

# The Command Line Interface

In this lesson, we'll discuss the basics of perhaps the most important tool in your belt as a programmer: the command line

## We'll cover the following

- What is the terminal?
  - Listing files and directories
  - Checking what directory you are in
  - Changing directories
  - Reading files
  - Creating files
  - Creating a new directory
  - Moving files
  - Removing files

## What is the terminal?

The **command-line interface** is a tool into which you can type text commands to make the computer perform specific tasks in contrast to using a graphical user interface through pointing and clicking on menus and buttons with a mouse.

Since the command line interface allows you to control the computer directly by typing commands, many tasks can be performed more efficiently. For example, if you want to make one change in the names of around 2000 files, you can use a for loop on the command line and change the names of all of them in a few seconds. Therefore basic knowledge of the command line is an absolute essential.

The user interface that accepts typed commands and displays data on the screen is called a **shell**. There is a wide variety of shells that you can choose from, but the most common is the [Bash shell](#), which is the default on Linux

and Mac systems in the Terminal application. Windows systems only include the anemic Command Prompt application by default, which is nowhere near as powerful as Bash. If you're going to follow along on your windows machine, however, consider installing [cygwin](#).

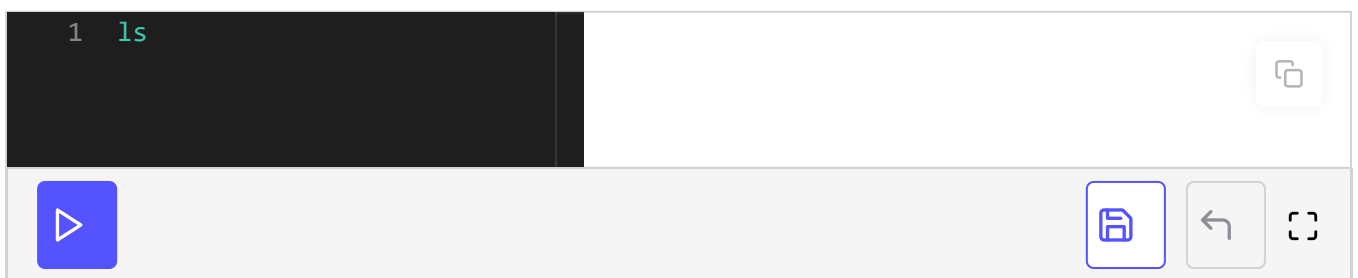
Open up a shell; you will see a blank (usually black) screen with a cursor to the right of something like:

```
accountname@machinname:~$
```

Let's finally look at a few commands now!

## Listing files and directories #

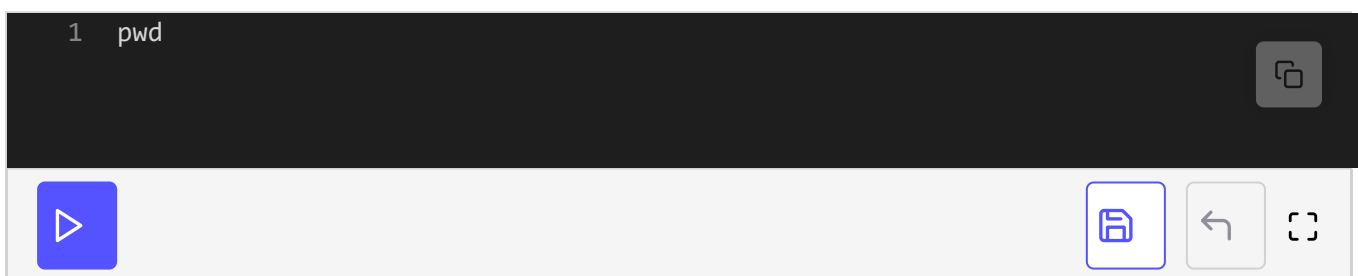
Type `ls` into a shell and hit enter (or try it below!) and watch the magic happen.



Voila! All the files and directories in your current directory get printed.

## Checking what directory you are in #

Also, to check what directory you are currently in, type `pwd`, and hit enter. Here is an example:



## Changing directories #

You can change to a directory that exists within your current directory with the command,

```
cd nameOfDirectory
```

you can move back to a parent directory with the command,

```
cd ..
```

```
ls
cd sampleDirectory
ls
cd ..
ls
```



## Reading files #

You can print files to the terminal with the command,

```
cat nameOfFile
```

main.sh

SampleFile.txt

```
cat SampleFile.txt
```



## Creating files #

You can create empty files with the command,

```
touch nameOfFile
```

```
touch nameOfFile.txt
ls
```



## Creating a new directory #

You can create a new directory with the command

You can create a new directory with the command,

```
mkdir newFolder
```

```
mkdir newFolder  
ls
```



## Moving files #

You can move files from one directory to another like so,

```
mv path/to/file/filename new/path  
mv path/to/file/new/path/filename .  
mv filename ..
```

Note that `.` indicates the current directory, so the second command above moves the file from `path/to/file/new/path/filename` to the current directory. As mentioned previously, `..` indicates the parent directory, so the last command moves the file `filename` to the parent directory.

main.sh

fileName.txt

```
mkdir newfolder  
mv fileName.txt newfolder  
ls  
mv newfolder/fileName.txt .  
ls
```



## Removing files #

You can remove files with the command

```
rm fileName
```

```
touch fileName # Creating a file  
ls  
rm fileName # Removing it  
ls
```





Quick quiz on the command line!

1

What does the `ls` command do?



A) It lists all the directories



B) It lists all the files, directories, and files within child directories



C) It lists all the files and directories within a folder



D) It lists all the files, hidden files and directories within a folder

COMPLETED 0%

1 of 2



Now that we know some command line basics, let's move on to learning version control with Git in the next lesson!