Linux Terminal Tips - Tab completion, command history

Learning Objectives

After completing this reading, you will be able to:

- Use tab completion to autocomplete commands
- Use command history to quickly navigate previous commands

Tab Completion

Many shells support a feature called **tab completion**.

Tab completion allows you to autocomplete a command you're typing on the command line.

Suppose you're in your home directory ~, which contains the directories:

- `Pictures`
- `Videos`
- `Documents`
- `Downloads`

And suppose your `Documents` folder *only* contains the folder:

- `python-examples`

Tab completion:

In this exercise, you will use tab completion to quickly enter a command to navigate to the `Pictures` directory.

If you type:

```
~ $ cd P
```

and press `Tab`, the shell will autocomplete the command to:

```
~ $ cd Pictures/
```

because the `Pictures` directory is the only directory within your current folder that starts with a "P".

Tab completion for long paths:

You can also use tab completion to autocomplete longer paths.

If you type:

```
~ $ cd Do
```

and press `Tab`, nothing will happen because your current directory contains more than one directory that starts with "Do". The shell won't know whether to autocomplete with `Documents` or `Downloads`.

On the other hand, if you type:

```
~ $ cd Doc
```

and press `Tab`, the shell will autocomplete to:

```
~ $ cd Documents/
```

because in this case, there's only one directory that starts with "Doc," `Documents`.

If you press `Tab` again, the shell will further autocomplete to:

```
~ $ cd Documents/python-examples/
```

because the folder `python-examples` is the only existing file within the `~/Documents` directory. With just a few clicks, you can quickly autocomplete longer paths in your command line.

Command History

Command history allows you to navigate previous commands you've entered using the `Up Arrow` and `Down Arrow` keys.

Let's say you've created a Python script called `hello_world.py` in your `python-examples` directory that simply prints `Hello, World!` when you run it.
Consider you have just entered the following sequence of commands, and the shell is awaiting your next input:

```
~ $ cd ~/Documents/python-examples
~/Documents/python-examples $ python3 myprogram.py
Hello, World!
~/Documents/python-examples $ cd /
/ $
```

In the following exercises, you will learn how to rerun a previous command withut having to retype it by pressing the `Up Arrow` key.

Running the last command:

If you press the `Up Arrow` key once, the shell will automatically insert the last command you entered:

```
~ $ cd ~/Documents/python-examples
~/Documents/python-examples $ python3 myprogram.py
Hello, World!
~/Documents/python-examples $ cd /
/ $ cd /
```

Notice that the last command you entered was `cd /`, which has been automatically inserted onto the command line for you.

Running previous command from session:

If you press the `Up Arrow` key two *additional* times (so, three times total), the shell will automatically insert the command you ran *three* commands ago:

```
~ $ cd ~/Documents/python-examples
~/Documents/python-examples $ python3 myprogram.py
Hello, World!
~/Documents/python-examples $ cd /
/ $ cd ~/Documents/python-examples
```

In this case, the command you ran three commands ago was `cd ~/Documents/python-examples`. Note that the line of printed output, `Hello, World!`, does not count as a command. Command history will only take you through commands you entered, not every line that is visible in the terminal.

> **Tip**: If you click the `Up Arrow` key too many times, you can use the `Down Arrow` key to cycle through your command history in the opposite direction.

Press `Enter` to return to your `~/Documents/python-examples` directory. You should see something like the following:

```
~ $ cd ~/Documents/python-examples
~/Documents/python-examples $ python3 myprogram.py
Hello, World!
~/Documents/python-examples $ cd /
/ $ cd ~/Documents/python-examples
~/Documents/python-examples $
```

Summary

Congratulations! You now know how to use a few convenient shortcuts to speed up your command line interactions!

In this reading, you learned how to:

- Use tab completion to autocomplete commands
- Use the command history to quickly navigate to previous commands