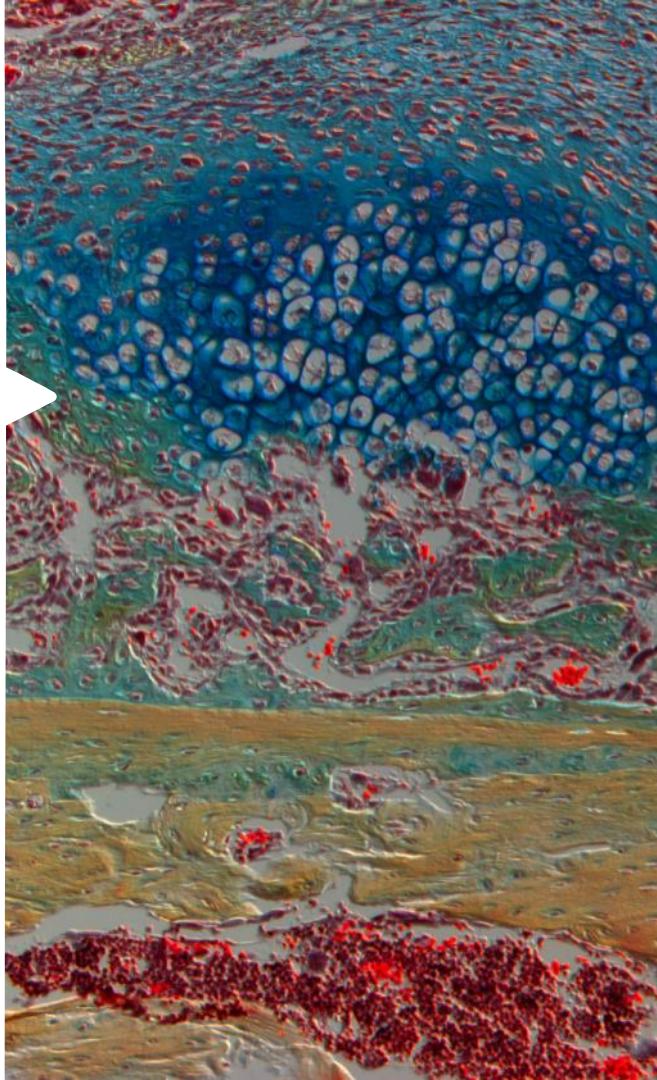




# Google Colab & Git/GitHub

Mark Grivainis

09/10/2020



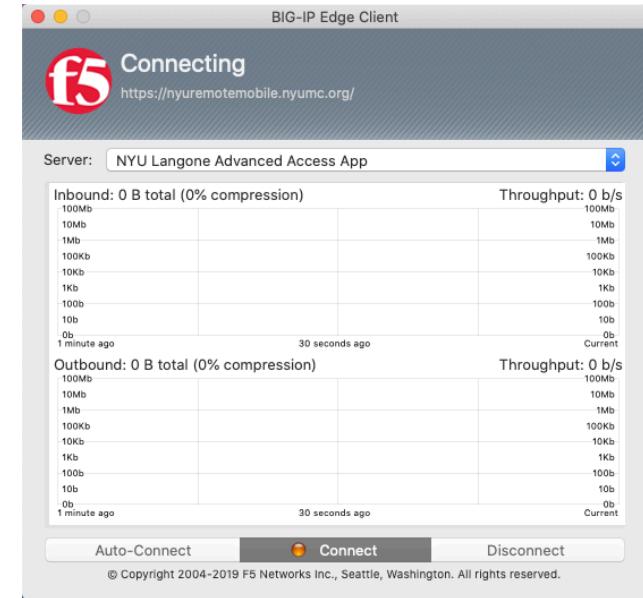
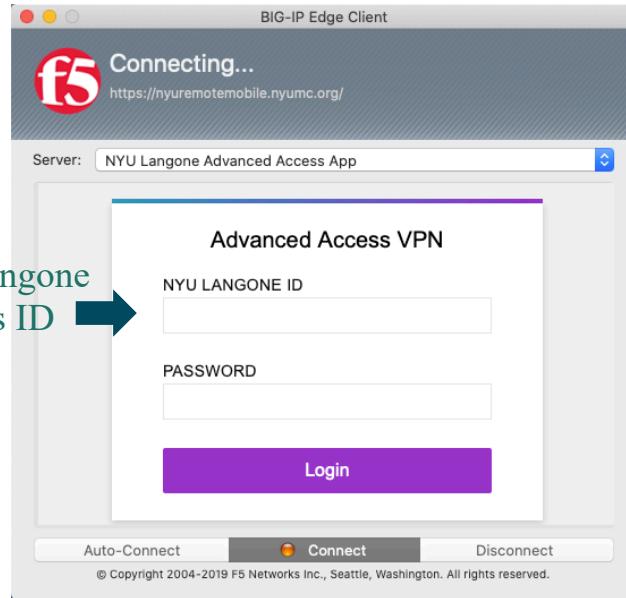
# Todays Lecture

