# 

**Programming 1**

**AUTUMN ASSIGNMENT #1**

Pankaj Singh Rawat – C00290835

Date of Submission: November, 2022

Table of Contents

**INTRODUCTION3**

**ACTIVITY3**

**ANALYSIS & INTERPRETATION** **4**

**CONCLUSIONS**6

**APPENDICES** 7

**REFERENCES** 9

# INTRODUCTION

The main goal of this assignment is to ping an IP address to see if it is alive or unreachable. This program/assignment teaches us about iteration (loops) and function use. Specifically, command line arguments are arguments that are passed as parameters so that the user can enter or perform a function in a command prompt/terminal.

# ACTIVITY

The Activities that we have performed in the assignment are as follow:

* Write the program "pinger2.py" in Python.
* Develop management tools for ping and license data.
* Build a function named main () that is activated after it is determined that the program is executing its own instance and is not being called by another program.
* The software MUST verify that any IP addresses it receives, whether as terminal arguments or from a prompt, are valid IP addresses.
* The functionality of the program should verify whether the IP addresses given are reachable or not.
* This program should ask the user for an IP address, check connectivity, and, as shown below, send a message back to the terminal indicating whether the address is reachable or not.
* The **-h** or -**-help** switch to return a helpful message.

~$ **./pinger2.py -h**

usage: pinger2.py [-h] [-l] [-v] [-i [IP [IP ...]]]

pinger2.py IP address pinger program

optional arguments:

-h, --help show this help message and exit

-l, --licence pinger2.py licence information

-v, --version pinger2.py version information

-i [IP [IP ...]], --ip [IP [IP ...]]

IP address list

~$ **./pinger2.py --help**

usage: pinger2.py [-h] [-l] [-v] [-i [IP [IP ...]]]

pinger2.py IP address pinger program

optional arguments:

-h, --help show this help message and exit

-l, --licence pinger2.py licence information

-v, --version pinger2.py version information

* -i [IP [IP ...]], --ip [IP [IP ...]]
* Upon a -l or —licence switch, give the user the value of the \_\_author\_\_, \_\_copyright\_\_, and \_\_licence\_\_ dunders.

~$ **./pinger2.py -l**

Author : Diarmuid O'Briain

Copyright : Copyright 2021, Institute of Technology Carlow

Licence : European Union Public Licence v1.2

~$ **./pinger2.py --licence**

Author : Diarmuid O'Briain

Copyright : Copyright 2021, Institute of Technology Carlow

Licence : European Union Public Licence v1.2

~$ **./pinger2.py --license**

Author : Diarmuid O'Briain

Copyright : Copyright 2021, Institute of Technology Carlow

Licence : European Union Public Licence v1.2

* Upon a -v or —version switch, give the user the value of the version dunder.
* Present a prompt to the user after they have entered nothing, and depending on the IP address they have provided, return alive|unreachable|Error.
* For testing, enter a list of IP addresses using the -i or —ip switch. Every ip address is put to the test, and the program returns alive, unreachable, or error.

# ANALYSIS & INTERPRETATION

Following are the interpretation that are made from the assignment:

1. First of all, we will discuss about the libraries that we have imported for our assignment these all libraries are pre-installed in python, no 3rd party library is used:

* **Import argparse:** The argparse module simplifies the creation of user-friendly command-line interfaces. The program specifies which arguments it requires, and argparse determines how to extract them from sys.argv. The argparse module also generates help and usage messages automatically. When users provide the program with invalid arguments, the module will generate errors. (Python.org, 2022)
* **Import ipaddress:** ip address allows you to create, manipulate, and operate on IPv4 and IPv6 addresses and networks.

The functions and classes in this module make it simple to handle various IP address-related tasks, such as determining whether two hosts are on the same subnet, iterating over all hosts in a specific subnet, determining whether a string represents a valid IP address or network definition, and so on.(Python.org, 2022)

* **Import os:** This module offers a portable method of using functionality that is dependent on the operating system. See open (if you simply want to read or write a file), os.path (if you want to modify paths), and fileinput (if you simply want to read every line in every command-line file). The tempfile module should be used to create temporary files and directories, and the shutil module should be used to handle high-level file and directory operations.

1. Now we will discuss about the function that we have used in our assignment:
   * **Ip\_address() function:** In this function we are checking the formatting of IP address if the given IP address is given in proper format. The ip address () function returns an object of type IPv4Address, indicating that it can convert a string into a valid IP address.

