Python Fundamental Project

Ng Jing Ren

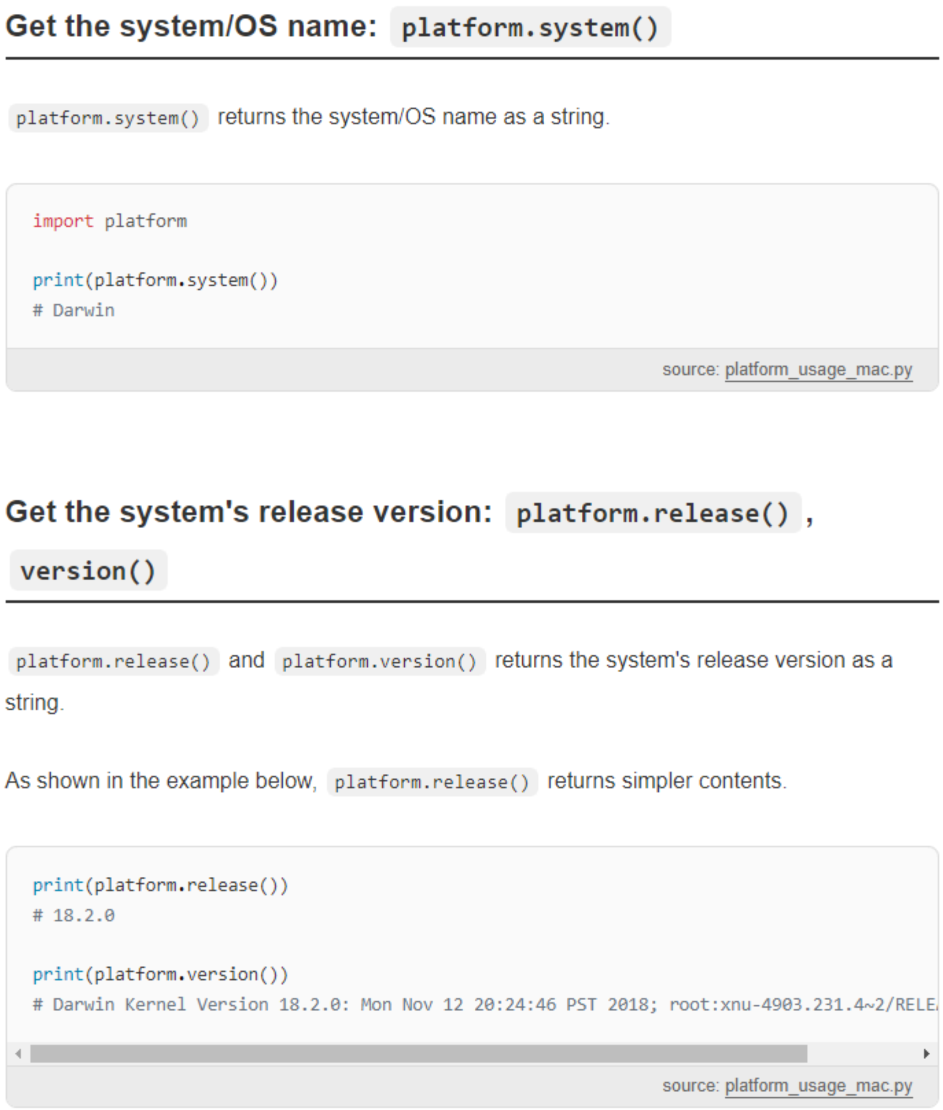
Table Of Content

Introduction

In this file, the first part presents how to display operating system details using Python commands. This includes showing the OS version, private IP address, public IP address, default gateway, hard disk information, top 5 directories, and CPU usage details. Additionally, I've included a script to look up country information using Python, which provides the relevant data.

Introduction

Research on obtaining OS information



Get the system/OS name: `platform.system()`

`platform.system()` returns the system/OS name as a string.

```python
import platform

print(platform.system())
# Darwin
```

source: platform_usage_mac.py

Get the system's release version: `platform.release()` , `version()`

`platform.release()` and `platform.version()` returns the system's release version as a string.

As shown in the example below, `platform.release()` returns simpler contents.

