# Installing PHP on Windows 10 Bash and Starting a Local Server

Bash on Ubuntu/Windows is a way to use a command line to run all kinds of programs (including git!). But we'll want a Bash terminal to run PHP and local servers (unless you use the MAMP alternative). If you don't have Ubuntu, you can find a very helpful guide on getting [it installed here](#).

If you have Windows 10 with Ubuntu, you are already almost ready to go with a local server and running a PHP service!

**Starting a Bash Terminal**
To open an Ubuntu Bash terminal, go to the Search panel in Windows and type "Ubuntu". You should see the following result appear:
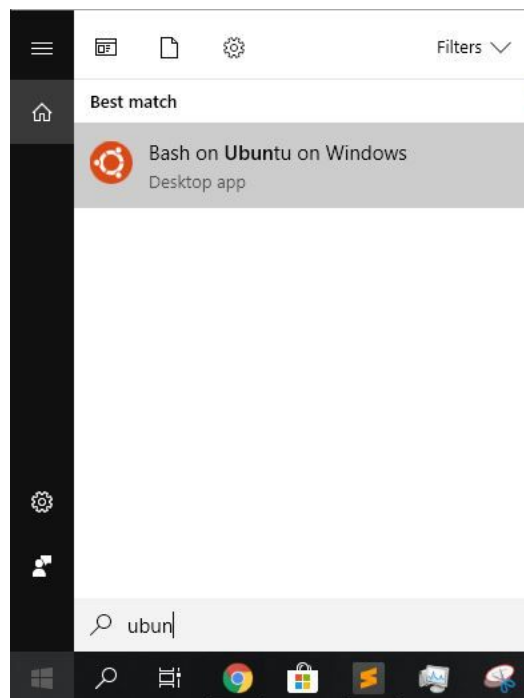


TIp: Once you open Bash, you may find it helpful to pin at the bottom of your desktop for easy navigation later:
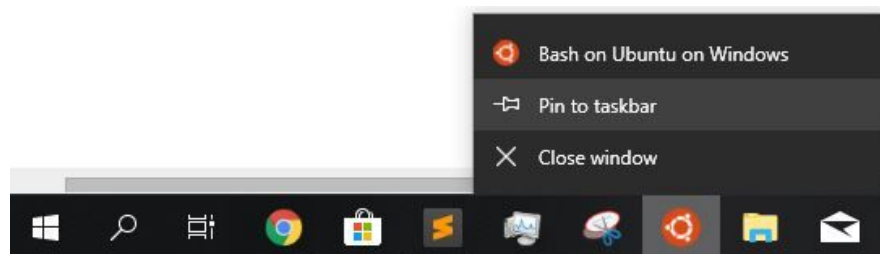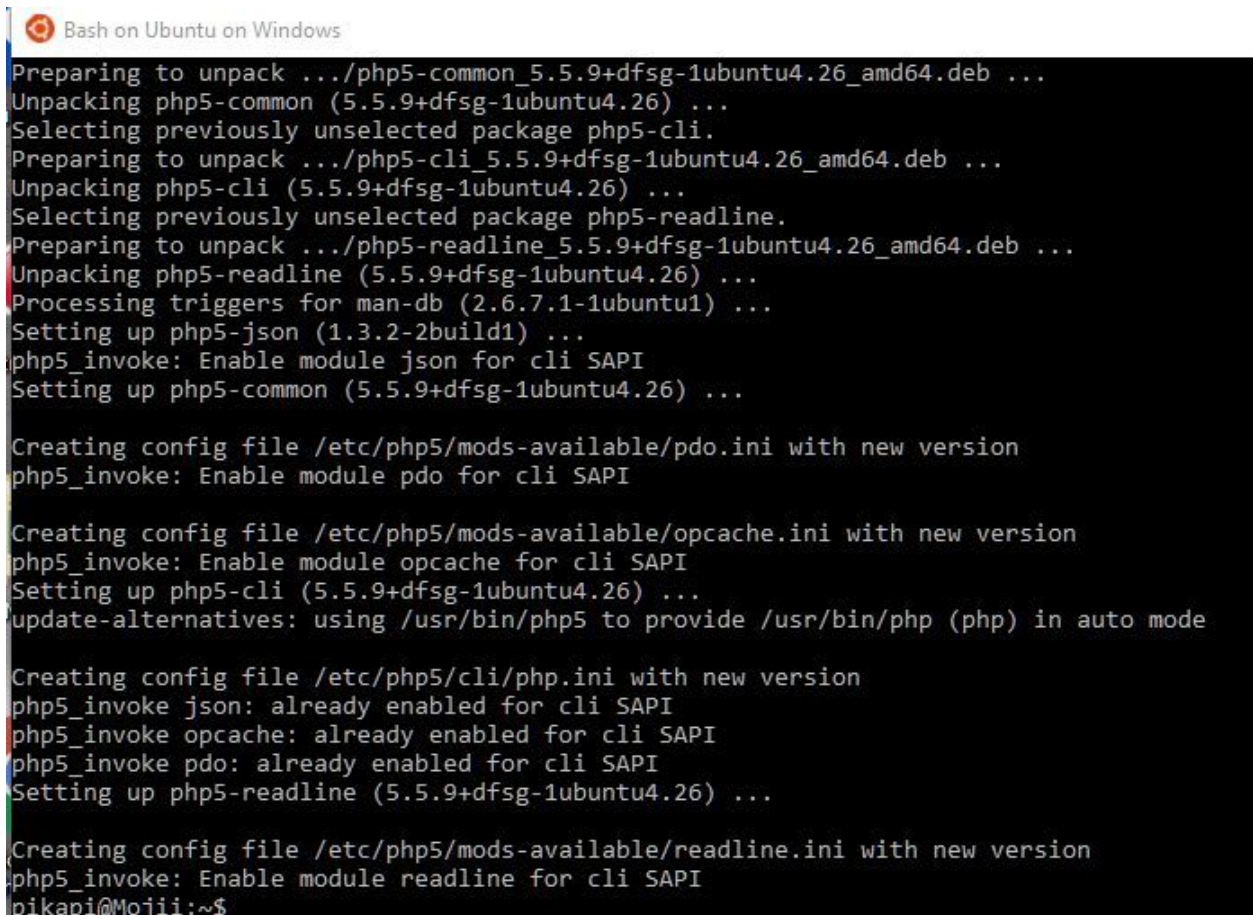
## Installing PHP from Bash

