

DOS Lab

The purpose of this lab is to teach you some basic commands in the DOS operating system for those who are using a Windows machine. If you are using a Mac, read the [Unix Lab](#) instead.

Introduction

You are probably already familiar with interacting with files and folders on your computer through the "File Explorer" program. File Explorer is a GUI ([Graphical User Interface](#)) which allows you to navigate through your file system and create new folders and documents, and organize them in ways that make sense to you. Sometimes in programming however, you will need to use your file system in a different way. For this, on Windows, you will use a program called "Command Prompt". The Command Prompt (formerly called a DOS Command Prompt) is known as a CLI ([Command Line Interface](#)). That means in addition to accessing your file system, you can also use the Command Prompt to run Python programs directly, among many other programming tasks.

You will find these basic commands useful as you try to run a Python program through a command line interface or to browse through the file system on your computer. Sometimes, you may have to deal with some of the internals of the Windows operating system in which case you may have to use some of the DOS commands.

Also note that the file system on your computer is structured in a *hierarchical* way. That means that files and folders are nested inside of other folders. There is some specific terminology we will use when dealing with hierarchical structures like this: [directory](#) (also called *folder*), parent directory, sub-directory, and file. When one directory contains another directory, the containing one is called the "parent directory", and the contained one is called the "sub-directory."

Instructions

1. Launching Command Prompt

Open the Command Prompt window on Windows 10 by going to:

1. Start -> type "cmd" in the text input window.

2. It should bring up "Command Prompt" as the best match, press enter to open it.

You should see a mostly-blank window open that shows your user directory followed by the > symbol. This is called the command **prompt**, and you will often read or hear the phrase "enter this text at the prompt." That means type that text after the >.

2. Entering Commands

An important thing to note is that in the Command Prompt you can use "tab-complete" to automatically finish the name of a file or directory so you don't have to type the whole thing. This will save you a tremendous amount of time and avoid typing mistakes. To tab-complete, begin typing the name of the file or directory and then hit the Tab key. If the Command Prompt is able to map what you've typed to a unique location, it will complete the name for you. If the Command Prompt finds multiple files or folders with a similar name, it will choose the first one and you can continue hitting the Tab key to cycle through all the options.

If at any point you want more information about a particular command you have entered, you can type

```
help
```

followed by the name of the command you want information on. This command will display the instruction manual for that command, including a description of what it does and all the extra options you can add to change what the command does.

Enter the following at the command prompt:

```
help dir
```

In this case, it will tell you about the `dir` command, which prints the files and folders from where you currently are in the file system.

3. Navigating

In the following exercises, you will try moving around various directories and creating and modifying files on your computer. As you follow the steps, try to find the same directories and files using the File Explorer program (*not* Internet Explorer) at the same time so you can see the effects of the commands you are entering.

The Lab

1. Create a new directory

```
mkdir NewDirectory
```

`mkdir` stands for *make directory*, and it creates a new directory (or folder) named whatever you type after `mkdir`.

2. List all of the files and directories in your current location

```
dir
```

`dir` stands for *directory*. `dir` prints out additional information such as the file size and time each file was last edited.

3. Move to the NewDirectory you just created

```
cd NewDirectory
```

The `cd` command stands for *change directory*, and it moves you in the Command Prompt to the directory you entered after `cd`.

4. Open the current directory in File Explorer

```
start .
```

By entering the `.` after the `start` command, you are asking to open the File Explorer in the current directory.

5. Create a new file with File Explorer

All the changes you make in File Explorer and the Command Prompt are reflected on your actual file system. You can see this by using File Explorer to create a new file (right-click -> new -> Text Document), which you can then name "HelloDOS.txt". Next open the file and enter in your name and major in the file and save it. Now we will go see this new file by going back to the Command Prompt.

6. View the contents of HelloDOS.txt

```
more < HelloDOS.txt
```

The `more` command shows you the contents of the file page by page.

7. Rename the new file you just created

```
rename HelloDOS.txt GoodbyeDOS.txt
```

This will rename the HelloDos.txt file to the new name GoodbyeDOS.txt. If you instead wanted to make a copy, you could instead type:

```
copy HelloDOS.txt anotherGoodbyeDOS.txt.
```

8. Delete the file you just renamed

```
del /P GoodbyeDOS.txt
```

Here you use the /P option with del so that you get a confirmation prompt before deleting the file, because once a file has been deleted, it is gone forever! If you are sure you want to delete something, you can omit the /P option and just enter del <filename>, but be careful with that command!

9. Navigate back out of NewDirectory

```
cd ..
```

A single . indicates the directory you are in. If you were to type cd . nothing would happen! But when you use two dots, that means go to the *parent* directory of wherever you are. It moves you one level up the hierarchy.

10. View a list of all the commands you just typed

```
doskey /History
```

This will show you all of the commands you have entered into the Command Prompt so far.

This lab only covered some basics of the DOS Command Prompt, so if you're interested in learning more there is a lot more information available online.