

Git

What is Git?

Git is a version control system (VCS) for tracking changes in computer files. It tracks every single version and every single change that is made. That allows multiple developers to work on one single project at the same time and tracks every change and revision made by every developer. This also allows developers to revert back to any “commit” or revision which makes your code very safe.

Why do we need it?

- When working in teams, we want everyone to agree on, and work on, the most current version.
- It gives us a record (including a description and author name) of everything that has been written into our code based project.
- We don't want to delete any of our brilliant code!

Workflow



We can create a local repository (one sitting on our computer) or work with a remote repository hosted on a service such as GitHub, GitLab, or GitBucket.

To work locally, you can use the **git init** command to create a working directory. To ‘checkout’ a repository hosted on a remote server you would use the **git clone** command to create a local working directory.

The general workflow has three trees, or stages. The working directory holds all the files that you are currently working on. Those files get moved to the staging area with the **git add** command. Files sitting in the staging area are now tracked files waiting to be put into the new saved version (commit). You can make a commit, or a repository version, when you move everything sitting in the staging area to the final head with the **git commit -m** command. Once a commit is made the working tree is empty and ready to start all over again.

Installing Git

Check to see if you have git installed on your machine with the command **git --version**. If the command line does not find a version of git you can download and install it from here: <https://git-scm.com/downloads>

Command Line Prompts

Some common command line prompts that you will need to be familiar with are:

cd	changes directory
dir or ls	lists all the content in the current directory
git	shows all the available commands
git init	initializes the local git repository
git clone	brings the repository from GitHub to your desktop (checkout)
git status	shows the status of our working tree, and/or the difference from our local repository and the remote repository
git add filename	add that file to the staging area
git add -A	adds all the files to the staging area
git commit -m "..."	stores our changes with a messages describing what we've changed
git remote add origin URL	establishes where that repository is
git fetch	downloads the changes from the remote repository
git pull	downloads the changes from the remote repository and merges them into the current branch
git push	will sync our commits with the remote repository
git push -u origin	

Create a Local Repo

1. Start a new project folder.
2. Open the command line terminal and navigate to the new project folder.

```
Last login: Thu Feb  6 11:57:30 on ttys000
Bridgettes-iMac:~ bjconnell$ cd /Users/bjconnell/Desktop/new-project
Bridgettes-iMac:new-project bjconnell$
```

3. Initialize the project as a repository with the **git init** command.

```
Last login: Thu Feb  6 11:57:30 on ttys000
Bridgettes-iMac:~ bjconnell$ cd /Users/bjconnell/Desktop/new-project
Bridgettes-iMac:new-project bjconnell$ git init
Initialized empty Git repository in /Users/bjconnell/Desktop/new-project/.git/
Bridgettes-iMac:new-project bjconnell$
```

4. Start working on the project. Add an `index.html` and some content. Note the visual cues within VS code as this is an untracked file.
5. Run a **git status** command to see where these files are in the workflow.

```
On branch master

No commits yet

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

       index.html

nothing added to commit but untracked files present (use "git add" to track)
Bridgettes-iMac:new-project bjconnell$
```

6. Move that file to the staging area with **git add index.html**. The terminal doesn't confirm anything for us, so run another **git status** to see what happened.

```
Bridgettes-iMac:new-project bjconnell$ git add index.html
Bridgettes-iMac:new-project bjconnell$ git status
On branch master

No commits yet

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

       new file:   index.html
```

7. The file is green and it's sitting in the staging area. Keep working on the project. Add an external stylesheet and a new page. Move those new files to the staging area too.

```
[Bridgettes-iMac:new-project bjconnell$ git add -A
[Bridgettes-iMac:new-project bjconnell$ git status
On branch master

No commits yet

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

       new file:   css/styles.css
       new file:   index.html
       new file:   second-page.html
```

8. Finally, lets make our first commit. Make sure to include a message with a description as to what all these changes are.

```
[Bridgettes-iMac:new-project bjconnell$ git commit -m"new project with index.html
, second-page.html, and a stylesheet"
[master (root-commit) 5962b1b] new project with index.html, second-page.html, an
d a stylesheet
Committer: Bridgette <bjconnell@Bridgettes-iMac.hitronhub.home>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

3 files changed, 12 insertions(+)
create mode 100644 css/styles.css
create mode 100644 index.html
create mode 100644 second-page.html
Bridgettes-iMac:new-project bjconnell$ █
```

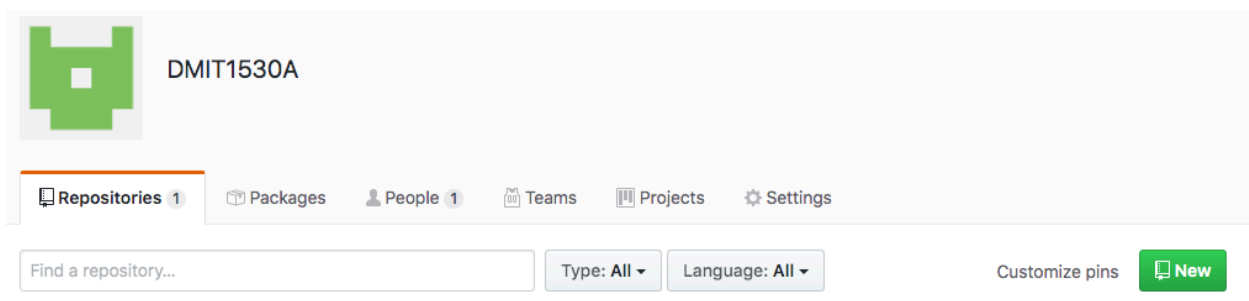
9. Check your working directory and you will see that it's now clean and empty.

```
Bridgettes-iMac:new-project bjconnell$ git status
On branch master
nothing to commit, working tree clean
Bridgettes-iMac:new-project bjconnell$ █
```

Move this project to GitHub



Create a GitHub account and remember your username and password (save them in a password manager).

Click the green "New" button to create a new repository.



Write in the repository name (following best naming standards) and give it a description. Since we are importing an existing repository **do not** initialize it or add a .gitignore.


Owner Repository name *


 DMIT1530A ▾ / new-project 

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

Description (optional)


My First Repo

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

 **Private**
Your current plan does not support private repositories. [Upgrade to Team](#)

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

We are going to push our repository from the command line. Copy these lines and paste them into the terminal.

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

```
git remote add origin https://github.com/DMIT1530A/new-project.git  
git push -u origin master
```



Refresh the page to see your repository on GitHub.

DMIT1530A / new-project Unwatch 1 Star 0 Fork 0


[Code](#) [Issues 0](#) [Pull requests 0](#) [Actions](#) [Projects 0](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)




My First Repo Edit

[Manage topics](#)

[1 commit](#) [1 branch](#) [0 packages](#) [0 releases](#) [0 contributors](#)

Branch: master [New pull request](#) [Create new file](#) [Upload files](#) [Find file](#) [Clone or download](#)

 **Bridgette** and **Bridgette** new project with index.html, second-page.html, and a stylesheet Latest commit 5962b1b 9 minutes ago

 css	new project with index.html, second-page.html, and a stylesheet	9 minutes ago
 index.html	new project with index.html, second-page.html, and a stylesheet	9 minutes ago
 second-page.html	new project with index.html, second-page.html, and a stylesheet	9 minutes ago

Help people interested in this repository understand your project by adding a README. [Add a README](#)