

Intro to git: create a repository

If you need help at any time, put your **red** sticky note on the back of your laptop. When you've finished the steps on the *front* of this page, put your **green** sticky note on the back of your laptop. Then, you can turn the page over and try out some of the more advanced tricks on the back while you wait for the rest of the group to be ready.

Configure your git client

Let's start by telling git who you are. In a terminal, run

```
git config --global user.name "Your Name"
git config --global user.email "your.email@nyu.edu"
```

(substituting your own name and email address). Now git will sign every change you make to your code with your name and email address. This is very useful in a project with multiple contributors - you can assign credit (or blame) appropriately.

You only need to configure this once when you use git for the first time on a new computer.

Create a remote repository, clone locally

One way to start a new git repository is to create one using an online service, such as Github or Bitbucket, that offers free git hosting.

On Github, click on [New Repository](#), and supply a repository name. Check the box that says "Initialize this repository with a README"; you may also choose to add an open-source license to your project. If you're not sure what license to choose, the MIT License is a popular license for open source code that lets people do whatever they want with your code, as long as they 1) credit you, and 2) don't hold you liable if something bad happens because they used your code.

When you're ready, click [Create Repository](#). You should see your new repository, with a README.md file already in it, and also a LICENSE file if you chose to add a license when you created the repository.

Click on the button that says [Clone or Download](#). It should open a small pop-up titled "Clone with HTTPS", followed by a link that you should copy. (If it says "Clone with SSH", click on the small "Use HTTPS" link right next to it to switch back to HTTPS.)

Now, in a terminal, run `git clone` to get a copy of your repository onto your laptop:

```
git clone https://github.com/ffund/my-first-git-repo.git my-first-git-repo
```

The first argument to `git clone` is the HTTPS URL that you copied from the "Clone or Download" pop-up on the Github repository page. **Substitute the URL that you copied - don't use mine!** The second argument is the name of the directory into which git will put your project. (If you don't specify a directory name, it will use the name of the repository by default.)

Now, if you run

```
ls
```

you should see that a new directory has been created with the name that you specified in the `git clone` command. Use

```
cd my-first-git-repo
```

to enter that directory, and verify that it contains the README.md file.

Create locally, add remote

Sometimes you might want to use git with a project that *already exists on your local computer*.

To try this, create a new directory that is *not* inside an existing git repository (i.e. *don't* put it inside the directory in which you cloned your repository in the previous step - cd out of that directory first). Put a new file called `README.md` inside that new directory:

```
cd # go to the location where you want to keep your git repository
mkdir my-second-git-repo # make a new directory
cd my-second-git-repo    # enter the new directory
echo "Hello world" > README.md # create a README inside this directory
```

Now you have an existing project, and you'd like to start using git with it. From inside the folder, run

```
git init
```

You should see a message indicating that you have initialized an empty git repository. At this stage, you should be able to run any git commands that don't involve pushing or pulling code to and from a "remote". (A "remote" in git is just another place your repository is stored.)

To collaborate or share your work with others, you will want to store your repository online, using a service like Github or Bitbucket.

On Github, click on [New Repository](#), supply a repository name, and click [Create Repository](#). Don't add a README or LICENSE file.

On the next screen, Github will give you instructions regarding how to add a "remote" location to your existing repository.

Some additional notes:

- If you're not collaborating or sharing your code with anyone, and just want to use git for version control, you don't *need* to use a remote - you can just use git locally. However, you may want to use a hosting service like Github or Bitbucket anyways 1) to have an offsite backup of your work, and 2) because you may find some features of the web-based UI useful.
- If you use a local git repository for a while, then add a remote later, all of your past commits will appear in the remote. This means that any past versions can be retrieved from the remote, for example.