Welcome to Shell

When you log into a server over the network the shell program is started and acts as your default interface to the system. The shell is nothing more than a program that accepts your commands and executes those commands. Said another way, the shell is a command line interpreter.

Let's look at the shell prompt you'll be working with. The prompt just sits and stares at you waiting for you do something interesting like give it a command to execute. Here is Bob's shell prompt.

```
bob@linuxsvr $
```

Bob's prompt is in a common format of `username@servername $`. In this example, the prompt is displaying the username, the server name, and if that user is using the system as a normal user (`$`) or a superuser (`#`).

The superuser on a Linux system is also called root. Anything that can be done on a server can be done by root. However, normal users can only do a subset of the things root can do. Root access is typically restricted to system administrators, but if you happen to support an application on a Linux server you may need root privileges to install, start, or stop it. There are ways to grant specific users root privileges for specific cases. This is often accomplished with the `sudo` -- SuperUser Do -- program. That will be covered later. For now, just know that most of your day to day activities will be performed using a normal user account.

Your prompt might not look like Bob's. Common items that appear in prompts include the username, server name, present working directory, and the current time. Here are a few more prompt examples.

```
[bob@linuxsvr /tmp]$
linuxsvr:/home/bob>
bob@linuxsvr:~>
[16:45:51 linuxsvr ~]$
$
%
>
```

In two of the prompt examples you will notice a tilde (`~`). The tilde is a shorthand way of representing your home directory. In this example the tilde (`~`) is equivalent to `/home/bob`, which is Bob's home directory. This is called tilde expansion. A username can be specified after the tilde and it will be expanded to the given user's home directory. For example, `~mail` would expand to the home directory of the mail user which is `/var/spool/mail`. Another example is `~pat` expanding to `/home/pat`.

Prompts do not have to be contained on a single line. They can span multiple lines as in the following examples.

```
linuxsvr:[/home/bob]
$

(bob@linuxsvr)-(06:22pm-:-11/18)-]-
(~)

[Mon 13/11/18 18:22 EST][pts/0][x86_64]
<bob@linuxsvr:~>
zsh 14 %

*linuxsvr* Mon Nov 18 06:22pm
~/
```

For the remainder of this course the prompt will be shorted to the dollar sign (`$`) unless displaying the full prompt provides additional clarity. Also, the default prompt may vary from system to system, but you can customize it to your liking. That, along with other shell related topics, is covered in a later lesson.