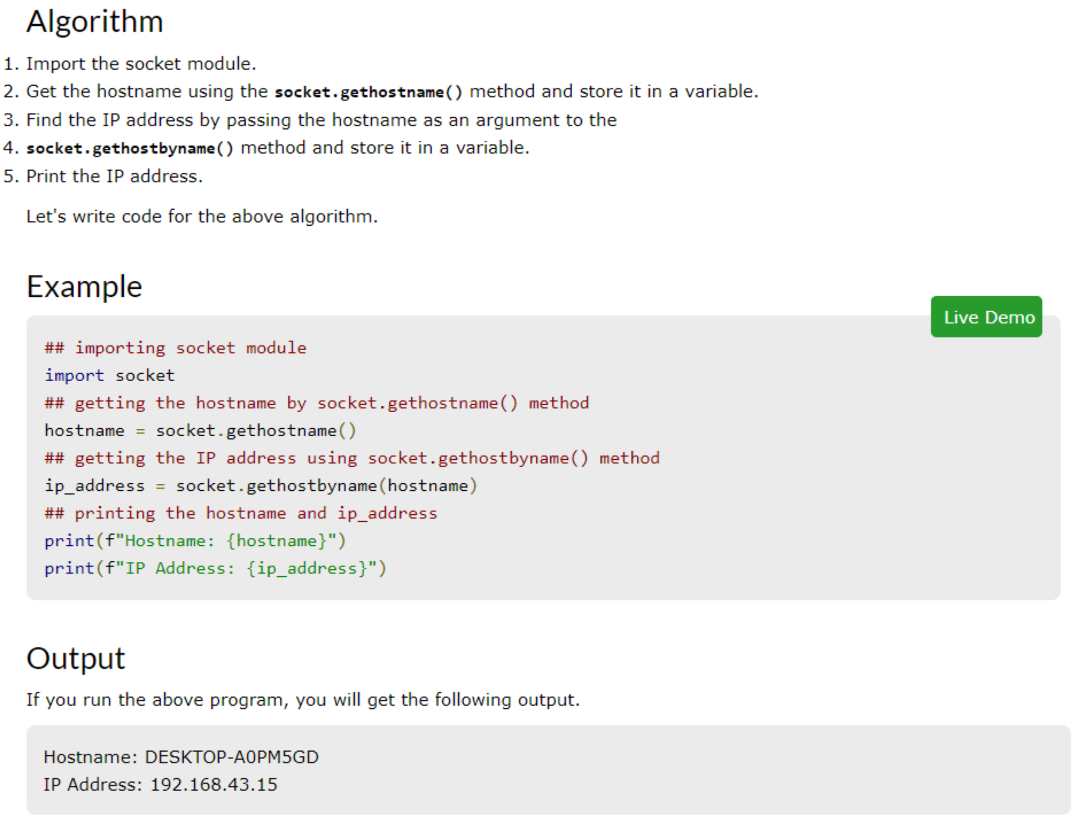
```python
print(platform.release())
# 18.2.0

print(platform.version())
# Darwin Kernel Version 18.2.0: Mon Nov 12 20:24:46 PST 2018; root:xnu-4903.231.4~2/RELE
```

source: platform_usage_mac.py

nkmk. “Get the OS Name and Version in Python.” *Nkmk Note*, 5 Feb. 2024, note.nkmk.me/en/python-platform-system-release-version/.

Introduction

Research on obtaining the private IP address



Algorithm

1. Import the socket module.
2. Get the hostname using the **socket.gethostname()** method and store it in a variable.
3. Find the IP address by passing the hostname as an argument to the
4. **socket.gethostbyname()** method and store it in a variable.
5. Print the IP address.

Let's write code for the above algorithm.

Example

```
## importing socket module
import socket
## getting the hostname by socket.gethostname() method
hostname = socket.gethostname()
## getting the IP address using socket.gethostbyname() method
ip_address = socket.gethostbyname(hostname)
## printing the hostname and ip_address
print(f"Hostname: {hostname}")
print(f"IP Address: {ip_address}")
```

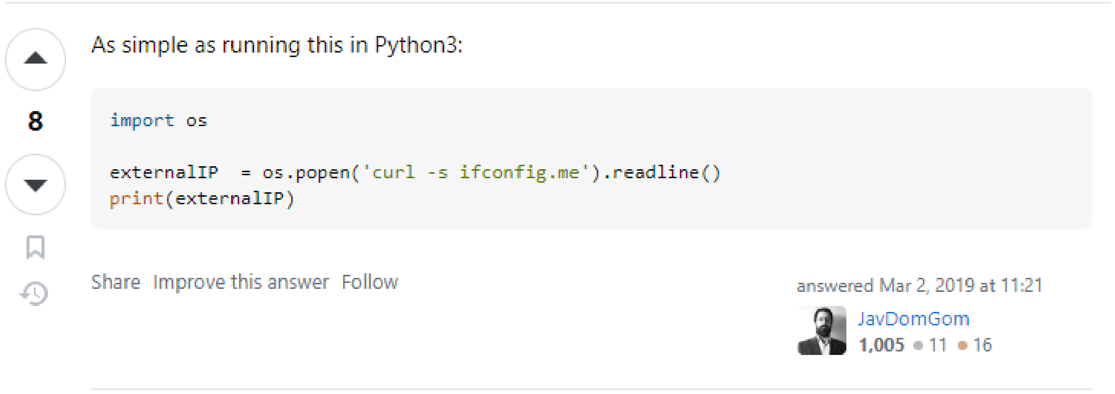
Output

If you run the above program, you will get the following output.

```
Hostname: DESKTOP-A0PM5GD
IP Address: 192.168.43.15
```

