BASH SCRIPTING 101

Commands on Command

We all have at least some expereicne using the command line. If you have ever used 'mkdir', 'cd', 'ls', or 'pwd' you have command line experience.

Bash Scripting allows us to take these, and more complex commands, and turn them into an executable file called a 'shell script' or '.sh' file.

Example

The Whole #!

The #! symbol is called a 'shebang' it tells the interpreter that it's dealing with a shell script.

For bash scripting we use:

#!/usr/bin/bash

Hello World

- echo behaves like a print statement
- "-e" allows for escape characters such as newline "\n"
-"$1" refers to the first parameter passed in.

```
#!/usr/bin/bash

echo -e "Hello World!"
echo -e "Hello $1"
```

Variables

-Like most languages bash has variables
-Variables are declared in bash like they are in Python however in bash there should be **no spaces** in the variable declaration

```
1 #!/usr/bin/bash
2
3 FRIENDS="Daleks"
4 FOE="The Doctor"
5
6 echo -e "Hello ${FRIENDS}"
7 echo -e "Exterminate ${FOE}"
```

If Statements

-"; then" is required for if statements

-if statements must be closed with 'fi'

```
1 #!/usr/bin/bash
2 #Dave gets a special greeting
3 if [[ $1 == Dave ]]; then
4     echo -e "Hi $1, thanks for inviting me to present!"
5 else echo -e "Hello $1, it's lovely to meet you."
6 fi
```

For Loops

-Bash standard uses brackets instead of parentheses
-Brackets require padding with a space
-"; do" is required for loop structures
-loop structures must be ended with "done"

```
1 #!/usr/bin/bash
2
3 friends='Harry Ron Hermione'
4
5 for friend in $friends; do
6     echo -e "Hello $friend"
7 done
```

Functions

-functions must be declared before they can be called
- line 7 is calling the declared function say_hello

```
1 #!/usr/bin/bash
2
3 function say_hello {
4     echo -e "Hello $1"
5 }
6
7 say_hello "World"
```

Functions

-Functions can also be delared with parenthesis but the parenthesis will always be empty.

```
1 #!/usr/bin/bash
2
3 say_hello () {
4     echo -e "Hello $1"
5 }
6
7 say_hello "World"
```

What is Vim?

-Vim is a text editor, an advanced text editor, but a text editor
-Most bash terminals will have vim built in, and you can access it with the 'vim' command
-Vim is powerful and the best way to learn it is to start using it

How To Exit Vim

:qa!

Hard quit, what you would use if you accidently launch vim when commiting

:wq

Write quit. Save your work and exit

:q

Quit, you've already saved (:w) your work, or haven't made changes and you want to quit

Resources

Linux:

http://www.linfo.org/pipes.html

https://shapeshed.com/unix-exit-codes/

https://misc.flogisoft.com/bash/tip_colors_and_formatting

Bash:

https://ryanstutorials.net/bash-scripting-tutorial/

Vim:

https://www.vim.org/about.php

https://vim.rtorr.com/

Examples:

https://github.com/cmpeters08/bash-presentation