

GitHub





Git is a **distributed**, non-linear **version control** system

Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later. **Distributed** version control allows multiple copies to exist on multiple systems simultaneously.



“GitHub is a **collaboration** platform built on top of a distributed version control system called Git.”

GitHub provides **free*** remote hosting for Git repositories

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.



Vocabulary

- **Git** – A version-control protocol.
- **GitHub** – A service providing remote Git repo hosting.
- **“Repository” (“Repo”)** – A collection of code.
- **“Add”** – Process of staging changed files before...
- **“Commit”** – Process of creating a change record.
- **“Push”** – Uploading change records to remote host.
- **“Pull”** – Download change records from remote host.
- **“Clone”** – Download a personal copy of a code repo.
- **“Branch”** – A separate, working copy of a repository.
- **“Merge”** – Combining two branches into a single copy

Setting Up Git

Try Me!

1. Register for a GitHub Account

<http://github.com/join>

2. Install Git

<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

- Windows: Download GitHub for Windows (see below)
- Mac: Install Xcode Tools (run “`git`” from terminal)
- Linux: Use package-manager (i.e. `yum install git-all`)

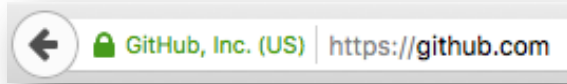
3. (OPTIONAL) Install GitHub Desktop

<http://mac.github.com>

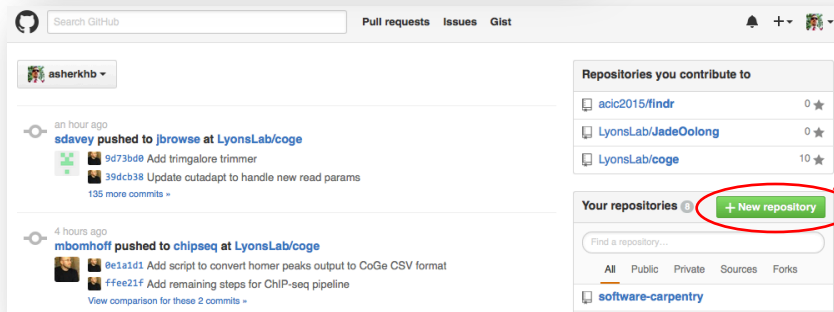
<http://windows.github.com>

Your First Repository in 1, 2, 3, 4

Try Me!



1. Navigate to github.com & Log In



2. Select "New Repository"

Create a new repository

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

Owner: asherknb / Repository name: new-repo ✓

Great repository names are short and memorable. Need inspiration? How about **super-winner**.

Description (optional): brand new repository to store my code

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

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

☐ 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 ⓘ

Create repository

3. Name & Describe Repository

NOTE: Because we are going to be importing an existing project, do not initialize with any files

Linking Project & GitHub

Try Me!

Initialize Project with Git

1. Navigate to project directory
2. Initialize Git
3. Add remote host URL

```
2. bash
Last login: Fri Jan 29 14:38:17 on ttys000
[~/] SeñorRift: cd Desktop/new-repo/
[~/Desktop/new-repo] SeñorRift: git init
[~/Desktop/new-repo] SeñorRift: git remote add origin https://github.com/asherkhb/new-repo.git
[~/Desktop/new-repo] SeñorRift: 
```

Check Status, Add & Commit Changes

Check Status
git status

Add New/Modified Files
git add <file>

```
2. bash
[~/Desktop/new-repo] SeñorRift: git status
On branch master

Initial commit

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

    myfile

nothing added to commit but untracked files present (use "git add" to track)
[~/Desktop/new-repo] SeñorRift: git add myfile
[~/Desktop/new-repo] SeñorRift: git status
On branch master

Initial commit

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

    new file:   myfile

[~/Desktop/new-repo] SeñorRift: git commit -m "added myfile"
[master (root-commit) 5c9545b] added myfile
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 myfile
```

Commit Changes

git commit -m "<msg>"

Linking Project & GitHub, Cont.

Try Me!

Uploading content to GitHub

Upload Content 
git push

```
[~/Desktop/new-repo] SeñorRift: git push
fatal: The current branch master has no upstream branch.
To push the current branch and set the remote as upstream, use

git push --set-upstream origin master
```

NOTE: Error Message!

Set Upload Path & Upload 

--set-upstream <host> <branch>

NOTE: Only necessary first
time, or on new branches
(more later...)

```
[~/Desktop/new-repo] SeñorRift: git push --set-upstream origin master
Username for 'https://github.com': asherkhhb
Password for 'https://asherkhb@github.com':
Counting objects: 3, done.
Writing objects: 100% (3/3), 205 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/asherkhb/new-repo.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
[~/Desktop/new-repo] SeñorRift: git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean
[~/Desktop/new-repo] SeñorRift: 
```