- Overview of Concepts - 15 min
- Setting up a Git repository - 20 min
- Concept Check poll - 5 min
  - Bash and Git/GitHub
- Break / Questions - 10 min
- Google Colab - 40 min

# VPN Access

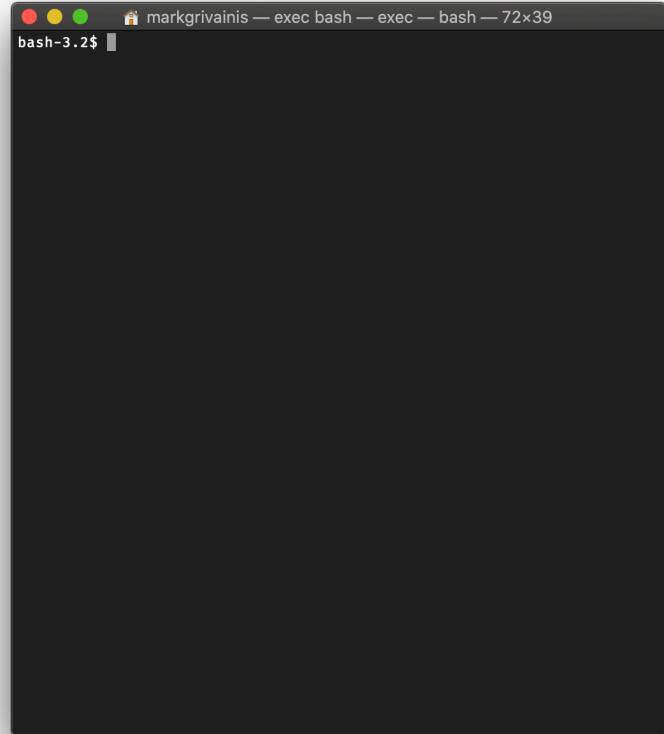
- Install BIG-IP Edge Client

- <http://atnyulmc.org/help-documentation/NYU-Langone-Advanced-Access-App>



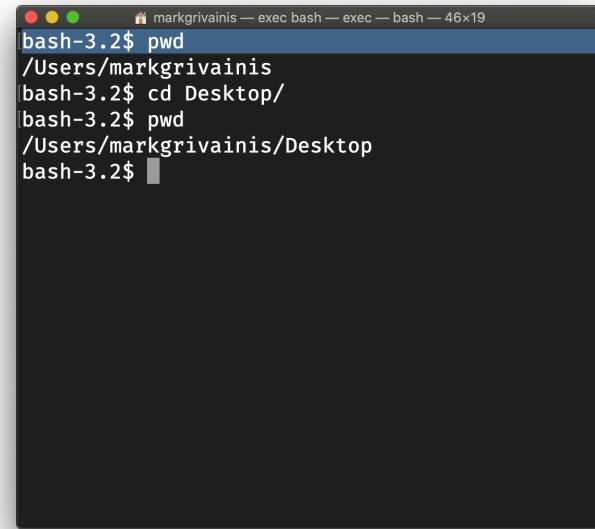
# Git – How to work with Git

- Mac / Linux
  - Terminal
- Windows
  - Git Bash (Preferred for now)
  - Powershell



# Git – Common Bash Commands

- **pwd**
  - This will list the currently active path
- **cd <path>**
  - This will change the directory
  - Uses either a relative or absolute path
  - ‘Relative’ uses the current directory as the base
  - ‘Absolute’ starts from the lowest level
    - On Unix based systems: ‘/’
    - On Windows, a drive letter: “C:/”
  - ‘.’ refers to the current directory
  - ‘..’ refers to the previous directory



