GitHub

CAPSTONE PROJECT

Reference: *githubtraining.github.io*

Getting Ready for the Project

Step 1: Set Up Your GitHub.com Account

You can set up your free account by following these steps:

1. Access GitHub.com and click Sign up.
2. Choose the free account.
3. You will receive a verification email at the address provided.
4. Click the link to complete the verification process.

If you already have an account, verify that you can visit github.com within your organization's network.

Step 2: Install Git

Download Git at www.git-scm.com

Where is Your Shell?

- If you are working on Windows, we recommend Git Bash which is installed with the Git package, so that you can follow along with the facilitator who will be using Bash.
- If you are working on a Mac or other Unix-based system, you can use the built-in Terminal application.

You may already have Git installed so let's check! Open Terminal if you are on a Mac, or PowerShell if you are on a Windows machine, and type:

```
$ git --version
```

You should see something like this:

```
$ git --version

git version 2.11.0
```

Step 3: Configuring Your User Name and Email

Git uses the config settings for your user name and email address to generate a unique fingerprint for each of the commits you create. You can't create commits without these settings:

```
$ git config --global user.name "First Last"
$ git config --global user.email "you@email.com"
```

Working Locally with Git

Using the command line, you can easily integrate Git into your current workflow.

Creating a Local Copy of the repo

Step 4: Try cloning with HTTPS

Open your chosen shell, and type:

```
git clone https://github.com/skill-curb/scratch.git
```

If the clone is successful you'll see:

```
$ git clone https://github.com/githubschool/scratch
Cloning into 'scratch'...
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), done.
```

Check `git status`

```
$ git status

On branch master

Your branch is up-to-date with 'origin/master'.

nothing to commit, working tree clean
```

`git status` is a command you will use often to verify the current state of your repository and the files it contains. Right now, we can see that we are on branch master, everything is up to date with origin/master and our working tree is clean.

Using Branches Locally

```
$ git branch
```

If you type `git branch` you will see a list of local branches.

```
$ git branch --all
$ git branch -a
```

If you want to see all of the branches, including the read-only copies of your remote branches, you can add the `--all` option or just `-a` .

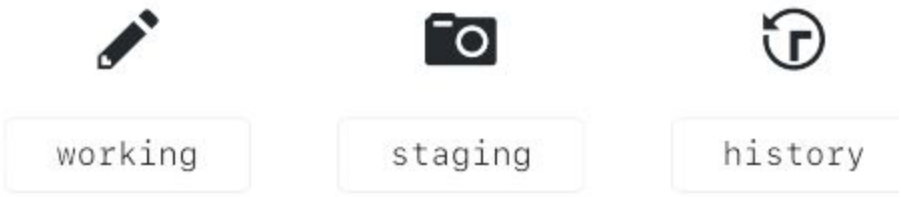
Switching Branches

```
$ git checkout <BRANCH-NAME>
```

To check out the branch you created online, type git checkout, and the name of your branch. Git will provide a message that says you have been switched to the branch and it has been set up to track the same remote branch from the origin.

The Two-Stage Commit

After you have created your file, it is time to create your first snapshot of the repository. When working from the command line, you will need to be familiar with the idea of the two-stage commit.



When you work locally, your files exist in one of four states. They are either:

- Untracked
- Modified
- Staged or,
- Committed.

In order to make a file part of the version-controlled directory, we will first do a git add and then we will do a git commit.

Let's do it now.

1. First, let's check the status of our working tree: `git status`
2. Create a new file in the working directory, type: `touch my-file.md`
3. Move the file from the working tree to the staging area: `git add my-file.md`
4. Let's see what happened: `git status`
5. Now let's take our first snapshot: `git commit`
6. Git will open your default text editor to request a commit message. Simply type your message on the top line of the file. Any line without a # will be included in the commit message.
7. Save and close the commit message
8. Let's take another look at our repository status: `git status`