Let’s try with an invalid ip address

This time, the string we passed to the ip address() function does not represent a valid IP address, and the function throws a ValueError exception. (Sabato, 2021)

* + **Single Ping:** Same working as link ping but does not have iteration
  + **List Ping:** The arguments are saved in this function (i.e. – pinger2.py, — i, list of all IP address). We are using iteration in this function to find the IP address, and we are starting the loop from index 2 because the first two indexes do not contain an IP address. In the first iteration, we will validate the IP address. The ip address function is then used. If the first iteration validates our IP address, the output = output.read will be printed (). We sent data packets for this purpose, -c is passed as an argument for data packet, and the next value is data packet quantity in argument, followed by waiting time in seconds. If no response is received for the data packet, this indicates ip address is unreachable
  + **Main function:** The first topic we will cover in the main function is Try except, which is used to handle error exceptions. When we use this exception, the code will not crash. This function also handles file handling and calls other functions in the main function.

We analysed the following things based on the assignment activity:

1. Used shebang (#! /usr/bin/env python3) to make the python script executable without explicitly calling python. The shebang is a special character sequence in a script file that specifies which program should be called to run the script. The shebang is always on the first line of the file and is composed of the characters #! followed by the path to the interpreter program.
2. Used import argparse (argparse is what obtains the input from command line), import ipaddress(ipaddress is used to validate if the IP is in correct format) and import os module (os is used for operating system related tasks)
3. Validate the format of IP address using ipaddress() function
4. ping function where IP is provided to get back the status of the IP.
5. validate if the IP is of the correct format
6. Return an IPv4Address or IPv6Address
7. ipaddress.ip\_address(ip) raises a Error if if the address passed isn't either a IPv4Address or IPv6Address
8. ping command with a timeout of 1 seconds
9. timeout is required to know if an IP is unreachable
10. os.popen runs the command through command-line
11. using the read () method on output from os.popen gives us the string that would be printed on terminal
12. licesne function return the license information( author's name, copyright and licence).

# CONCLUSIONS

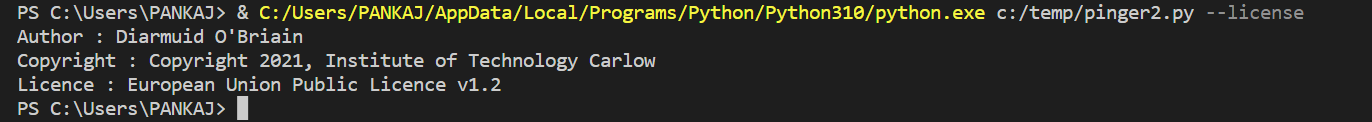
## The assignment taught us how to properly implement loops and iterations, use Python functions, and link them together. In particular, we learned about Parsing command line arguments in Python. Thanks to all of these functions, we can check the validity of IP addresses and ping them to see if they are active or inactive.

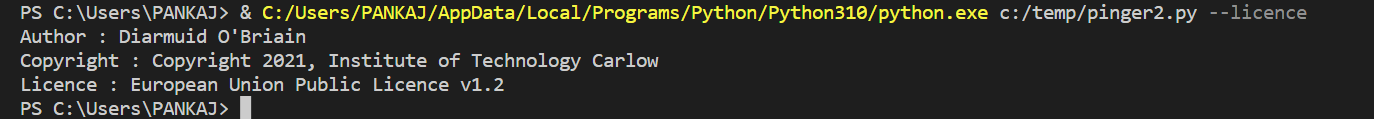
# APPENDICES

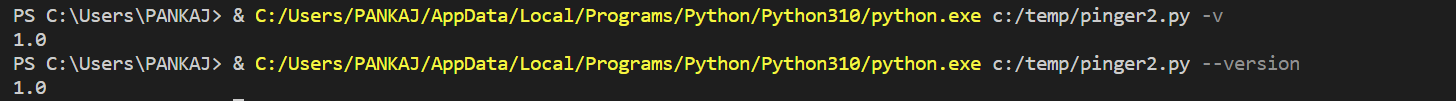
# Test showing examples of the code running.

# 

# Text Description automatically generated







Text

Description automatically generated

Text

Description automatically generated Shape

Description automatically generated with low confidence Text

Description automatically generated Text

Description automatically generated

# REFERENCES

1. Python.org. (2022). *os — Miscellaneous operating system interfaces — Python 3.11.0 documentation*. [online] Available at: https://docs.python.org/3/library/os.html [Accessed 17 Nov. 2022].
2. Python.org. (2022). *argparse — Parser for command-line options, arguments and sub-commands — Python 3.11.0 documentation*. [online] Available at: https://docs.python.org/3/library/argparse.html [Accessed 17 Nov. 2022].
3. Python.org. (2022). *An introduction to the ipaddress module — Python 3.11.0 documentation*. [online] Available at: https://docs.python.org/3/howto/ipaddress.html [Accessed 17 Nov. 2022].

‌

‌