

git started!

Command line commands will be provided. Execute these while in the directory you cloned to.
GUI clients will follow the same general steps, but how to do them will depend on the client.

install

Install Git

GitHub Desktop: <https://desktop.github.com/>

Lots of other GUI clients exist as well. Look around for one you like!

Using Command Line:

Installing Git in Command Line:

If you're using command line, you may already have git installed. Check for an install with

```
git --version
```

Get Access

You will need to be given editing privileges for the repository.

Steps for getting access will depend on where the repository is kept.

Join GitHub: <https://github.com/join>

Join GitLab: https://gitlab.com/users/sign_up

Using ssh:

You'll be asked for credentials when you try to do something that requires them, so don't worry if you aren't sure where they come in just yet.

clone

Get Your Friend's Code

```
git clone https://link.to/repository.git
```

If you are able to ssh to another computer, you can clone from that computer - repositories don't necessarily have to be hosted on the web.

Creating Repos: <https://help.github.com/en/articles/create-a-repo>

```
git checkout branchname
```

Now Don't Break It



WARNING

Don't checkout filenames!
This discards your changes.

Talk to your teammates about which branch to work on. If no one has any preferences, it's safe to stick to the default (master) branch.

Branches are often used to ensure that certain kinds of breaks in code aren't sent to everyone - for example, if the master branch is where your program is released, you want to be sure that version of the program runs!

add
commit

Record Your Changes

<pre>git add filename</pre>	: Add a single file to the repository
<pre>git add *</pre>	: Add all files in current directory to the repo
<pre>git commit file</pre>	: Confirm your saved changes for a single file
<pre>git commit</pre>	: Confirm your saved changes for the repo

Send changes:

```
git push
```


Receive Changes:

```
git pull
```

Share Your Code!

```
git status
```

 will let you see how your files compare with the origin (the place the original repository is stored).

Pushing and pulling often will mean having to merge changes less!