3.1.1 The Command Line

Tom's manager, Seth, would like him to become familiar with the command line, which will be used to make updates to the GitHub repository, access local files and folders, and write and run Python programming scripts. Using the command line is a common practice among programmers, and you want to make a great impression. In this section, you'll get up to speed on the command line in order to help Tom with the election analysis.

Software developers and programmers typically use the command line to navigate and perform tasks on their computers. We'll be using the command line to access files and folders as well as update GitHub repositories, and starting the Python interpreter to write and run Python scripts.

How you access the command line depends on what operating system you're using. If you use macOS, you'll use the Terminal application to access the command line; if you have a computer running Windows, you'll use the Command Prompt. However, for cloning and updating GitHub repositories, you'll need to use Terminal for macOS or Git Bash for Windows.

Git Bash is an application for Microsoft Windows environments that installs Bash and Git on a Windows operating system. If you have a computer running Windows, we'll install Git Bash later. Bash is an acronym for "Bourne-again shell." A **shell** is a terminal application, like Terminal on macOS, used to interface with an operating system through written commands.

NOTE

> In this module, we'll be using the term **command line** to refer to both the Terminal application for macOS and Command Prompt for Windows.

Check out the macOS instructions below, or jump to the **Windows instructions**.

---

macOS

Opening the Terminal Application

On a Mac, the command line is accessed via the Terminal application. To open the Terminal application, complete the following steps:

1. Press Command+Space to open Spotlight Search.

2. Type "Terminal" into the search bar and press Enter.

3. Next, select the Terminal application to launch it. You can save Terminal to your dock for easy access, as we'll be using it often.

Using the Terminal Application

After we have launched the Terminal application using the previous instructions, we'll see an empty window. However, if you've logged in previously, you'll probably see the date and time of your most recent login. For example:

```
Last login: Thu May  2 11:27:18 on ttys001
Toms-MBP:~ tom$
```

At this point, the computer is waiting for further instruction. Let's break down what we see in this window.

The first line shows the details of the last login. The second line provides the name of the computer (in this case, `Toms-MBP)`. The second part, `~ tom$`, has three key parts:

1. The squiggly line, `~`, is how the computer lets us know we are in the home directory.
2. The name "tom" is the name of the **home directory**. The home directory on a Mac is the little house that appears in the sidebar of the Finder window under Favorites.

3. The `$` is what programmers call the **prompt**, which is where we will type commands.

SHOW PRO TIP

NOTE

> Throughout this module, we'll be working on Tom's computer, since we're helping him with the election analysis. Therefore, images and code will show directory naming that is specific to Tom's computer. When you follow the steps on your computer, you'll see naming that corresponds to your specific machine.