Next, we need to install PHP so we can run a local server. To install, first run

```
sudo apt-get update
sudo apt-get install php5-cli
      (If the previous command didn't work try:
            sudo apt-get install php libapache2-mod-php)
```

Note: These commands will prompt you for your user password - you should provide this when prompted, but note that it won't display the characters you type.



## Moving your command line location to your CSE 154 code

When you open a Bash terminal, you will need to change the current directory to point to one in your WIndows User directories. To do so, type the following:

```
cd /mnt/c/Users/yourusername/Desktop/yourcse154folder
```

For example:



Alright, now that we've installed PHP and are in our desired working directory, we can start running our first PHP program as well as a local server (useful for fetching files or using web services we will write in PHP without needing to publish them to a public server).

In this demo, I've created a simple index.html file, index.css file, as well as a hello.php file in my cse154 directory. You can see the three different ways I can view the contents of cse154 which is in my Desktop:
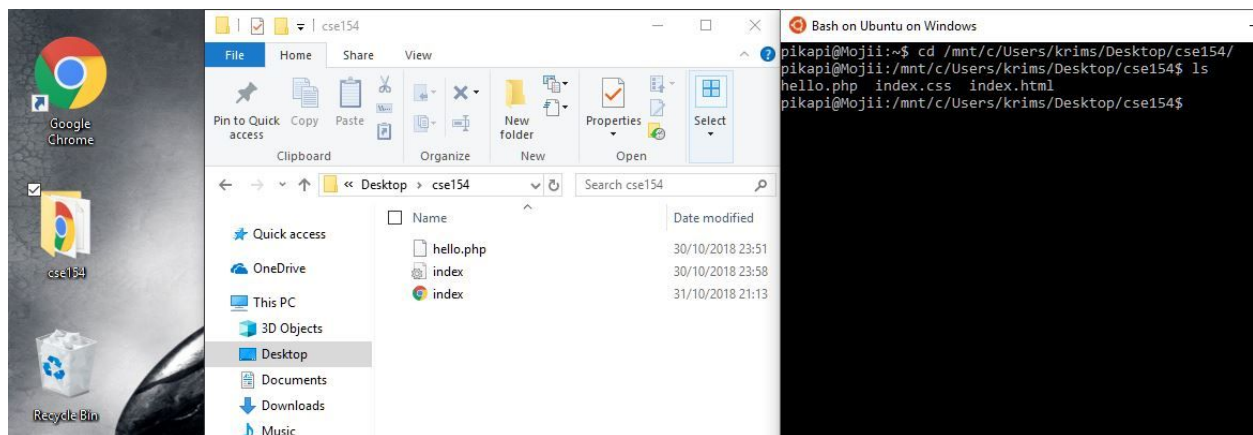