Elance, Pradeep. “Python Program to Find the IP Address of the Client.” *Tutorialspoint*, 25 Aug. 2023, www.tutorialspoint.com/python-program-to-find-the-ip-address-of-the-client.

Research on obtaining the public IP address

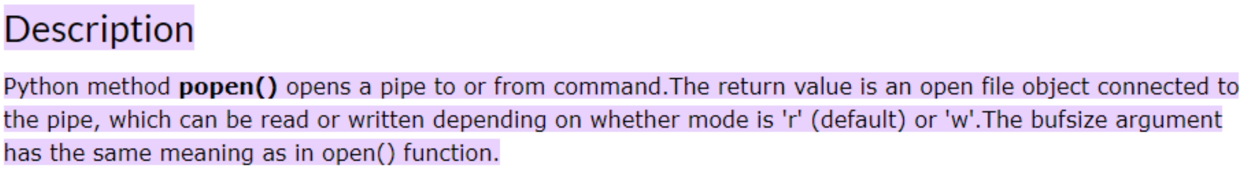


As simple as running this in Python3:

```
import os

externalIP  = os.popen('curl -s ifconfig.me').readline()
print(externalIP)
```

8
Share Improve this answer Follow
answered Mar 2, 2019 at 11:21
JavDomGom
1,005 ● 11 ● 16

Cit. “Getting a Machine’s External IP Address with Python.” *Stack Overflow*, 24 Nov. 2017, stackoverflow.com/questions/2311510/getting-a-machines-external-ip-address-with-python.