Downloading changes from GitHub

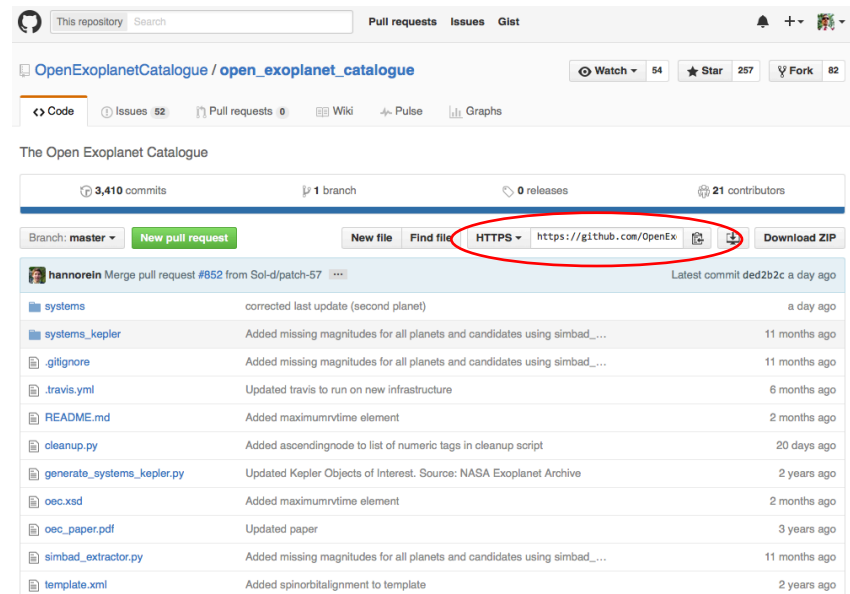
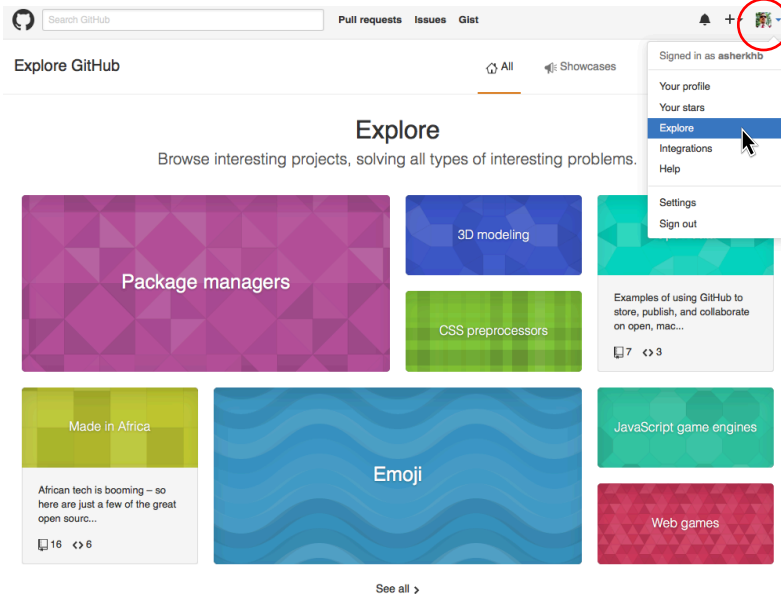
Download Changes 
git pull

```
[~/Desktop/new-repo] SeñorRift: git pull
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/asherkhb/new-repo
 5164674..763ff5f master    -> origin/master
Updating 5164674..763ff5f
Fast-forward
 README.md | 3 +++
 1 file changed, 3 insertions(+)
 create mode 100644 README.md
[~/Desktop/new-repo] SeñorRift: 
```


Obtaining Other Code

Try Me!

Finding New Repositories



Downloading New Repositories

Cloning a Repo
`git clone <url>`

```
2. bash
[~/Desktop] SeñorRift: git clone https://github.com/OpenExoplanetCatalogue/open_exoplanet_catalogue.git
Cloning into 'open_exoplanet_catalogue'...
remote: Counting objects: 64100, done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 64100 (delta 12), reused 0 (delta 0), pack-reused 64073
Receiving objects: 100% (64100/64100), 17.15 MiB | 9.11 MiB/s, done.
Resolving deltas: 100% (57250/57250), done.
Checking connectivity... done.
[~/Desktop] SeñorRift: 
```

Review

Setup

- Initializing a Project as Git..... `git init`
- Setting Remote Host..... `git remote add origin <url>`

Basics

- Checking Status..... `git status`
- Adding Changes..... `git add <file>`
- Committing Changes..... `git commit -m <message>`
- Uploading Changes..... `git push`
- Downloading Changes..... `git pull`
- Obtaining Additional Repos.. `git clone <url>`

Git & GitHub Resources

Interactive Lessons

<https://www.codecademy.com/learn/learn-git>

<https://try.github.io/>

Cheat Sheets

<https://training.github.com/kit/downloads/github-git-cheat-sheet.pdf>

<https://help.github.com/articles/good-resources-for-learning-git-and-github/>

<https://www.git-tower.com/blog/git-cheat-sheet/>

Pro-Git: A Comprehensive book on using Git (*free online*).

<https://git-scm.com/book/en/v2/>