A screenshot of a macOS terminal window titled "markgrivainis — exec bash — exec — bash — 46x19". The window shows the following command history:

```
bash-3.2$ pwd  
/Users/markgrivainis  
bash-3.2$ cd Desktop/  
bash-3.2$ pwd  
/Users/markgrivainis/Desktop  
bash-3.2$ █
```

# Git – Common Bash Commands

- **ls**

- This will list the contents of the current dir
  - Includes both files and folders
  - Does not include hidden files (dot files)
- To show hidden files add the ‘**-a**’ flag (all)
- To show detailed information about the files add the ‘**-l**’ (long) flag.
  - **ll** is often used as an alias for this
  - Columns
    - Permissions, # links, owner, group, size, modified date, filename

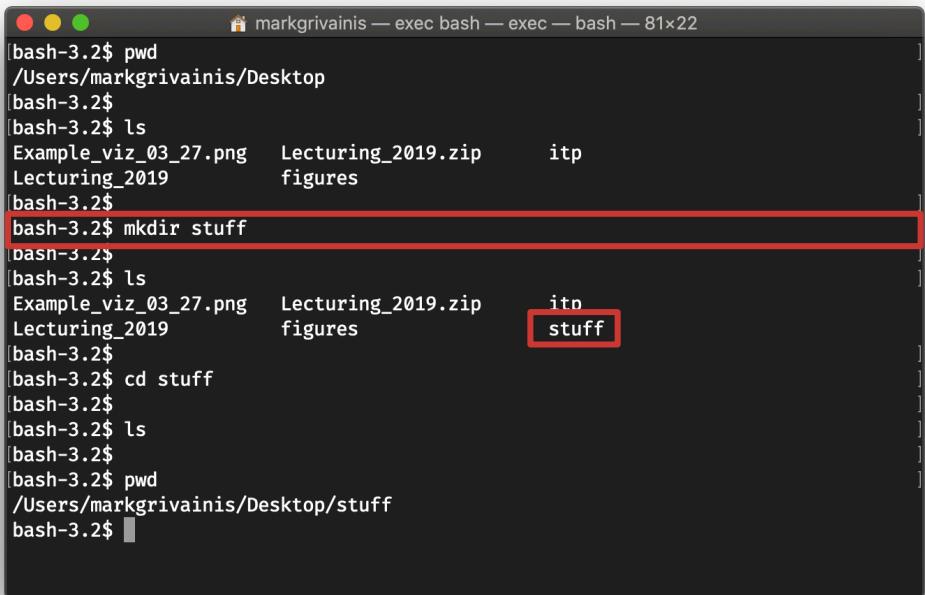
The screenshot shows a terminal window with three distinct sections, each demonstrating a different usage of the `ls` command:

- Section 1 (Red Box):** Shows the basic usage of `ls` without flags, listing files and folders in the current directory: `Example_viz_03_27.png`, `Lecturing_2019.zip`, `itp`, and `Lecturing_2019`.
- Section 2 (Yellow Box):** Shows the usage of the `-a` flag to include hidden files: `..`, `..`, `.DS_Store`, `localized`, `Example_viz_03_27.png`, `figures`, `Lecturing_2019`, and `itp`.
- Section 3 (Green Box):** Shows the usage of the `-l` flag to get detailed file information, including permissions, owner, group, size, and modification date:

File/Folder	Permissions	Owner	Group	Size	Last Modified	File/Folder
	total	527280				
<code>Example_viz_03_27.png</code>	-rw-r--r--@	1	markgrivainis	staff	614856	Mar 27 20:54
<code>Lecturing_2019</code>	drwxr-xr-x@	7	markgrivainis	staff	224	May 15 15:11
<code>Lecturing_2019.zip</code>	-rw-----@	1	markgrivainis	staff	256547798	Mar 31 14:42
<code>figures</code>	drwxr-x---	69	markgrivainis	staff	2208	Aug 19 18:50
<code>itp</code>	drwxr-xr-x	12	markgrivainis	staff	384	Aug 12 11:40

# Git – Common Bash Commands

- **mkdir <name>**
  - Creates a new directory called <name>
  - Does **not** cd into the directory
  - This directory will be empty
- **touch <file\_name>**
  - This will create a new blank file called <file\_name>
  - Remember to add an extension if it is required
    - .py, .txt, .md, etc..