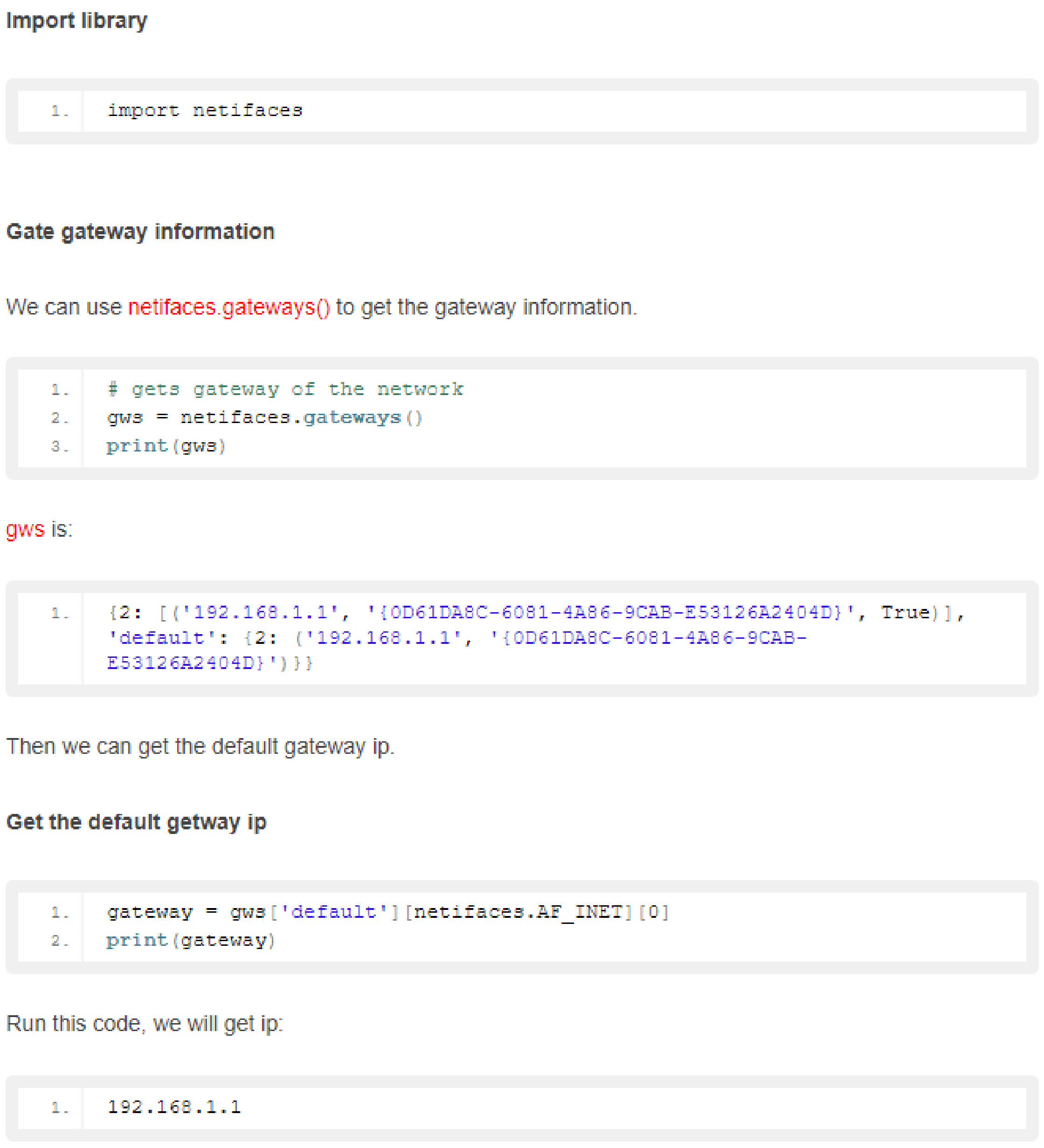


Description

Python method **popen()** opens a pipe to or from command.The return value is an open file object connected to the pipe, which can be read or written depending on whether mode is 'r' (default) or 'w'.The bufsize argument has the same meaning as in open() function.

Tutorials Point. “Python Os.Popen() Method.” *Tutorialspoint*, www.tutorialspoint.com/python/os_popen.htm#:~:text=Description,as%20in%20open()%20function.

Research on getting default gateway



Import library

```
import netifaces
```

Gate gateway information

We can use netifaces.gateways() to get the gateway information.

```
# gets gateway of the network
gws = netifaces.gateways()
print(gws)
```

gws is:

```
{2: [('192.168.1.1', '{0D61DA8C-6081-4A86-9CAB-E53126A2404D}', True)], 'default': {2: ('192.168.1.1', '{0D61DA8C-6081-4A86-9CAB-E53126A2404D}')}}
```

Then we can get the default gateway ip.

Get the default getway ip

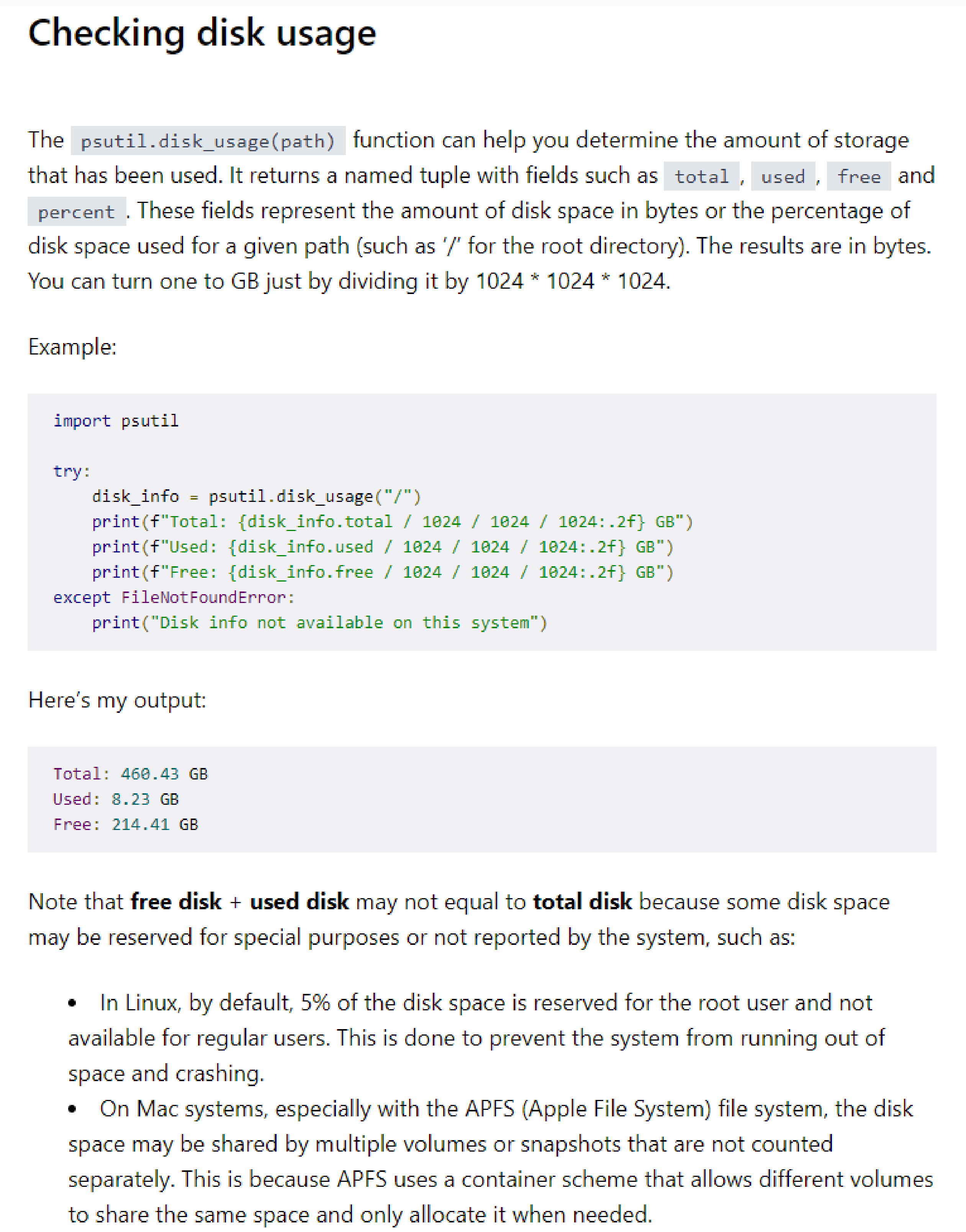
```
gateway = gws['default'][netifaces.AF_INET][0]
print(gateway)
```