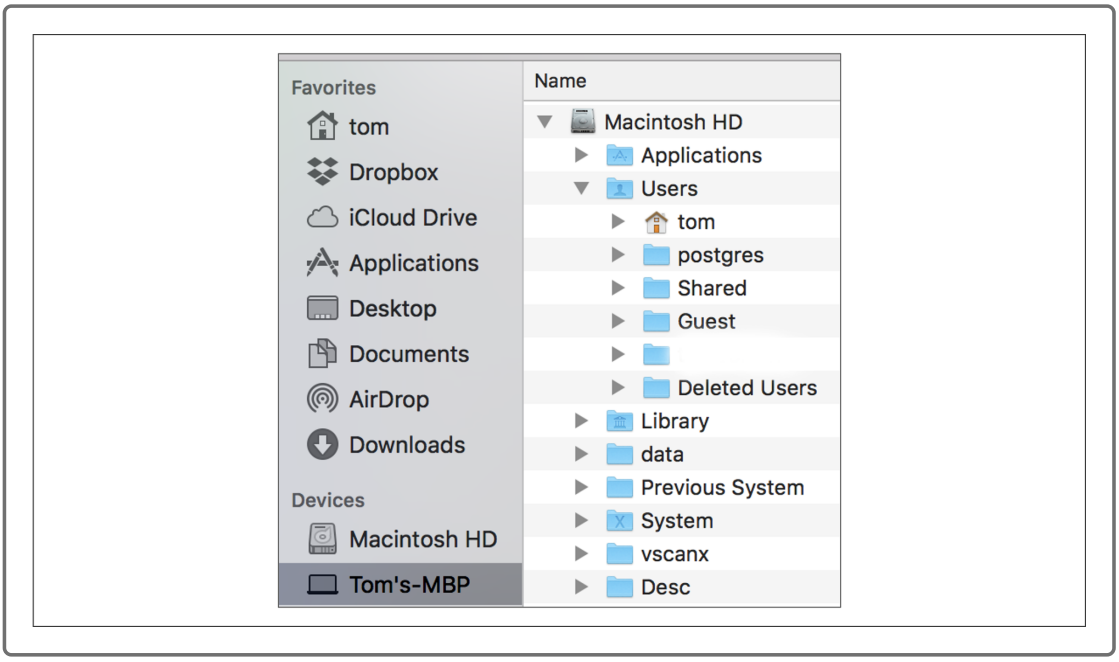
Finding Your Home Directory

Every time you open the terminal application, it will open in the home directory. Using commands, we can find the terminal "path" to the home directory.

To find the path to your home directory (or any directory), enter the command `pwd` after the prompt, `$`. Then press Enter to run, or execute, the command. `pwd` stands for "print working directory." The **working directory** is the current directory you are working in. For example, right now we are working in the home directory.

```
Toms-MBP:~ tom$ pwd
```

After we run the `pwd` command, the computer responds with `/Users/tom` on Tom's computer. This means that we're currently in the folder "tom," which is in the Users folder.



To find where `/Users/tom` is located on Tom's computer, we need to do the following:

1. Open the Finder window.
2. Locate "Devices." This may be listed under Favorites. For example, on Tom's computer, the name of the device is "Toms-MBP".
3. Find the Users folder. Click on that folder to find the "tom" folder. This is the folder we are in when we see `Toms-MBP:~ tom$` in the command line.

You probably view files and folders on your computer by using the user interface and clicking on a folder name. With the command line, we can navigate to any folder on our computer by typing the appropriate commands.

The command to view files and folders in the terminal is `ls`, which means "list files." If we type `ls` after the prompt and press Enter, the terminal will print the folder names in alphabetical order from top to bottom and right to left. These are the folders in Tom's home directory:

