Basic Command Syntax

This module deals exclusively with the CLI or Command Line Interface, rather than a GUI or Graphical User Interface you may be familiar with. The CLI terminal is a powerful tool that is often the primary method used to administer small low-power devices, extremely capable cloud computing servers, and everything in between. A basic understanding of the terminal is essential to diagnosing and fixing most Linux based systems. Since Linux has now become so ubiquitous, even those who plan on working primarily with systems not utilizing the Linux kernel can benefit from having a basic understanding of the terminal.

What is a command? A command is a software program that when executed on the CLI (command line interface), performs an action on the computer. When you type in a command, a process is run by the operating system that can read input, manipulate data and produce output. A command runs a process on the operating system, which then causes the computer to perform a job.

To execute a command, the first step is to type the name of the command. Click in the terminal on the right. Type `ls` and hit **Enter**. The result should resemble the example below:

```
sysadmin@localhost:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates
```

The name of the command is often based on what it does or what the developer who created the command thinks will best describe the command's function. For example, the `ls` command displays a *listing* of information about files. Associating the name of the command with something mnemonic for what it does may help you to remember commands more easily.

Consider This

Every part of the command is normally case-sensitive, so `LS` is incorrect and will fail, but `ls` is correct and will execute.

```
sysadmin@localhost:~$ LS
-bash: /usr/games/LS: Permission denied
```

Most commands follow a simple pattern of syntax:

```
command [options…] [arguments…]
```

In other words, you type a command, followed by any *options* and/or *arguments* before pressing the **Enter** key. Typically *options* alter the behavior of the command and arguments are items or values for the command to act upon. Although there are some commands in Linux that aren't entirely consistent with this syntax, most commands use this syntax or something similar.

In the example above, the `ls` command was executed without any options or arguments. When this is the case, it's default behavior is to return a list of files contained within the current directory.

```
sysadmin@localhost:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates
```


Toggle onscreen keyboard