## **SETTING UP GIT**

### **Introduction**

Git is a very popular version control system. You’ll become very familiar with this piece of software throughout TOP, so don’t worry too much about understanding it at this point. There are multiple lessons focused on Git later in the curriculum.

GitHub is a service that allows you to upload your code using Git and to manage your code with a nice web interface. GitHub and Git are not the same thing or even the same company.

### **Step 1: Install Git**

Click the Operating System you have chosen below:

**Linux**

#### **Step 1.1: Update the system**

Run these commands in the terminal to update the Linux system:

sudo apt update sudo apt upgrade

#### **Step 1.2: Install git**

It’s likely you have git installed already, but to make sure that we have the most up to date version of git, run the following commands:

sudo add-apt-repository ppa:git-core/ppa sudo apt update sudo apt install git

#### **Step 1.3: Verify version**

Make sure your git version is at least 2.28 by running this command:

git --version

If the version number is less than 2.28, follow the instructions again.

### **Step 2: Configure Git and GitHub**

#### **Step 2.1: Setup Git**

For Git to work properly, we need to let it know who we are so that it can link a local Git user (you) to GitHub. When working on a team, this allows people to see what you have committed and who committed each line of code.

The commands below will configure Git. Be sure to enter your own information inside the quotes (but include the quotation marks)!

git config --global user.name "Your Name" git config --global user.email "yourname@example.com"

GitHub recently changed the default branch on new repositories from master to main, change the default branch for Git using this command:

git config --global init.defaultBranch main

To enable colorful output with git, type

git config --global color.ui auto

To verify things are working properly, enter these commands and verify that the output matches your name and email address.

git config --get user.name git config --get user.email

#### **Step 2.2: Create a GitHub Account or Sign In**

Go to [GitHub.com](https://github.com/) and create an account! If you already have an account, sign in. You do not need to use the same email address you used before, but it might be a good idea to use the same one to keep things simple.

#### **Step 2.3: Create an SSH Key**

An SSH key is a cryptographically secure identifier. It’s like a really long password used to identify your machine. GitHub uses SSH keys to allow you to upload to your repository without having to type in your username and password every time.

First, we need to see if you have an SSH key already installed. Type this into the terminal:

ls ~/.ssh/id\_rsa.pub

If a message appears in the console containing the text “No such file or directory”, then you do not yet have an SSH key, and you will need to create one. If no message has appeared in the console output, you already have a key and can proceed to step 2.4.

To create a new SSH key, run the following command inside your terminal. The -C flag followed by your email address ensures that GitHub knows who you are.

Note: The angle brackets (< >) in the code snippet below indicate that you should replace that part of the command with the appropriate information. Do not include the brackets themselves in your command. For example, if your email address is odin@theodinproject.com, then you would type ssh-keygen -C odin@theodinproject.com. You will see this convention of using angle brackets to indicate placeholder text used throughout The Odin Project’s curriculum and other coding websites, so it’s good to be familiar with what it means.

ssh-keygen -C <youremail>

* When it prompts you for a location to save the generated key, just push Enter.
* Next, it will ask you for a password; enter one if you wish, but it’s not required.

#### **Step 2.4: Link Your SSH Key with GitHub**

Now, you need to tell GitHub what your SSH key is so that you can push your code without typing in a password every time.

First, you’ll navigate to where GitHub receives our SSH key. Log into GitHub and click on your profile picture in the top right corner. Then, click on Settings in the drop-down menu.

Next, on the left-hand side, click SSH and GPG keys. Then, click the green button in the top right corner that says New SSH Key. Name your key something that is descriptive enough for you to remember where it came from. Leave this window open while you do the next steps.

Now you need to copy your public SSH key. To do this, we’re going to use a command called [cat](http://www.linfo.org/cat.html) to read the file to the console. (Note that the .pub file extension is important in this case.)

cat ~/.ssh/id\_rsa.pub

Highlight and copy the output, which starts with ssh-rsa and ends with your email address.

Now, go back to GitHub in your browser window and paste the key you copied into the key field. Then, click Add SSH key. You’re done! You’ve successfully added your SSH key!

#### **Step 2.5 Testing your key**

Follow the directions in [this article from GitHub](https://help.github.com/en/articles/testing-your-ssh-connection) to verify your SSH connection (Don’t forget to omit the $ when you copy and paste the code!). You should see this response in your terminal: Hi username! You’ve successfully authenticated, but GitHub does not provide shell access. Fear not the part about not being provided shell access. If you see this message, you’ve successfully added your SSH key and you can move on. If the output doesn’t correctly match up, then try going through these steps again or come to [the Discord chat](https://discord.gg/fbFCkYabZB) to ask for help.

### **Step 3: Let us know how it went!**

You’ve completed the basic installations section, good job! As you progress through the Paths there will be other tools to install, so keep an eye out!

You probably felt like you were in way over your head, and you probably didn’t understand much of what you were doing. That’s 100% normal. Hang in there. You can do this! And we’ve got your back.