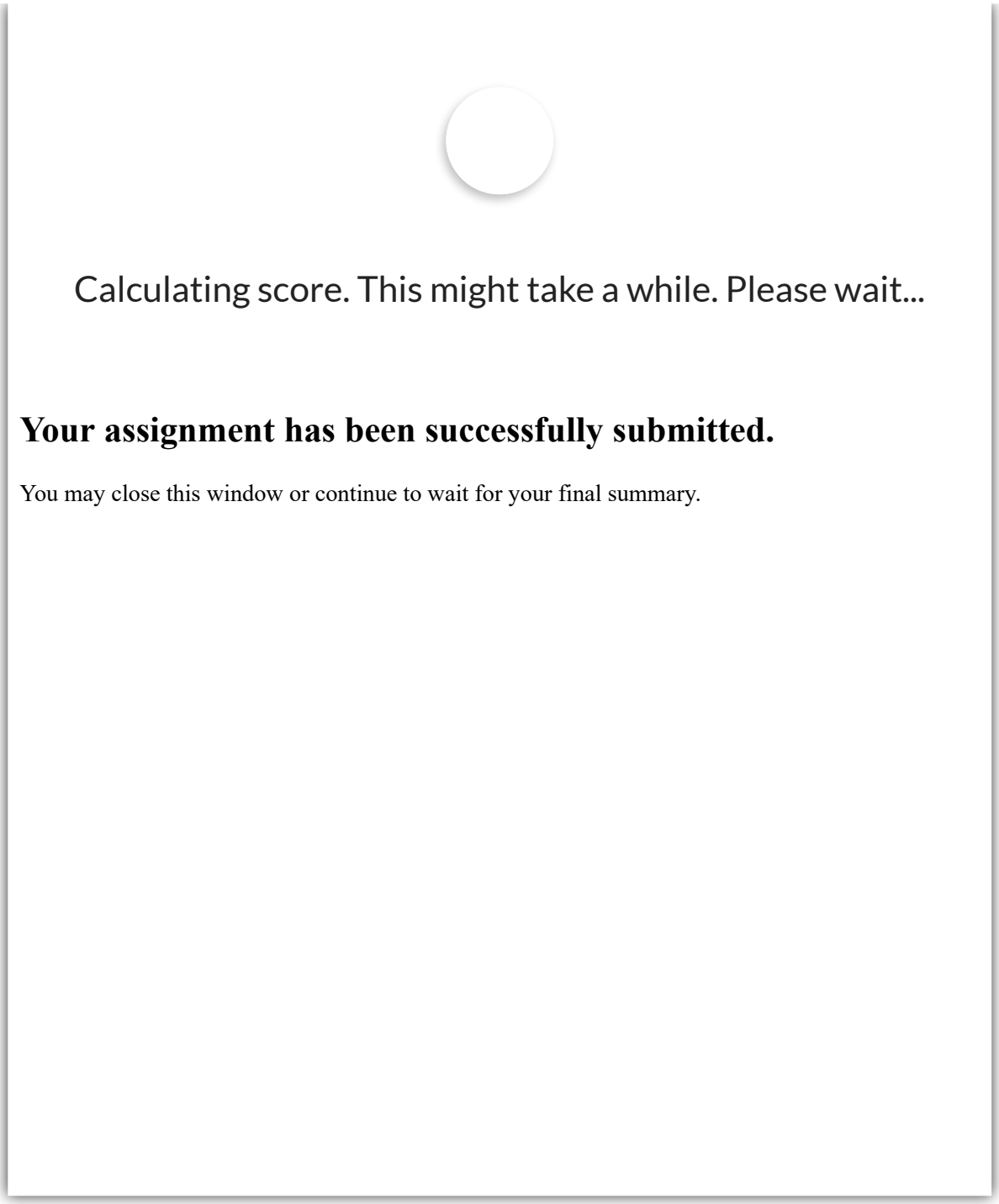
```
Applications Downloads Pictures
Desktop      Movies    Public
```

```
Documents    Music
```

Navigating with the Terminal Application

Let's say Tom wants to navigate to the Downloads folder. To do this, type `cd Downloads` after the prompt. The `cd` command means "change directory," and "Downloads" is the folder name. After typing the command, press Enter. The terminal returns the following:

```
Toms-MBP:~ Downloads tom$
```



If Tom wants to view folders on the Desktop, he would need to tell the computer to return to the home directory and then "change directory" to Desktop. (Remember, the Desktop folder is in the home directory, or the "tom" folder.)

If Tom's current working directory was `Toms-MBP:~ Downloads tom$`, then he would need to type `cd ..` after the prompt and press Enter. The two dots after `cd` tells the computer to go back one level, which, in our case, is the

home directory. Next, he would type `cd Desktop` and press Enter. Now Tom would be in his Desktop directory, or folder.

An alternative to navigating to the Desktop folder for Tom would be for him to type `cd ~` after the prompt. Remember, the "~" line means the home directory. It's like a magic command! No matter where you are in the command line, if you type `cd ~`, you'll be taken back to the home directory.

A technique used by more experienced programmers is to type `cd ../Desktop` after the prompt and then press Enter. This command allows you to go back one directory, or folder, level from the Downloads directory to the home directory, and then navigate to the Desktop folder. Let's break this command down in detail.

