**Getting ready for class**

While you are waiting for class to begin, please take a few minutes to set up your local work environment.

**Step 1: Set up your GitHub.com account**

For this class, we will use a public account on GitHub.com. We do this for a few reasons:

* We don't want you to "practice" in repositories that contain real code
* We are going to break some things, so we can teach you how to fix them (therefore, refer to the bullet above)

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.

GitHub is designed to run on the current versions of all major browsers. Take a look at our list of **supported browsers**.

**Step 2: Install Git**

Git is an open source version control application. You will need Git installed for this class.

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 2.33.1

In general, we recommend you stay within two minor versions of the [**latest version**](https://git-scm.com/downloads) of Git. The minor version is the second number in the series x.x.x.

We suggest having version [**2.29.2**](https://github.com/git-for-windows/git/releases/tag/v2.29.2.windows.2) or more recent if you're using Git for Windows or Git Bash.

**Downloading and installing Git**

If you don't already have Git installed, you can download Git at [**www.git-scm.com**](http://www.git-scm.com/).

If you need additional assistance installing Git, you can find more information in [**the ProGit chapter on installing Git**](http://git-scm.com/book/en/v2/Getting-Started-Installing-Git).

**Where is your shell?**

Now is a good time to create a shortcut to the command-line application you will want to use with Git:

* 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 macOS or another Unix-like system, you can use the built-in Terminal application

**Step 3: Try cloning with HTTPS**

Open your chosen shell, and type:

git clone https://github.com/githubschool/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.

If your clone is unsuccessful, read about [**authenticating with GitHub from Git**](https://docs.github.com/github/getting-started-with-github/set-up-git#next-steps-authenticating-with-github-from-git). Please note: many corporate networks restrict SSH traffic, so we highly recommend using HTTPS and verifying the clone works before class. Also, if you have two-factor authentication enabled and wish to use HTTPS, you will need to [**set up a personal access token**](https://docs.github.com/github/authenticating-to-github/accessing-github-using-two-factor-authentication#using-two-factor-authentication-with-the-command-line).

**Proxy configuration**

If your organization uses a proxy, you will need to configure the proxy settings in Git. Open Git Bash (on Windows) or Terminal (on Mac or \*nix) and complete the appropriate steps below:

**If your proxy does not require authentication:**

git config --global http.proxy https://YOUR.PROXY.SERVER:8080

Replace YOUR.PROXY.SERVER with your proxy's URL.

**If your proxy does require authentication:**

git config --global http.proxy https://YOUR\_PROXY\_USERNAME:YOUR\_PROXY\_PASSWORD@YOUR.PROXY.SERVER:8080

Replace YOUR\_PROXY\_USERNAME with the username used to authenticate into your proxy, YOUR\_PROXY\_PASSWORD with the password used to authenticate into your proxy, and YOUR.PROXY.SERVER with your proxy's URL.

**Step 4: Set up your text editor**

For this class, we will use a basic text editor to interact with our code. Let's make sure you have one installed and ready to work from the command-line.

**Step 5: Install GitHub CLI (optional)**

The GitHub command-line tool has funtionality that may be useful during the labs setup.

Install GitHub CLI from [**here**](https://github.com/cli/cli#installation), then confirm it is available on your console:

gh --version

**Pick your editor**

You can use almost any text editor, but we have the best success with the following:

* [**Visual Studio Code**](https://code.visualstudio.com/)
* Notepad
* Vi or Vim
* Sublime
* Notepad++
* GitPad

If you do not already have a text editor installed, go ahead and download and install one of the above editors now! You can also configure Visual Studio Code as your default text editor for Git commands using the [**instructions at docs.github.com**](https://docs.github.com/github/using-git/associating-text-editors-with-git).

**Your editor on the command-line**

After you have installed an editor, confirm you can open it from the command-line.

If installed properly, the following command will open the Visual Studio Code editor:

code .

**Exploring**

Congratulations! You should now have a working version of Git and a text editor on your system. If you still have some time before class begins, here are some interesting resources you can check out:

* [**github.com/explore**](https://www.github.com/explore) - Explore is a showcase of interesting projects in the GitHub Universe. See something you want to re-visit? Star the repository to make it easier to find later.
* [**lab.github.com**](https://lab.github.com/) - The Learning Lab bot will guide you through projects and provide feedback right from your GitHub repository, helping you build every step of the way.