Run this code, we will get ip:

```
192.168.1.1
```

Admin. “Python Get Default Gateway IP Using Netifaces: A Step Guide - Python Tutorial.” *Tutorial Example*, 1 Sept. 2020, www.tutorialexample.com/python-get-default-gateway-ip-using-netifaces-a-step-guide-python-tutorial/.

Research on obtaining hard disk information



Checking disk usage

The `psutil.disk_usage(path)` function can help you determine the amount of storage that has been used. It returns a named tuple with fields such as `total`, `used`, `free` and `percent`. These fields represent the amount of disk space in bytes or the percentage of disk space used for a given path (such as '/' for the root directory). The results are in bytes. You can turn one to GB just by dividing it by 1024 * 1024 * 1024.

Example:

```
import psutil

try:
    disk_info = psutil.disk_usage("/")
    print(f"Total: {disk_info.total / 1024 / 1024 / 1024:.2f} GB")
    print(f"Used: {disk_info.used / 1024 / 1024 / 1024:.2f} GB")
    print(f"Free: {disk_info.free / 1024 / 1024 / 1024:.2f} GB")
except FileNotFoundError:
    print("Disk info not available on this system")
```

Here's my output:

```
Total: 460.43 GB
Used: 8.23 GB
Free: 214.41 GB
```

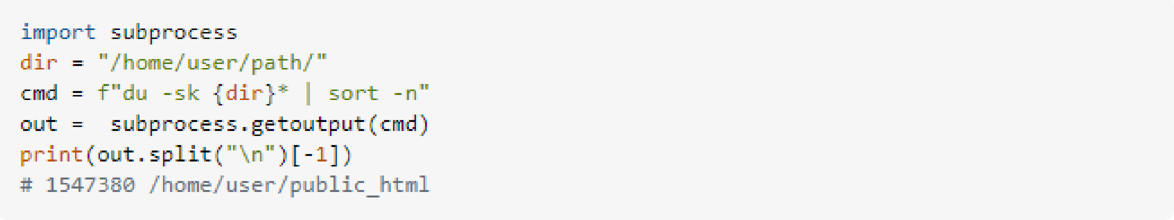
Note that **free disk** + **used disk** may not equal to **total disk** because some disk space may be reserved for special purposes or not reported by the system, such as:

- In Linux, by default, 5% of the disk space is reserved for the root user and not available for regular users. This is done to prevent the system from running out of space and crashing.
- On Mac systems, especially with the APFS (Apple File System) file system, the disk space may be shared by multiple volumes or snapshots that are not counted separately. This is because APFS uses a container scheme that allows different volumes to share the same space and only allocate it when needed.

Khue. "Python: Checking System RAM/CPU/Disk Usage." *Sling Academy*, 19 May 2023, www.slingacademy.com/article/python-checking-system-ram-cpu-disk-usage/.

