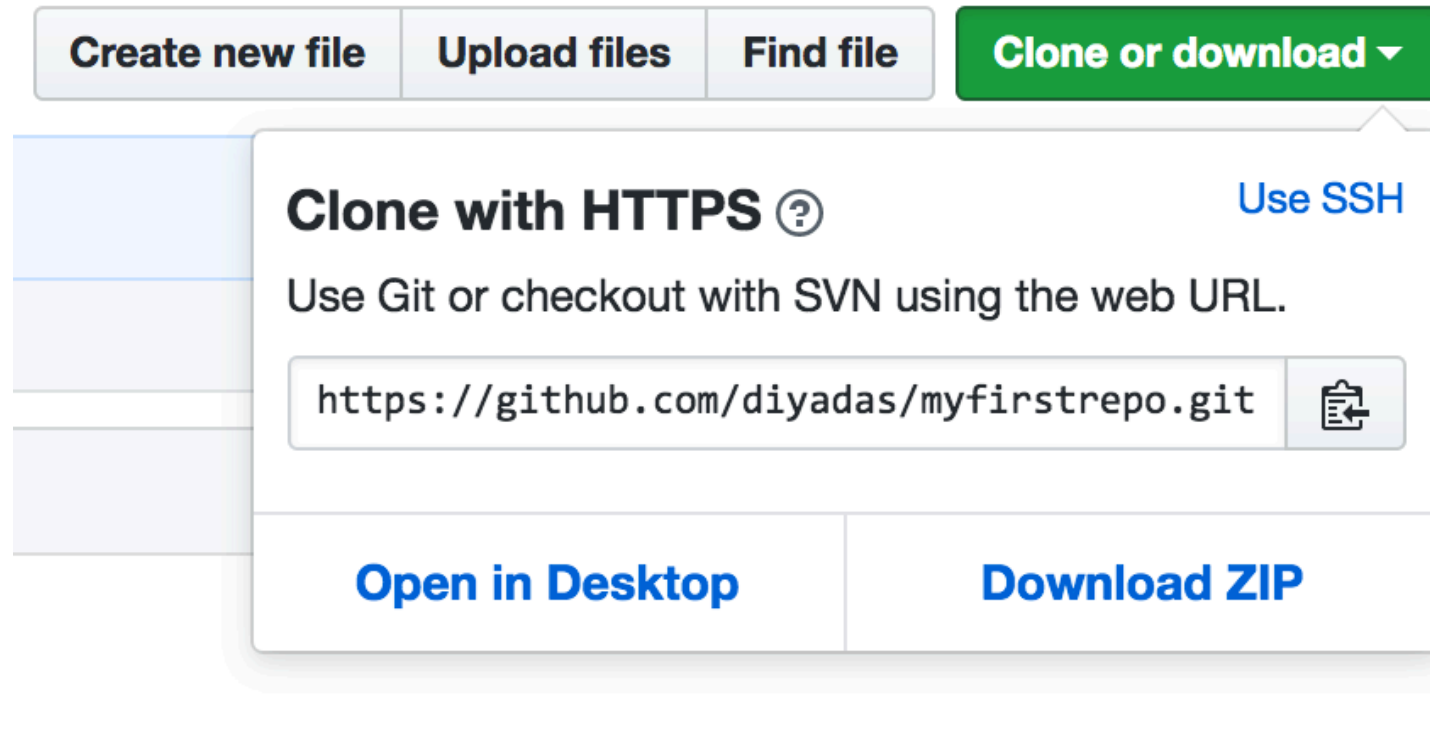
This repository doesn't have any collaborators yet. Use the form below to add a collaborator.

Search by username, full name or email address

You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.

Add collaborator

Clone myfirstrepo



```
git clone https://github.com/diyadas/myfirstrepo.git
```

What's the basic workflow?

- ~~(Initialize a repo. Easiest on GitHub.)~~
- ~~(**Clone** your repo — or someone else's — onto your computer.)~~
- Terminal: `cd myfirstrepo`
- **Pull** changes from the server.
- Edit a file, make a change. Nothing to do with Git.
- **Add** your file to the queue to be staged. x N
- **Commit** your changes with a message that explains them.
- **Push** your changes to the server.

What's the basic workflow?