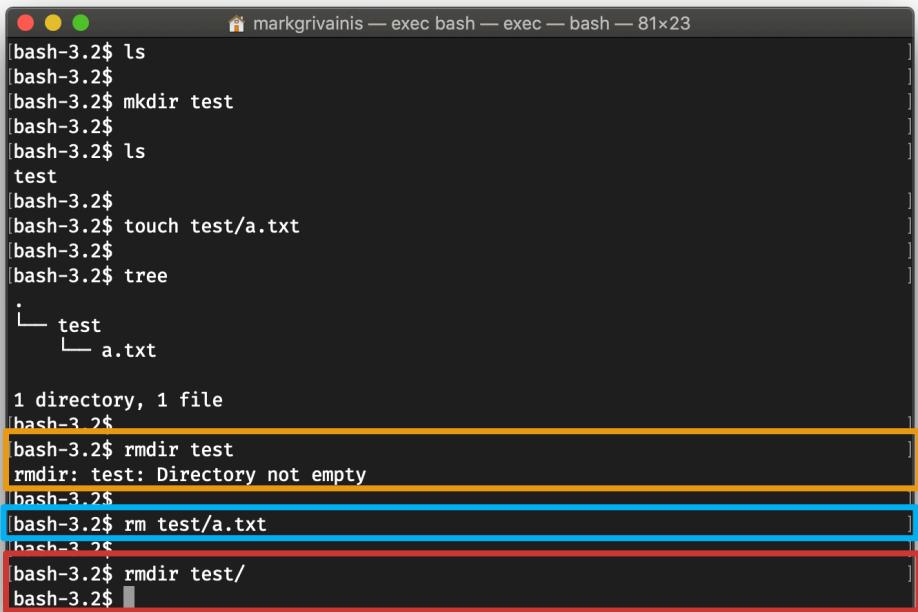
A screenshot of a macOS terminal window titled "markgrivainis — exec bash — exec — bash — 81x22". The terminal shows the following session:

```
bash-3.2$ pwd
/Users/markgrivainis/Desktop
bash-3.2$ ls
Example_viz_03_27.png  Lecturing_2019.zip      itp
Lecturing_2019          figures
bash-3.2$ mkdir stuff
bash-3.2$ ls
Example_viz_03_27.png  Lecturing_2019.zip      ito
Lecturing_2019          figures      stuff
bash-3.2$ cd stuff
bash-3.2$ ls
bash-3.2$ pwd
/Users/markgrivainis/Desktop/stuff
bash-3.2$
```

The command `mkdir stuff` is highlighted with a red rectangle, and the resulting directory "stuff" is also highlighted with a red rectangle.

# Git – Common Bash Commands

- **rmdir <dir\_name>**
  - Removes the directory called <dir\_name>
  - Will result in an **error** if the directory is not empty
- **rm <file\_name|wildcard pattern>**
  - This will remove all files in the current folder with the specified name or that match the pattern
  - The recursive flag ‘-r’ allows for deleting all sub-directories and files
    - Double check when using, as you can wipe your hard drive with this command



```
markgrivainis — exec bash — exec — bash — 81x23
|bash-3.2$ ls
|bash-3.2$
|bash-3.2$ mkdir test
|bash-3.2$
|bash-3.2$ ls
|test
|bash-3.2$ touch test/a.txt
|bash-3.2$ tree
|   └── test
|     └── a.txt

1 directory, 1 file
|bash-3.2$ rmdir test
|rmdir: test: Directory not empty
|bash-3.2$ rm test/a.txt
|bash-3.2$ rmdir test/
|bash-3.2$
```

# Git – Project Setup

1. `cd ~/Desktop`

- ‘~’ refers to the current user's home directory
- Every user will have a home directory

