

# Setting Up Conda in Git Bash

If you're a Windows user following along with the Data Science or Computer Science career paths on Codecademy (or the Learn Python courses), you may have noticed the [recommendation to use Git Bash](#) for your terminal, and the [instructions on installing Python via Miniconda](#), but the lack of guidance on using conda within the Git Bash terminal rather than CMD or Anaconda Prompt.

If you use and enjoy Git Bash and want to take advantage of the power of conda without switching terminals, follow the instructions below to get it up and running.

*(This post assumes you have already installed and used Git Bash previously)*

## What is conda, and why would I want to use it?

Before we dive into the instructions, it's important to understand a little about conda and how it differs from pip. In short, conda is a very powerful package manager that excels at managing dependencies and offers an easy way to create and use virtual environments for your projects. It is used primarily in the Data Science world, but can be used for much, much more.

For a detailed breakdown, read the excellent post by [Jake VanderPlas](#) (Software Engineer, Google), [Conda: Myths and Misconceptions](#)

## Instructions

### 1. Install Anaconda

Although Codecademy recommends installing Miniconda, I **highly recommend** saving some headaches and installing the full version of Anaconda if you have the space on your PC and you are serious about getting into Data Analysis/Data Science. Miniconda is a barebones version of the Anaconda distribution, and is a little less beginner-friendly, coming with only Python, conda, pip, and a couple of dependencies necessary to make them work.

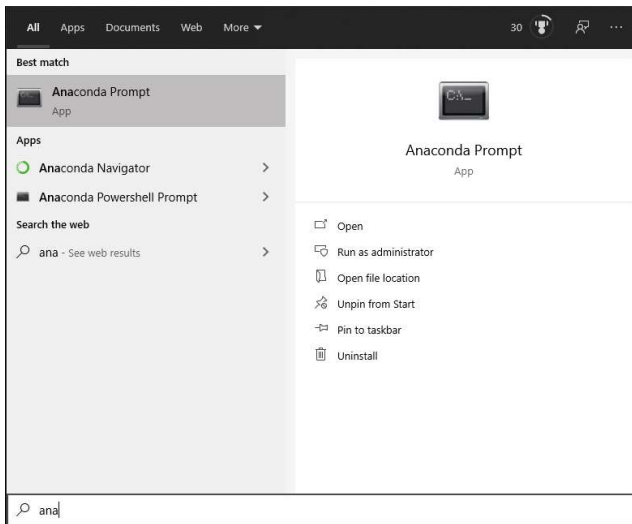
If you are low on hard drive space, Miniconda is fine — just don't be surprised by the number of packages you'll have to install that would be ready out of the box with Anaconda (Pandas, Numpy, Matplotlib, Seaborn, Jupyter, Scikit-learn, etc.).

You can download the installer for Anaconda [here](#).

### 2. Run Anaconda Prompt (skip this section if you are familiar with conda)

Once Anaconda is downloaded and installed, you will be ready to use conda. Before we set it up in Git Bash, it is useful to see what it looks like when you are using conda.

Anaconda and Miniconda come with a program called Anaconda Prompt on Windows, which is essentially just CMD terminal that is pre-set-up for conda. If you type **Anaconda** in your Windows search bar, you will see Anaconda Prompt come up. Click on it to use conda for the first time:



Once you open it up, you will see something like this:



Notice that at the beginning of the prompt, you see `(base)`. This tells you that you are currently inside of the conda virtual environment called `base`. This base environment is where all of the pre-loaded packages are contained. Later on, as you become more familiar with `conda`, you should begin creating new environments for each project you work on. When you do, you will see the name of the currently active `conda` environment inside of the parentheses. For example, if you make and activate an environment called “test”, you will see `(test)` at the beginning of the prompt.

If you are interested in seeing which packages are installed in your base environment, simply type the command `conda list`.

### 3. Make conda run in Git Bash

#### A) Open Git Bash

Now that you have an idea of how `conda` should look, go ahead and open Git Bash. As you can see, there is nothing on the prompt to indicate whether you are inside an active `conda` environment or not. Unfortunately, it's not quite that simple. If you type `conda list` in Git Bash, you will receive an error that looks something like this:

