

Setting Up Your Command Line

The command line is a powerful tool used by developers to find, create, and manipulate files and folders. This short tutorial will walk you through the steps for setting up the command line application on your computer.

Command Line Interfaces (CLIs) come in many forms. The CLI we'll use is called Bash.

WHAT IS BASH?

Bash, or the **B**ourne-**A**gain **S**Hell, is a CLI that was created over twenty-seven years ago by Brian Fox as a free software replacement for the Bourne Shell. A **shell** is a specific kind of CLI. Bash is "open source" which means that anyone can read the code and suggest changes. Since its beginning, it has been supported by a large community of engineers who have worked to make it an incredible tool. Bash is the default shell for Linux and Mac. For these reasons, Bash is the most used and widely distributed shell. If you want to learn more about Bash, [this Wikipedia article](#) is a good place to start.

BASH SETUP FOR MAC AND WINDOWS

MAC USERS:

As mentioned before, Bash is the default shell on Linux and Mac OS X, so good news, you don't have to install anything!

To access Bash in OS X, you can use an application called ***Terminal***.

1. First open the **Applications** folder, then open the **Utilities** folder.
2. Once you're in the **Utilities folder** you will see the application **Terminal**. Open the **Terminal** application and you're ready to go!
3. For ease of access later, you can keep Terminal in your Dock. Simply right click (alt-click) the Terminal icon in your dock, then select "Options", then "Keep In Dock."

Continue to the "Try it Out!" section below for some simple first steps with your new tool.

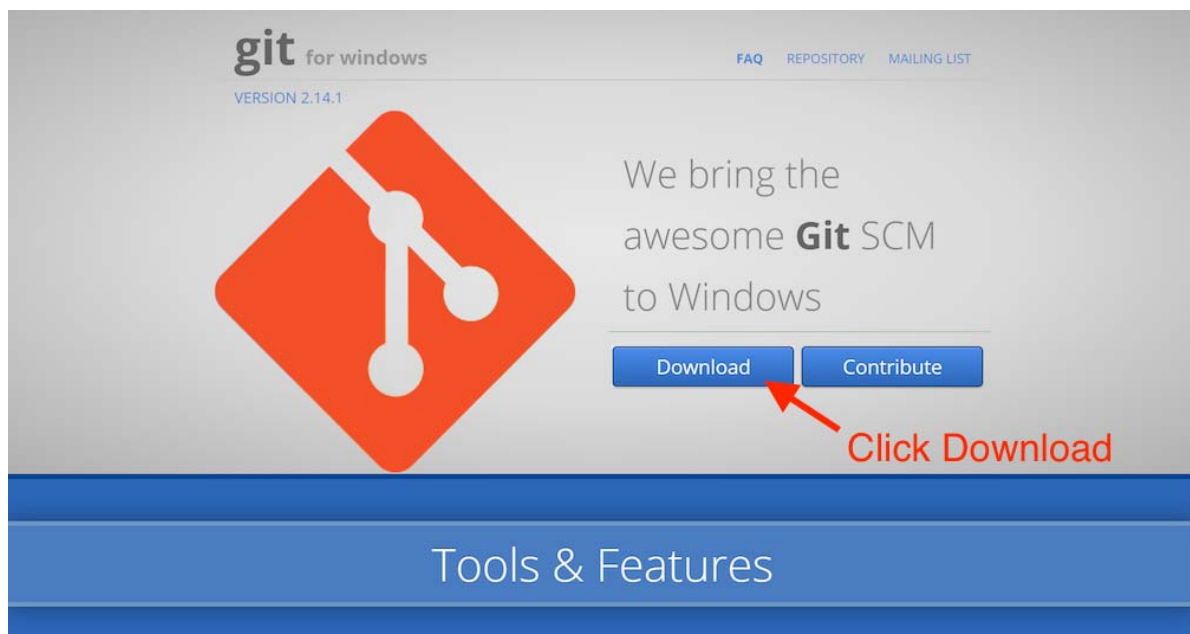
WINDOWS USERS:

Windows has a different CLI, called **Command Prompt**. While this has many of the same features as Bash, Bash is much more popular. Because of the strength of the open source community and the tools they provide, mastering Bash is a better investment than mastering Command Prompt.

To use Bash on a Windows computer, we will download and install a program called **Git Bash**. Git Bash allows us to easily access Bash as well as another tool we'll be using later called Git, inside the Windows environment.