2. `mkdir git_example`

3. `cd git_example`

4. `git init`

5. `git status`

6. `touch README.md`

7. `git status`

8. `git add README.md`

9. `git commit -m "Initial commit"`

```
markgrivainis — exec bash — exec — bash — 83x28
|bash-3.2$ cd ~/Desktop
|bash-3.2$ mkdir git_example
|bash-3.2$ cd git_example
|bash-3.2$ git init
|Initialized empty Git repository in /Users/markgrivainis/Desktop/git_example/.git/
|bash-3.2$ git status
|On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
bash-3.2$ touch README.md
|bash-3.2$ git status
|On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
bash-3.2$ git add README.md
|bash-3.2$ git commit -m "Initial commit"
|[master (root-commit) 934d0ab] Initial commit
  1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 README.md
bash-3.2$
```

# Git – Project Setup

11. `git status`

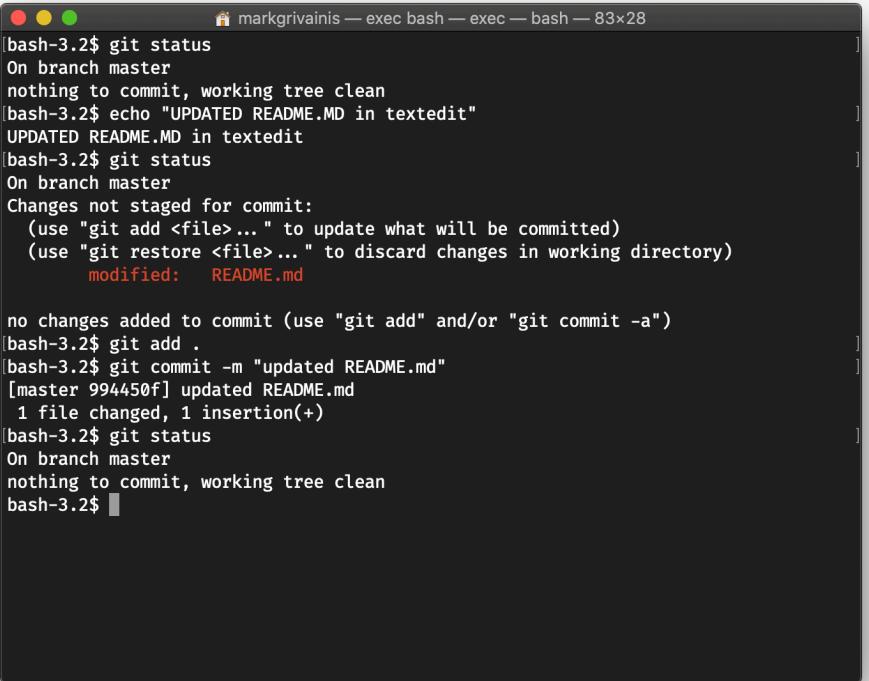
12. Make a change to `README.md` in a text editor

13. `git status`

14. `git add .`

15. `git commit -m "updated README.md"`

16. `git status`



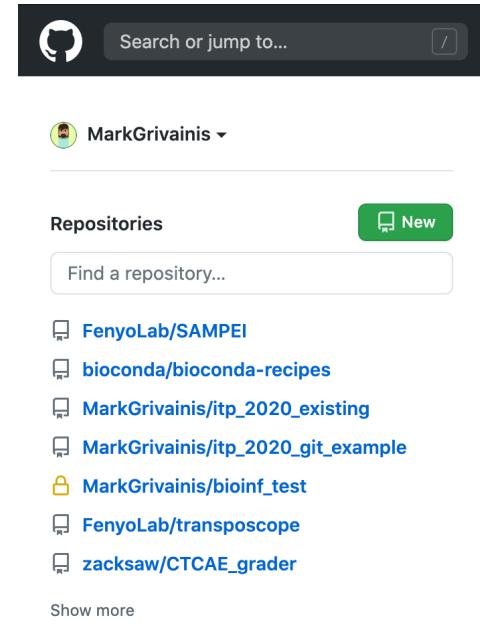
The screenshot shows a terminal window titled "markgrivainis — exec bash — exec — bash — 83x28". The terminal displays the following sequence of commands and their output:

```
|bash-3.2$ git status
On branch master
nothing to commit, working tree clean
|bash-3.2$ echo "UPDATED README.MD in textedit"
UPDATED README.MD in textedit
|bash-3.2$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
|bash-3.2$ git add .
|bash-3.2$ git commit -m "updated README.md"
[master 994450f] updated README.md
  1 file changed, 1 insertion(+)
|bash-3.2$ git status
On branch master
nothing to commit, working tree clean
bash-3.2$
```