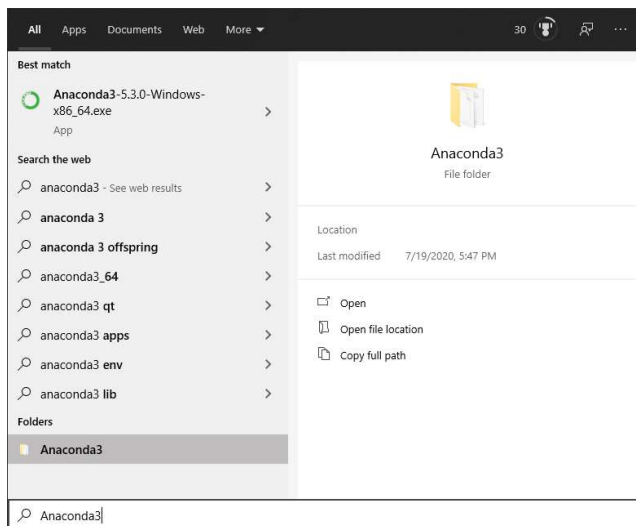
```
bash: conda: command not found
```

This is because `conda` has only been set up to run from Anaconda Prompt by default.

#### B) Add the conda shell script to your .bashrc

In order to make the `conda` command available in Git Bash, you need to add `conda`'s shell script to your `.bashrc` file. This is the same file that you store your bash aliases in (such as the `sqlite3` alias you probably created when you followed [these instructions](#) ).

The shell script we need is located inside of the folder your Anaconda distribution added to your computer. If you remember where that is, you can navigate there using the File Explorer. If you don't remember, the easiest way to find it is to use the search bar and search for "Anaconda3" and scroll down to the folders:

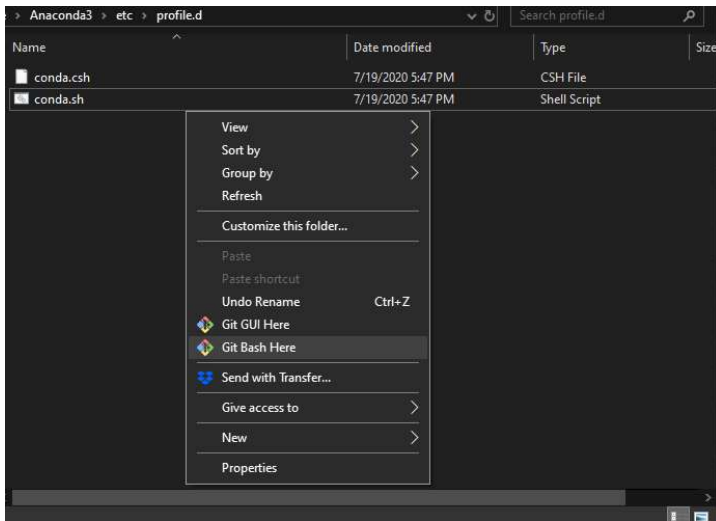


Open this folder, then navigate to `etc -> profile.d`.

Inside of the `profile.d` folder, you should see a file called `conda.sh`. This is the file you want to add to your `.bashrc` file.

Now, there are a few ways you can do this: a) you can open your `.bashrc` and type it in there; b) you can type the path to `conda.sh` in Git Bash and add it to your `.bashrc` from there; or c) you can open Git Bash in the `profile.d` folder and utilize the bash command `PWD` in order to lower the risk of spelling errors in the path. The last option is my preferred choice, so that is what I will cover here.

Go ahead and close Git Bash, and then right click inside the file explorer and choose "Git Bash Here":



This will open a new instance of Git Bash to the correct directory.

Now, so long as the path to your file has no spaces, you can add `conda.sh` to your `.bashrc` with this simple command:

```
echo ". ${PWD}/conda.sh" >> ~/.bashrc
```

If the path *does* contain spaces (such as in your username), you will need to add single quotes in the command like so:

```
echo ". '${PWD}'/conda.sh" >> ~/.bashrc
```

After running this command, you will need to close Git Bash and reopen it for it to take effect.

### C) Reopen Git Bash and activate conda!

When you reopen Git Bash, you still won't see `(base)` in your prompt — that's because you haven't activated conda yet! To activate it, simply type the command `conda activate` and hit Enter. You should see `(base)` pop up above your prompt:

A screenshot of a Git Bash terminal window. The title bar at the top reads "MINGW64:/c/Users/". The terminal has a black background with white text. The prompt is "\$". The user has entered the command "conda activate", and the terminal has responded with "(base)". The prompt is now "\$". The terminal window has standard Windows window controls (minimize, maximize, close) in the top right corner.

```
MINGW64:/c/Users/
$ conda activate
(base)
$
```

And that's it! You've successfully enabled `conda` in your Git Bash terminal. Now Git Bash can do virtually everything Anaconda Prompt can do, just make sure to use the `conda activate` command whenever you want to use `conda` in a new Git Bash window (you can also use `conda deactivate` if you want to exit your `conda` environment and use a different installation of Python).