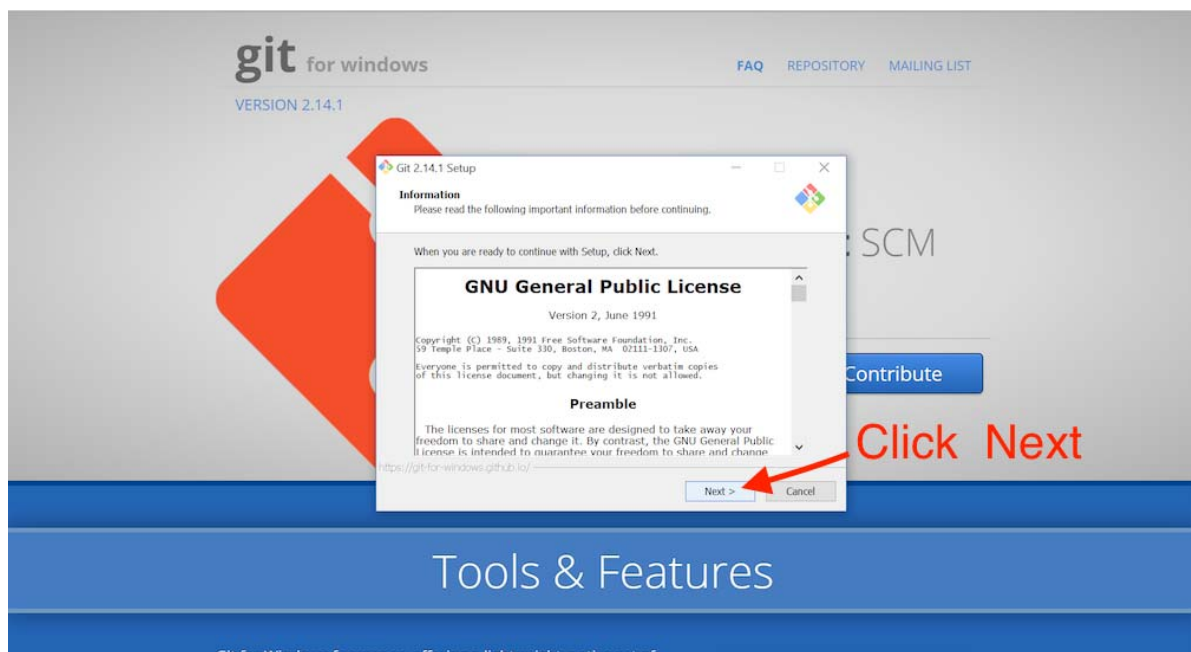
HOW TO INSTALL GIT BASH:

1. Navigate to the [Git Bash installation page](#) and click the Download button.

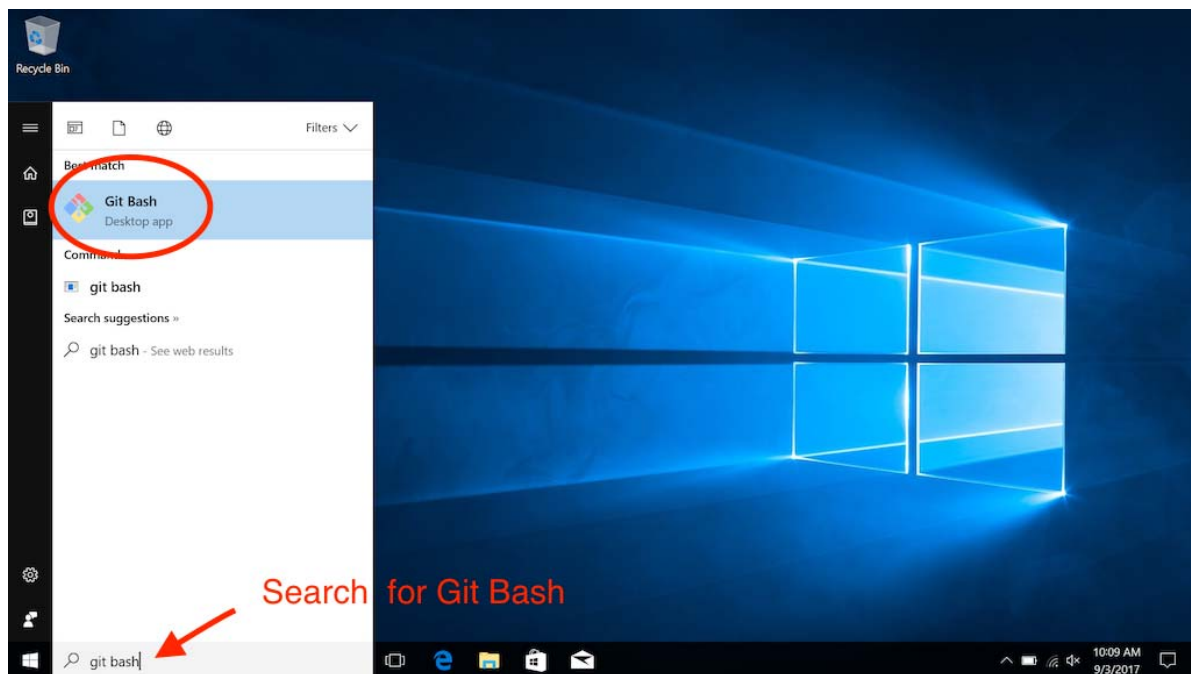


2. Once Git Bash is downloaded, run the downloaded **.exe** file and allow the application to make changes to your PC. You will get a prompt that says "Do you want to allow this app to make changes to your device?" Click **Yes**.

