Linux Terminal Tips - Tab completion, command history

Tab Completion

Many shells support a tool called "Tab Completion"

Tab completion allows a shell to auto-complete a command you're typing.

Support you're in your home directory `~`, which contains:

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

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

- `python-examples`

Ex 1 - Tab Completion:

If you type:

```
~ $ cd P
```

and press TAB, the shell will autocomplete this to:

```
~ $ cd Pictures/
```

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

Ex 2 - Tab Completion for Long Path:

If you type:

```
~ $ cd Do
```

and press TAB, nothing will happen.

Because your current directory contains more than one directory that starts with "`Do`".

On the other hand, if you type:

```
~ $ cd Doc
```

and press TAB, the shell will autocomplete this to:

```
~ $ cd Documents/
```

If you press TAB again, the shell will autocomplete this to:

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

Because the folder `python-examples` is the only existing file within the `~/Documents` directory.

Command history

Command history allows you to navigate previous commands you enetered using the arrow keys

Let's say you've entered the following commands:

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

You can run a previous command, by pressing the up arrow key

Ex 1 - Running the last command:

If you press UP once:

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

The shell will automatically put the last command you entered

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

Ex 2 - Running previous command from session

If you had pressed UP *three times* instead of once:

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

It would automatically place the command you ran three commands ago (`cd ~/Documents/python-examples`)

After pressing enter, youwould find yourself back in your `cd ~/Documents/python-examples` directory:

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