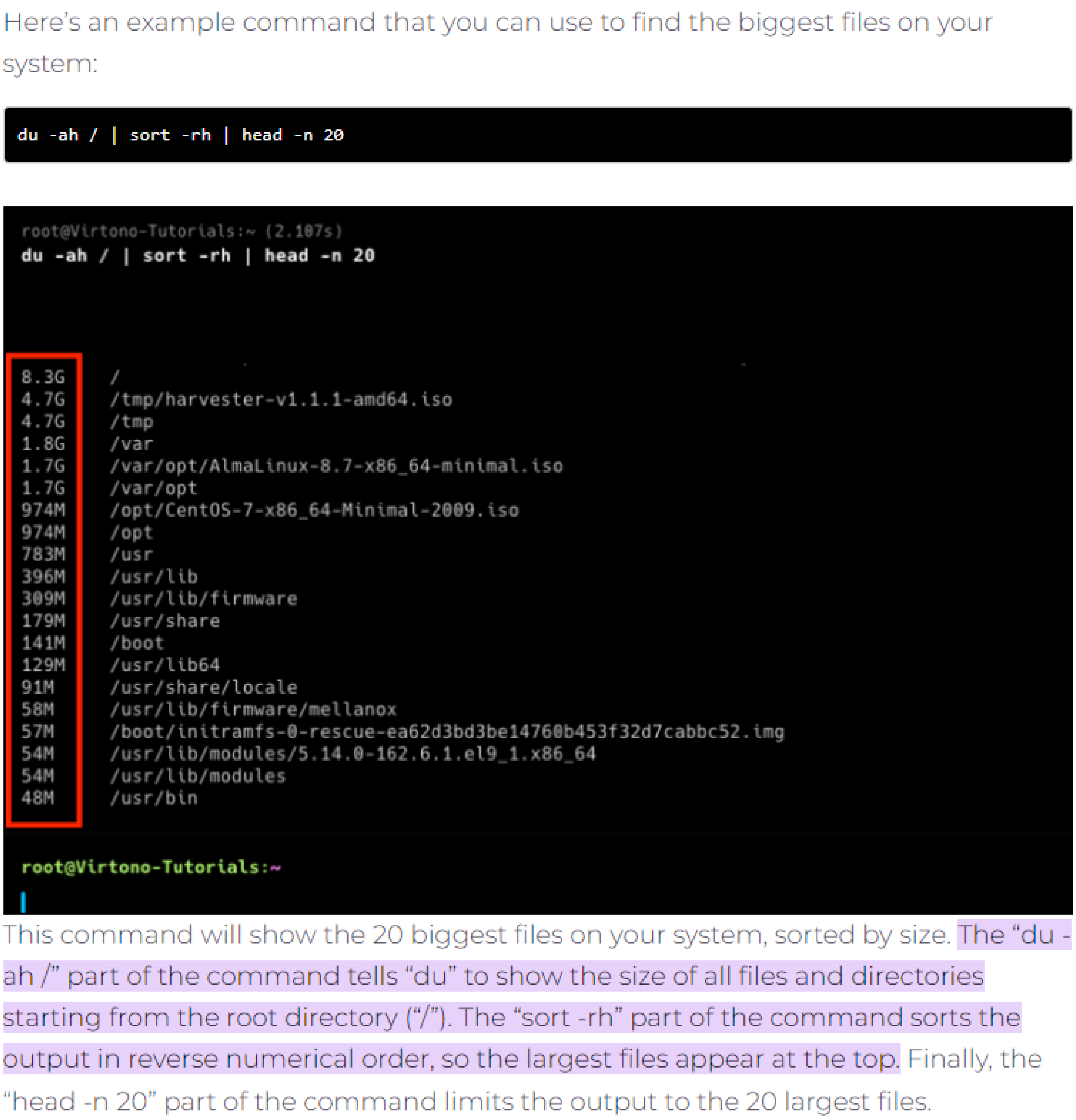
Introduction

Research on obtaining information about the top 5 directories



```
import subprocess
dir = "/home/user/path/"
cmd = f"du -sk {dir}* | sort -n"
out =  subprocess.getoutput(cmd)
print(out.split("\n")[-1])
# 1547380 /home/user/public_html
```

Lobito, Pedro. “Finding the Biggest Folder within a Directory Tree.” *Stack Overflow*, 14 Nov. 2019, stackoverflow.com/questions/58855112/finding-the-biggest-folder-within-a-directory-tree.