**Starting a local server:**
It's important your Bash terminal is in the correct folder you are editing your CSE 154 files in! We can start a local server with PHP using the command

```
php -S localhost:8000
```

which starts a server listening to the port 8000. This server will gives us the ability to run PHP web services and fetch from them in our JS files using relative links. Once you have a working website with HTML, CSS, JS, and PHP, you can publish to a public server like your student website. But it's tedious to use FIleZilla to update your files each time you make a change,

which is why a local server is very useful here. Without a server running, you won't be able to run a PHP service or fetch local files from your JS.

**Opening an index.html file from a local server location**
To open our index.html file, we open our browser and visit localhost:8000/index.html. localhost:8000 will point to the current location where you started the server in your Bash terminal. Since we started it in our cse154 directory with index.html and hello.php, adding index.html to the url will give us our index.html page, just like you've already seen when viewing local HTML files in your browser.
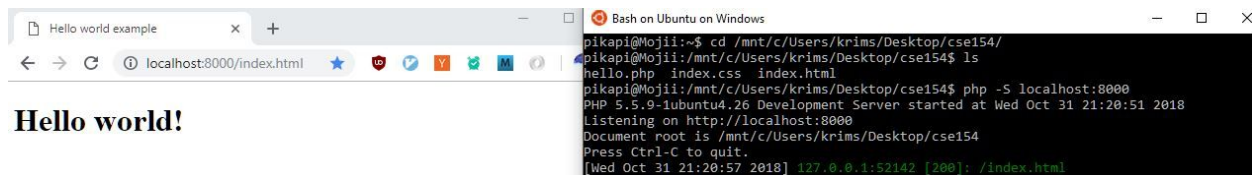


We can also run PHP services and view their output in our browser (we'll learn more Friday). Let's preview what that looks like!
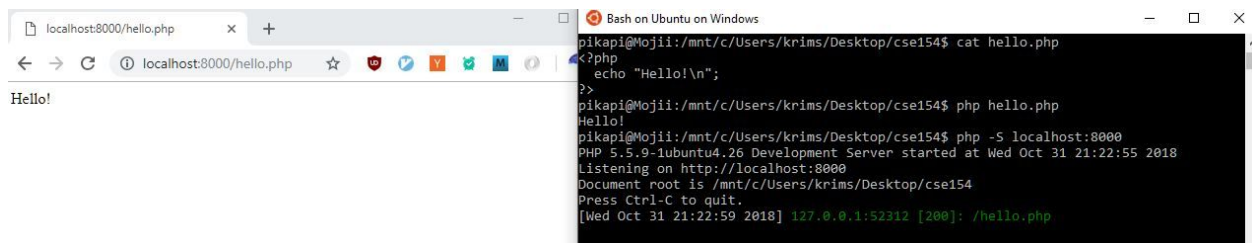


Let's take a closer look at the commands ran in the right terminal window.

First, I've ran the Bash command cat which outputs the contents of a file conveniently to my command line. You will see the entire contents of a simple PHP program hello.php printed out.

Next, I ran that PHP program directly in the command line with

```
php hello.php
```

This basic PHP program prints out (echoes) "hello!". When writing PHP programs, it's useful to be able to run them from the command line occasionally (you can't run a PHP program from Atom or Sublime).

What you'll usually use though is the browser to run your PHP programs (usually as web services). As you see in the screenshot, after starting the local server and visiting the url localhost:8000/hello.php, we see the output of our PHP program in the browser!

**Conclusion**

And that's it! You're ready to practice writing PHP programs now and run them from the command line or your browser with a local server running. Each time you close the terminal, your server will also finish, so make sure to restart it when you need. You will also need to move to your cse154 directory each time you open a terminal (come talk to Melissa if you want to learn how to automatically open a terminal in your desired directory) so make sure you have the

```
cd /mnt/c/Users/yourusername/Desktop/yourcse154folder
```

on hand somewhere.