# Git – Linking with a remote GitHub Repository

- Go to [github.com](https://github.com)
  - Create an account if you have not already
- Sign in
- Click new in the top left corner



# Git – Linking with a remote GitHub Repository

- Give the repository a name
  - I am using ‘bioinf\_git\_test’ for this example
- Optional: Give a description
- Public / Private options
  - This determines who can view the repository
- Initialize options
  - Leave these unchecked if you have already created a local git repository (this is what we did in the previous steps)

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

The screenshot shows the GitHub interface for creating a new repository. The top section has fields for 'Owner' (set to 'MarkGrivainis') and 'Repository name' (set to 'bioinf\_git\_test'). Below this is a note about repository names being short and memorable. The next section, 'Description (optional)', contains the text: 'This repo is being used to demonstrate the process of linking GitHub to a local repository'. The 'Public' radio button is selected, with the note: 'Anyone on the internet can see this repository. You choose who can commit.' The 'Private' radio button is also present with its note. The final section, 'Initialize this repository with:', contains three unchecked checkboxes: 'Add a README file' (with note: 'This is where you can write a long description for your project. [Learn more](#).'), 'Add .gitignore' (with note: 'Choose which files not to track from a list of templates. [Learn more](#).'), and 'Choose a license' (with note: 'A license tells others what they can and can't do with your code. [Learn more](#).'). At the bottom is a green 'Create repository' button.

Owner \* Repository name \*

MarkGrivainis / bioinf\_git\_test

Great repository names are short and memorable. Need inspiration? How about [ubiquitous-funicular](#)?

Description (optional)

This repo is being used to demonstrate the process of linking GitHub to a local repository

Public

Anyone on the internet can see this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

Initialize this repository with:

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

Add a README file

This is where you can write a long description for your project. [Learn more](#).

Add .gitignore

Choose which files not to track from a list of templates. [Learn more](#).

Choose a license

A license tells others what they can and can't do with your code. [Learn more](#).

Create repository

# Git – Linking with a remote GitHub Repository

The screenshot shows a GitHub repository page for 'MarkGrivainis / bioinf\_git\_test'. The 'Code' tab is selected. A 'Quick setup' section provides instructions for cloning the repository via Desktop, HTTPS, or SSH, and for creating a new file or uploading an existing one. It also recommends including a README, LICENSE, and .gitignore. Below this, a command-line setup section shows the steps to initialize a local repository and push it to GitHub. A red annotation 'We have already done these steps' points to the command-line instructions. Another red annotation 'Follow these instructions' points to the quick setup section.

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH git@github.com:MarkGrivainis/bioinf\_git\_test.git

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# bioinf_git_test" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M master
git remote add origin git@github.com:MarkGrivainis/bioinf_git_test.git
git push -u origin master
```

We have already done these steps

...or push an existing repository from the command line

```
git remote add origin git@github.com:MarkGrivainis/bioinf_git_test.git
git branch -M master
git push -u origin master
```

Follow these instructions

# Git – Linking with a remote GitHub Repository

The screenshot shows a GitHub repository page for 'MarkGrivainis / bioinf\_git\_test'. The repository has 1 branch and 0 tags. A recent commit by MarkGrivainis updated the README.md file 11 minutes ago. The README file contains the text: 'README for git example'.

**About**  
This repo is being used to demonstra

**Readme**

**Releases**  
No releases published  
Create a new release

**Packages**  
No packages published  
Publish your first package

© 2020 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

# Git – Updating the Local / Remote repository

- `git pull`
  - This will pull the latest versions of the files from the remote repository (GitHub)
- `git push`
  - This will push the latest local versions of the files to the remote repository (GitHub)
  - You will get an error if the local branch does not exist of the server and will need to use this command
    - `git push --set-upstream <remote_label> <branch_name>`
    - `git push --set-upstream origin test_branch`
    - `origin` is the default name of the remote repository

# Survey

- [https://nyumc.qualtrics.com/jfe/form/SV\\_6yztFdY5iQpnuBL](https://nyumc.qualtrics.com/jfe/form/SV_6yztFdY5iQpnuBL)