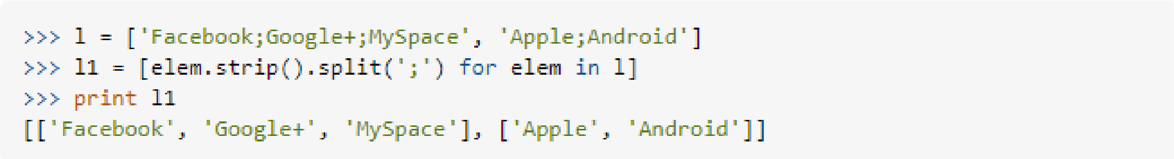
Here’s an example command that you can use to find the biggest files on your system:

```
du -ah / | sort -rh | head -n 20
```

```
root@Virtono-Tutorials:~ (2.107s)
du -ah / | sort -rh | head -n 20



8.3G    /
4.7G    /tmp/harvester-v1.1.1-amd64.iso
4.7G    /tmp
1.8G    /var
1.7G    /var/opt/AlmaLinux-8.7-x86_64-minimal.iso
1.7G    /var/opt
974M    /opt/CentOS-7-x86_64-Minimal-2009.iso
974M    /opt
783M    /usr
396M    /usr/lib
309M    /usr/lib/firmware
179M    /usr/share
141M    /boot
129M    /usr/lib64
91M     /usr/share/locale
58M     /usr/lib/firmware/mellanox
57M     /boot/initramfs-0-rescue-ea62d3bd3be14760b453f32d7cabbc52.img
54M     /usr/lib/modules/5.14.0-162.6.1.el9_1.x86_64
54M     /usr/lib/modules
48M     /usr/bin

root@Virtono-Tutorials:~
```

This command will show the 20 biggest files on your system, sorted by size. The “du -ah /” part of the command tells “du” to show the size of all files and directories starting from the root directory (“/”). The “sort -rh” part of the command sorts the output in reverse numerical order, so the largest files appear at the top. Finally, the “head -n 20” part of the command limits the output to the 20 largest files.

dlatikay. “How to Find the Largest Files in Linux.” *Virtono Community*, 12 Apr. 2023, www.virtono.com/community/tutorial-how-to/how-to-find-the-largest-files-in-linux/#:~:text=The%20%E2%80%9Cdu%20%2Dah%20%2F%E2%80%9D,files%20appear%20at%20the%20top.

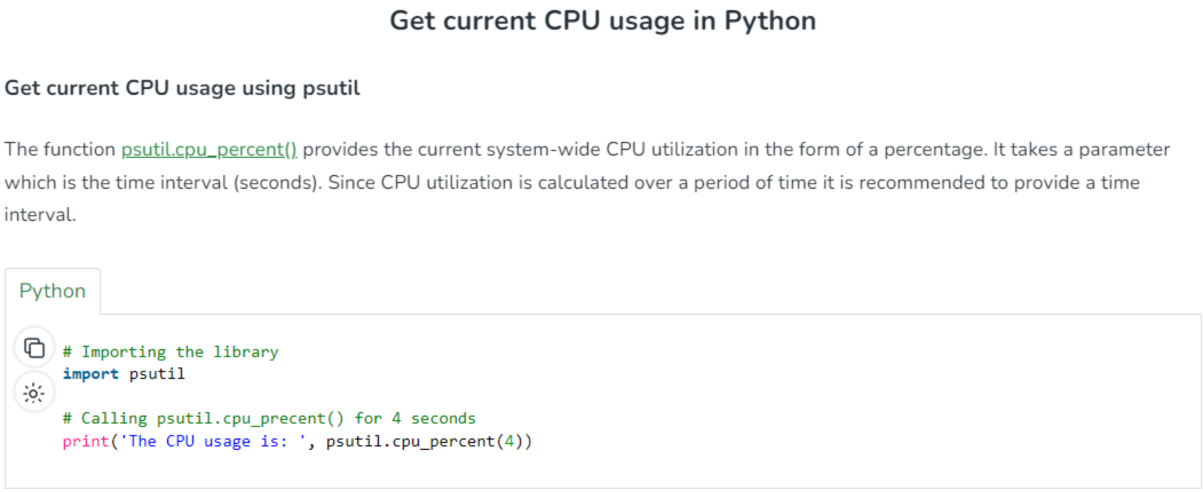


```
>>> l = ['Facebook;Google+;MySpace', 'Apple;Android']
>>> l1 = [elem.strip().split(';') for elem in l]
>>> print l1
[['Facebook', 'Google+', 'MySpace'], ['Apple', 'Android']]
```

Kalra, Sukrit. “Python 2: Attributeerror: ‘list’ Object Has No Attribute ‘Strip.’” *Stack Overflow*, 19 July 2013, stackoverflow.com/questions/17751322/python-2-attributeerror-list-object-has-no-attribute-strip.