- The `cd ..` tells the computer to go back up one level.
- The forward slash "/" is a path separator.
- After the forward slash, we add the directory or folder, `Desktop`.

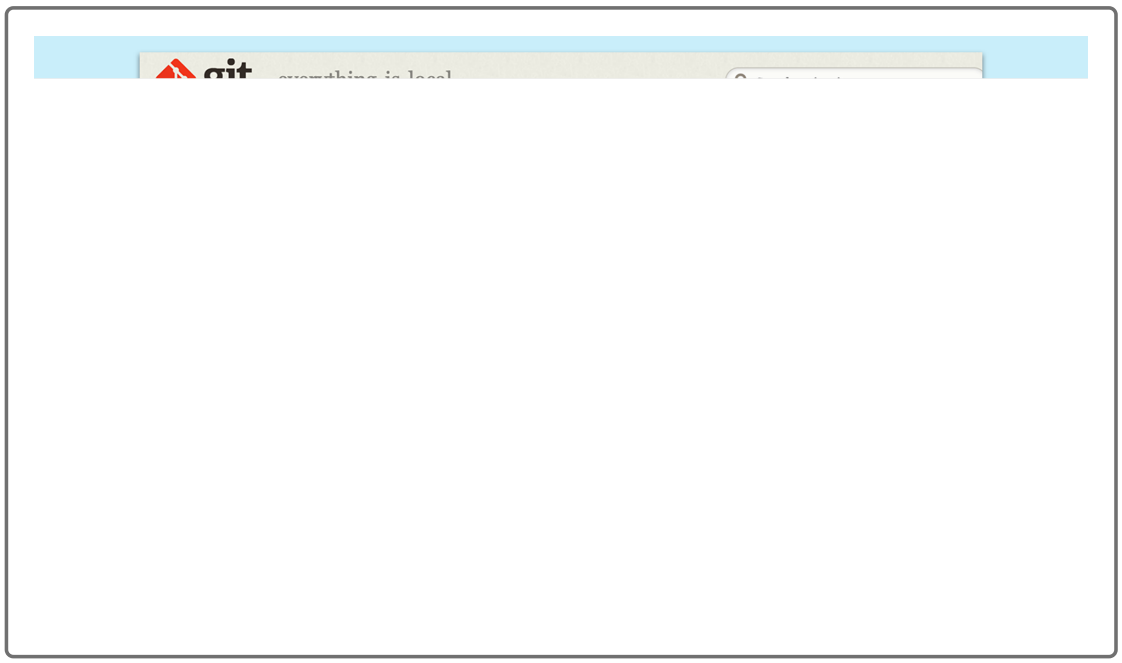
CAUTION

> If you misspell the name of the directory when performing any of these commands, or type the name of a folder that is not in a directory, you will get an error that looks something like this:
>
> `-bash: cd: ../Documents: No such file or directory`
>
> If you misspell something on your first attempt, type "ls" and then press Enter to find the correct spelling.

Download and Install Git and Git LFS

In order to interact with and use GitHub, we need to download the Git software. There are several options to install Git on a Mac as indicated on the **Git download page (https://git-scm.com/download/mac)**.

We will use the first option, `brew install git`.



If you already have Git installed skip to step 4 and 5.

First, to install Git using `brew` we'll need to install Homebrew.

1. Go to the **Homebrew (https://brew.sh/)** webpage and copy the script listed under "Install Homebrew" or copy the following script.

