Working with Programs: Takeaways ↗

Syntax

- Assign a variable

```
OS=Linux
```

OR

```
OS="Linux"
```

- Print a variable

```
echo $OS
```

- Create an environment variable

```
export LINUX="Linux is best"
```

- Run python inside of Bash

```
python
```

- Access environment variables

```
import os
print(os.environ["LINUX"])
```

- See what folders are in PATH

```
echo $PATH
```

- List files in directory in long mode

```
ls -l
```

- Specify a longer flag with two dashes

```
ls --ignore
```

Concepts

- A shell is a way to access and control a computer.
- Bash is the most popular of the UNIX shells, and the default on most Linux and OS X computers.
- Command line shells allow us the option of running commands and viewing the results, as opposed to clicking an icon in a graphical shell.
- Bash is essentially a language, which allows us to run other programs.
- A command language is a special kind of programming language through which we can control applications and the system as a whole.
- Quotations around strings in Bash are optional, unless they contain a space.
- We can set variables on the command line by assigning values to them.
- In the command line environment, variables consist entirely of uppercase characters, numbers, and underscores.
- We can assign any data type to a variable.

- Accessing the value of a variable is not much different than Python -- we just need to prepend it with a dollar sign (`$`).
- Environment variables are variables you can access outside the shell.
- We can run many programs from Bash, including Python.
- `os.environ` is a dictionary containing all of the values for the environment variables.
- The `PATH` environment variable is configured to point to several folders.
- Programs are similar to functions, and can have any number of arguments.
- Programs can also have optional flags, which modify program behavior.
- We can chain multiple flags that have single, short, character names.

Resources

- [UNIX Shells](#)
- [Environment Variables](#)