- ~~(Initialize a repo. Easiest on GitHub.)~~
- ~~(**Clone** your repo — or someone else's — onto your computer.)~~
- Terminal: `cd myfirstrepo`
- **Pull** changes from the server. `git pull`
- Edit a file, make a change. Nothing to do with Git.
- **Add** your file to the queue to be staged. x N
- **Commit** your changes with a message that explains them.
- **Push** your changes to the server.

What's the basic workflow?

- (Initialize a repo. Easiest on GitHub.)
- (**Clone** your repo – or someone else's – onto your computer.)
- Terminal: `cd myfirstrepo`
- **Pull** changes from the server.
- Edit a file, make a change. Nothing to do with Git. `emacs README.md`
- **Add** your file to the queue to be staged. x N
- **Commit** your changes with a message that explains them.
- **Push** your changes to the server.

What's the basic workflow?

- (Initialize a repo. Easiest on GitHub.)
- (**Clone** your repo – or someone else's – onto your computer.)
- Terminal: `cd myfirstrepo`
- **Pull** changes from the server.
- Edit a file, make a change. Nothing to do with Git.
- **Add** your file to the queue to be staged. x N `git add README.md`
- **Commit** your changes with a message that explains them.
- **Push** your changes to the server.

What's the basic workflow?

- (Initialize a repo. Easiest on GitHub.)
- (**Clone** your repo – or someone else's – onto your computer.)
- Terminal: `cd myfirstrepo`
- **Pull** changes from the server.
- Edit a file, make a change. Nothing to do with Git.
- **Add** your file to the queue to be staged. x N
- **Commit** your changes with a message that explains them.
- **Push** your changes to the server. `git commit -m "added name"`

What's the basic workflow?

- (Initialize a repo. Easiest on GitHub.)
- (**Clone** your repo – or someone else's – onto your computer.)
- Terminal: `cd myfirstrepo`
- **Pull** changes from the server.
- Edit a file, make a change. Nothing to do with Git.
- **Add** your file to the queue to be staged. x N
- **Commit** your changes with a message that explains them.
- **Push** your changes to the server. `git push`

Let's go take a look at our work!

diyadas / myfirstrepo

Unwatch

1

Star

0

Fork

0

<> Code

Issues

0

Pull requests

0

Projects

0

Wiki

Settings

Insights

this is a test

Edit

Add topics

2 commits

1 branch

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

diyadas added name

Latest commit 6daa3c7 13 seconds ago

README.md

added name

13 seconds ago

README.md

myfirstrepo

this is a test by Diya

<> Code

! Issues 0

🔗 Pull requests 0

📁 Projects 0

📖 Wiki

⚙ Settings

Insights ▾

Branch: master ▾

🔗 Commits on Jun 24, 2017

**added name**

diyadas committed a minute ago



6daa3c7

**Initial commit**

diyadas committed 16 minutes ago



ca8f3eb

**added name**[Browse files](#)

🔗 master



diyadas committed 2 minutes ago

1 parent [ca8f3eb](#) commit [6daa3c7343a26635500fdb7c228882c3dc05a119](#)📄 Showing 1 **changed file** with 1 **addition** and 0 **deletions**.

Ignore Spaces

Unified

Split

1 ■■■■ README.md



View



... @@ -1,2 +1,3 @@

1 # myfirstrepo

2 this is a test

1 # myfirstrepo

2 this is a test

3 +by Diya

Examining things locally: useful Terminal commands

git log

```
Diyas-MacBook-Pro:myfirstrepo diyadas$ git log
commit 6daa3c7343a26635500fdb7c228882c3dc05a119
Author: Diya Das <diyadas@users.noreply.github.com>
Date: Sat Jun 24 16:48:57 2017 -0700

    added name

commit ca8f3eb259a147ea5d2ecafa22cf55efebe3d79c
Author: Diya Das <diyadas@users.noreply.github.com>
Date: Sat Jun 24 16:33:48 2017 -0700

    Initial commit
```

git status

```
Diyas-MacBook-Pro:myfirstrepo diyadas$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean
```