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/instal
```

2. Paste the script into your command line and press enter.

3. After Homebrew has been installed type and run `brew install git`.

4. Next, check the version of Git by typing `git --version`.

- The output should be `git version 2.31.1` or greater.

5. If you have an older version of Git you can upgrade Git by typing and running `brew upgrade git` in the command line.

Finally, we'll install Git LFS (Large File Storage). Git LFS decreases the upload and download time for large files (>100MB) to and from GitHub.

6. To install Git LFS type and run `brew install git-lfs` in the command line.

7. Check the version of Git LFS by typing `git-lfs --version`.

    - The output should be `git-lfs/2.13.3` or greater.

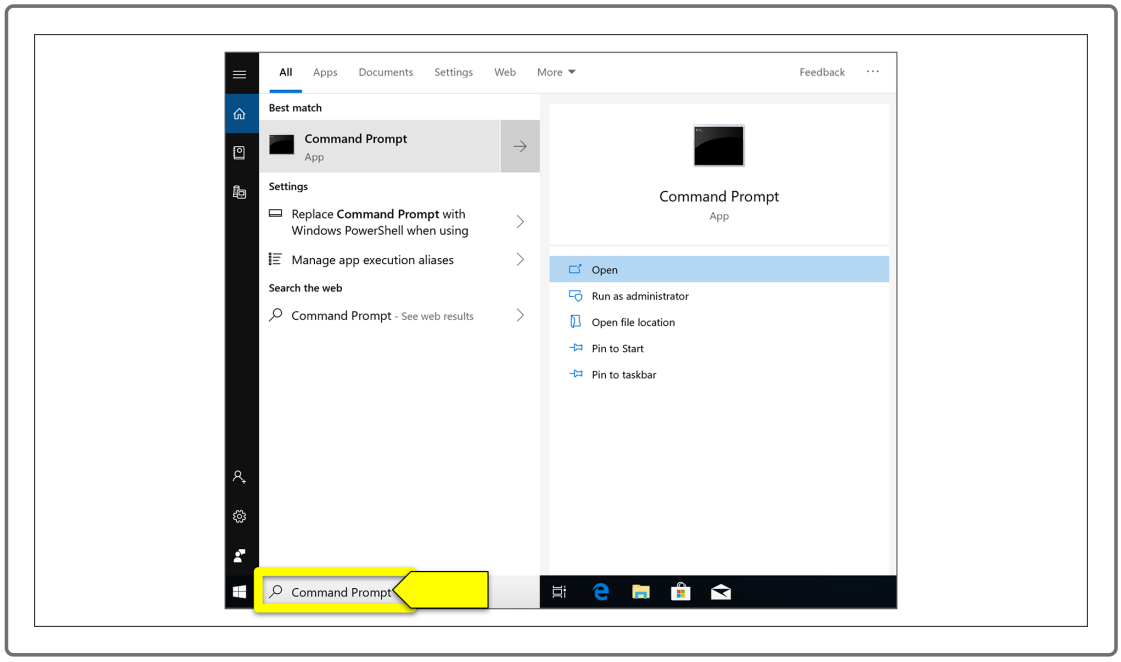
Once you've installed Git and Git LFS you are ready to interact with and use GitHub.

Windows

Opening the Command Prompt

