# Python - Custom Functions and OS Module

**Eli Etherton - August 2, 2024**

## Functions

- Like Apps within your script
- Reuse code - Write Once
- Must be defined BEFORE they are called
- Good coders are lazy coders
- Example
    - hello.py
    - function-layered.py

## Function Inputs

- Sending Variable Values to a Function
- Position Matters not the Name
- Local vs Global Variables
- Examples
    - function-input.py
    - function-input-multiple.py

## Function Returns

- Functions can return variable values
- Multiple Values Can Be Accessed Based on Index
- Return can be a Dictionary or List
- Examples
    - return.py
    - return-multiple.py
    - return-dict.py

## OS Module

- Allows you to send commands and receive responses from the Operating System
- This Makes Command Line Tools available in your Python Script
- Different OS’s use different base commands
- Beware of PING!!!
- What About the Subprocess Module???

## OS Basic Functions

- OS Module will use the appropriate command for the Operating System
- os.name
- os.getcwd()
- os.listdir()
- os.mkdir(‘name’) - os.remove(‘file’) - os.rmdir(‘directory’)
- os.join(directory, file)
  - Formats file path for OS
- Example = os-example.py

## OS - System Function

- Sends Commands to OS with No Return
- OS Commands are OS Dependent
- subprocess.run() - ???
- Example = os-system.py

## OS - popen() Function

- Sends command to OS and receives what would be displayed on the screen.
- GREP is your friend!
- read()
- readlines()
- Example = os-popen.py

## Try / Except

- What is an error really???
- try:
- except:
- except Exception as error:
- else:
- finally:

## Labs

- lab-command.py
- lab-up-down.py
- lab-dashboard.py