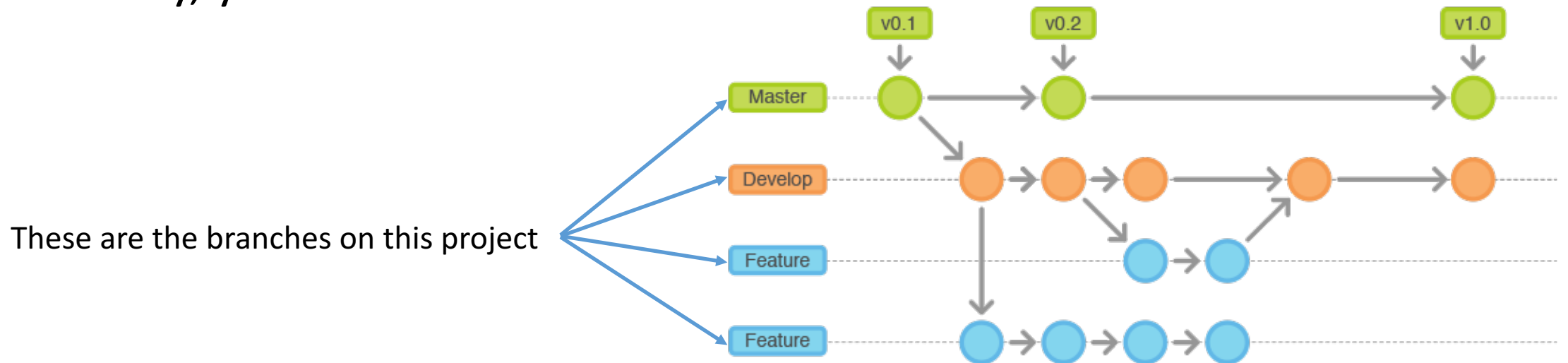
```
Diyas-MacBook-Pro:myfirstrepo diyadas$ emacs README.md
Diyas-MacBook-Pro:myfirstrepo diyadas$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   README.md

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

Slightly more advanced Git: branching

- You'll notice that a lot of messages mentioned being on branch "master" or being up to date with "master"
- master is the default (and only) branch in our repo
- If you want to work on a feature without affecting your main project history, you can create a **branch**



Abandoning all local changes

If you'd like to discard all local changes to tracked files:

1. `git fetch origin`

- Fetch all changes from origin (where you cloned from), but don't merge them (pull = fetch and then merge)

2. `git reset --hard origin/master`

- Reset your local repository to be in sync with origin/master

Unstaging a file (added, not yet committed)

```
Diyas-MacBook-Pro:myfirstrepo diyadas$ git add README.md
Diyas-MacBook-Pro:myfirstrepo diyadas$ 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:   README.md
```

Reverting a commit (oops)

- If you made a mistake, and you want to undo it, there's a way to do so in Git
- What you need to do depends on what you've done, so I recommend reading [this comprehensive StackOverflow answer](#)
 - Most common: Undo published commits with new commits
 - **Do not delete a published commit that others have access to**

Conflicting commits (the easy one)

- You pulled changes and started working on File A.
- Your collaborator pushed File B.
- You add and commit locally, but you can't push because you're out of sync!

```
Diyas-MacBook-Pro:myfirstrepo diyadas$ git push
To https://github.com/diyadas/myfirstrepo.git
! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'https://github.com/diyadas/myfirstrepo.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

Merging: resolving conflicting commits

`git pull :`

```
Merge branch 'master' of https://github.com/diyadas/myfirstrepo  
  
# 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 message aborts  
# the commit.
```

```
remote: Counting objects: 3, done.  
remote: Compressing objects: 100% (2/2), done.  
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0  
Unpacking objects: 100% (3/3), done.  
From https://github.com/diyadas/myfirstrepo  
    6daa3c7..f92b04f  master    -> origin/master  
Merge made by the 'recursive' strategy.  
 README.md | 4 +++-  
 1 file changed, 3 insertions(+), 1 deletion(-)  
Diyas-MacBook-Pro:myfirstrepo diyadas$ git status  
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
```



`git push`

Undoing commits of large files

- Important Note: GitHub is not designed by default to work with large files. Git is fine, though you'll have problems pushing to GitHub.
- Any file over 50 MB triggers a warning, and any file over 100 MB can't be pushed.
- `git filter-branch --force --index-filter \ 'git rm --cached --ignore-unmatch PATH-TO-YOUR-FILE-TO-REMOVE' \ --prune-empty --tag-name-filter cat -- --all`
 - Or use [BFG Cleaner](#) if your system is ok with Java

More useful resources

- Pragmatic Version Control Using Git, by Travis Swicegood
- Pro Git by Scott Chacum and Ben Straub: <https://git-scm.com/book/en/v2>
- The Docs: <https://git-scm.com/doc>
- My email is diyadas@berkeley.edu. I am not an expert, but I am pretty okay at Googling.