On a PC, the command line is accessed via the Command Prompt. If you are using Windows 8 or 8.1, click the search icon in the top right of your screen and type "Command Prompt." You will probably see the Command Prompt application appear before you finish typing the name. Click on the Command Prompt application to launch it.

If you are using Windows 10, type "Command Prompt" in the search bar. Click on the Command Prompt application to launch it.



Using the Command Prompt

Launch the Command Prompt on your machine. You can pin it to the taskbar for easy access, as we will be using it often. You'll likely have an empty window that looks like this:

At this point, the computer is awaiting further instructions. Let's take a closer look at this window; specifically, turn your attention to the line that says `C:\Users\tom >`.

`Users\tom` tells us the directory we're in. Currently, we're in the "tom" folder, which is located in the Users folder.

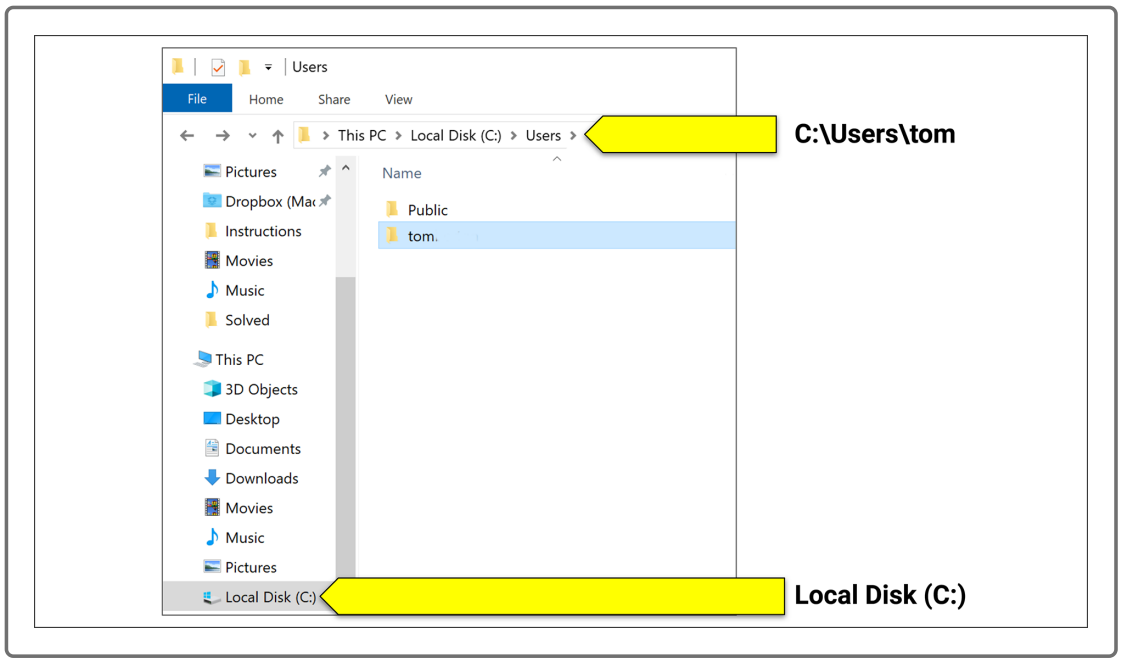
At any time, you can find the current directory with the command `chdir`. If you type this command after `>`, the Command Prompt will look similar to this:

```
C:\Users\tom >chdir
C:\Users\tom
```

```
C:\Users\tom >
```

SHOW PRO TIP

To locate `/Users/tom` on our computer, open the Desktop folder and click Local Disk (C:). Then, open the Users folder, where we'll find the "tom" folder. This is the folder we're in when we see `/Users/tom` in the Command Prompt.



Navigating with the Command Prompt

You probably use your computer's user interface to view files and folders by double-clicking on a particular folder to see the contents. But you can also view all of your computer's folders and files using the command line, specifically, the `dir` command. Entering this command after the prompt will display a list of files and folders.

Type `dir` and press Enter. The terminal will return a list of folders along with a timestamp for the last time the folder or directory was accessed.

We can also use the command line to navigate to any folder on our computer simply by typing the appropriate command.

For example, if Tom is in the current directory, `C:\Users\tom\Downloads` and he wants to find which folders are on the Desktop, he needs to:

1. Tell the computer to return to the home directory.
2. "Change directory" to Desktop, because the Desktop folder is inside the "tom" folder.

Let's try this.

Type `cd ..` after the prompt and press Enter. The two dots tell the computer to go back one level, which, in our case, is `C:\Users\tom`. Then, type `cd Desktop` and press Enter. We are now in the Desktop directory.

As you become more proficient in using the Command Prompt, you will be able to combine commands instead of executing them separately. Here are two other methods for executing the commands to navigate to the Desktop folder from the Downloads folder.

1. At the prompt type `cd ..\Desktop` after the prompt. This will take you to the Desktop folder, which is back one directory. The backslash, "\", is a path separator. The directory name, `Desktop`, is included after the backslash.
2. If you know your folder structure, type `cd C:\Users\tom\Desktop` after the prompt to go directly to the Desktop folder.

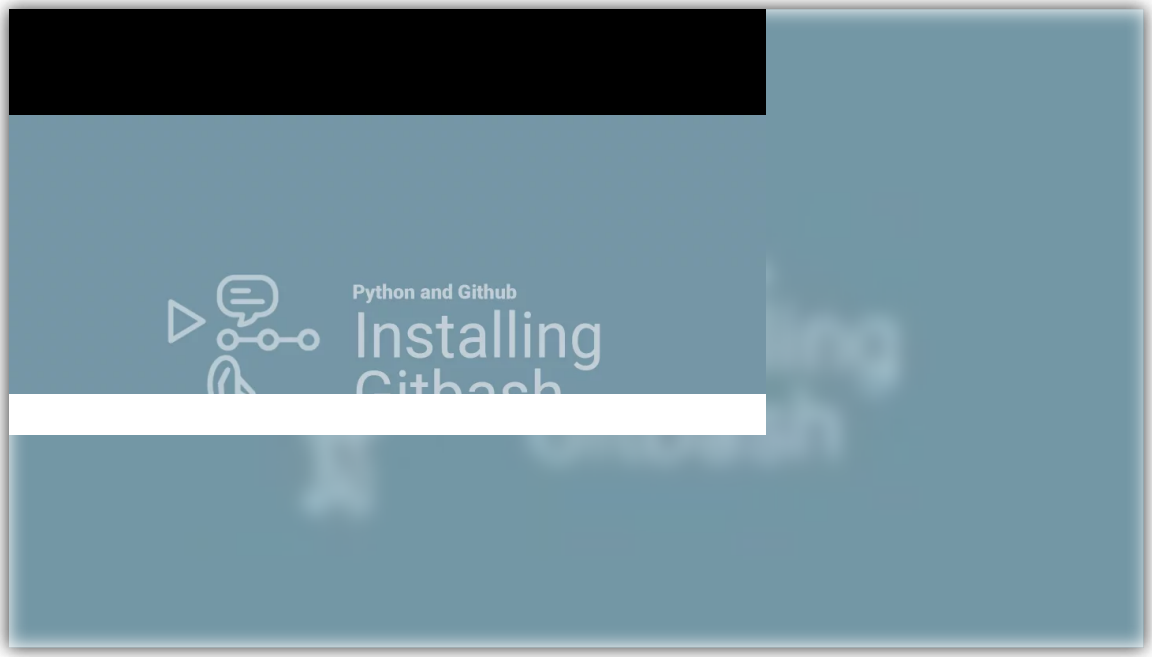
Download Git Bash and Git LFS

When it comes to using GitHub on a Windows PC, we can't use the Command Prompt. Git uses commands for Linux or macOS, so these commands will not work with the Command Prompt. This is where Git Bash comes to the rescue!

To use Git on Window we will need to install the Git Bash, often called "Bash" for short. Programmers depend on this tool for **version control,**

which is the process of logging the development of programs and applications. This comes in handy during collaborative programming, when teams of programmers change, add, and remove code throughout a project's directory. This process would be chaotic without Git.

Download the latest version of **Git Bash** **(https://git-scm.com/downloads)** for Windows and install it on your machine using the following steps:



NOTE

> The version of Git will change often, but the link will always specify which version of Git you are downloading.

Next, we'll install Git LFS (Large File Storage). Git LFS decreases the upload and download time for large files (>100MB) to and from GitHub.

Launch Git Bash and type and run `git lfs install` at the prompt.

Once you've installed Git Bash and Git LFS you are ready to interact with and use GitHub.

Navigating with Git Bash

Navigating through directories with Git Bash is the same as navigating with the Command Prompt. A common command is `pwd` ("print working directory"), which is used to find our current location, or working directory. Type the command and press Enter. The output is the following:

```
/c/Users/tom
```

This is similar to using `dir` with the Command Prompt. However, we are not getting all the files in the directory. For this, use the `ls` command.