3. To keep things simple, we will use the default settings for everything in this installation, so all you need to do now is keep clicking **Next**, and finally **Finish**.



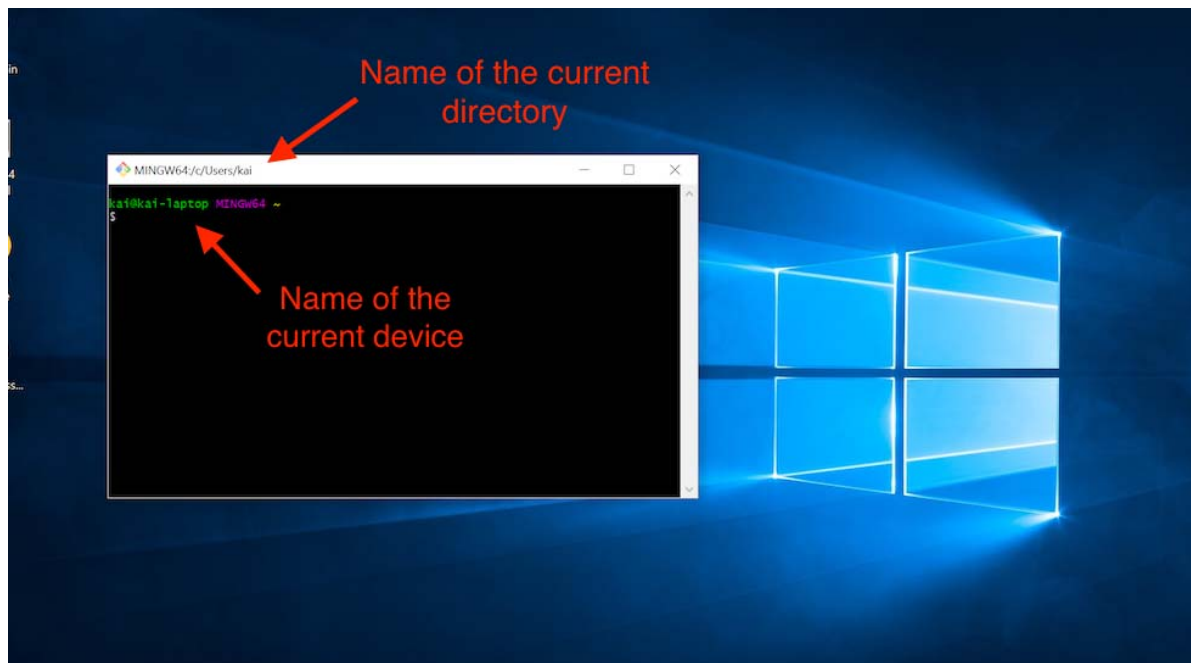
4. Open the Start menu by clicking on the Windows icon and typing "Git Bash" into the search bar. The icon for Git Bash and the words "Git Bash Desktop App" will appear. Click on the icon or the words "Git Bash Desktop App" to open Git Bash.



5. A new window will open. This is the Git Bash CLI where we will run Bash commands. Whenever a new window of the Git Bash app is opened, you will always be placed in the same directory, your **home directory**.

The home directory is represented by the tilde sign, `~`, in the CLI after `MINGW64`. The tilde is another way to say `/c/Users/username` in Git Bash or `C:\home\Users\username` in Windows' Command Prompt.

The absolute path of your current working directory, how you got from the root directory to the directory you are currently in, will always be noted at the top of the window:



Git Bash works by giving you a CLI that acts like a Bash CLI. That means you can now work with your files and folders using Bash commands instead of Windows commands.

Congratulations, you now have Bash installed on your computer, ready to use!

TRY IT OUT!

Now that you have your Command Line Interface open on your desktop, you are ready to use it. Go ahead and try some of the commands on your personal computer. Here are some good commands for practice:

`ls` to list the contents of the current directory

`mkdir test` to make a new directory named **test**

`cd test` to navigate into the new directory

`echo "Hello Command Line" >>`

`hello_cli.txt` to create a new file
named **hello_cli.txt** and add **Hello Command
Line** to that file

`cat hello_cli.txt` to print the contents of
the **hello_cli.txt** file to the terminal

Good job! You're ready to explore the world of the Command Line Interface on your own computer.

Completed