Introduction

Researching how to obtain CPU usage information



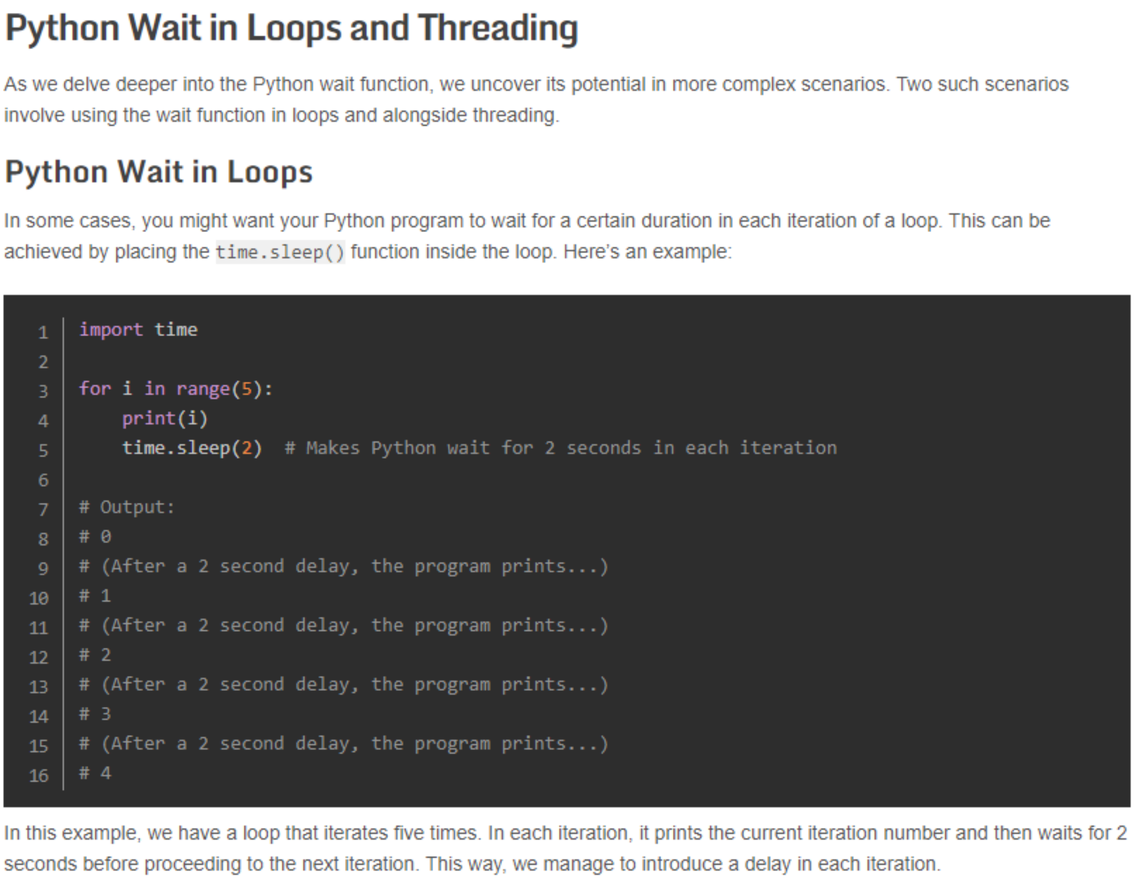
Get current CPU usage in Python

Get current CPU usage using psutil

The function psutil.cpu_percent() provides the current system-wide CPU utilization in the form of a percentage. It takes a parameter which is the time interval (seconds). Since CPU utilization is calculated over a period of time it is recommended to provide a time interval.

```python
# Importing the library
import psutil

# Calling psutil.cpu_precent() for 4 seconds
print('The CPU usage is: ', psutil.cpu_percent(4))
```

Goyal, Pranjal. “How to Get Current CPU and RAM Usage in Python?” *GeeksforGeeks*, GeeksforGeeks, 18 Oct. 2022, www.geeksforgeeks.org/how-to-get-current-cpu-and-ram-usage-in-python/.



Python Wait in Loops and Threading

As we delve deeper into the Python wait function, we uncover its potential in more complex scenarios. Two such scenarios involve using the wait function in loops and alongside threading.

Python Wait in Loops

In some cases, you might want your Python program to wait for a certain duration in each iteration of a loop. This can be achieved by placing the `time.sleep()` function inside the loop. Here’s an example:

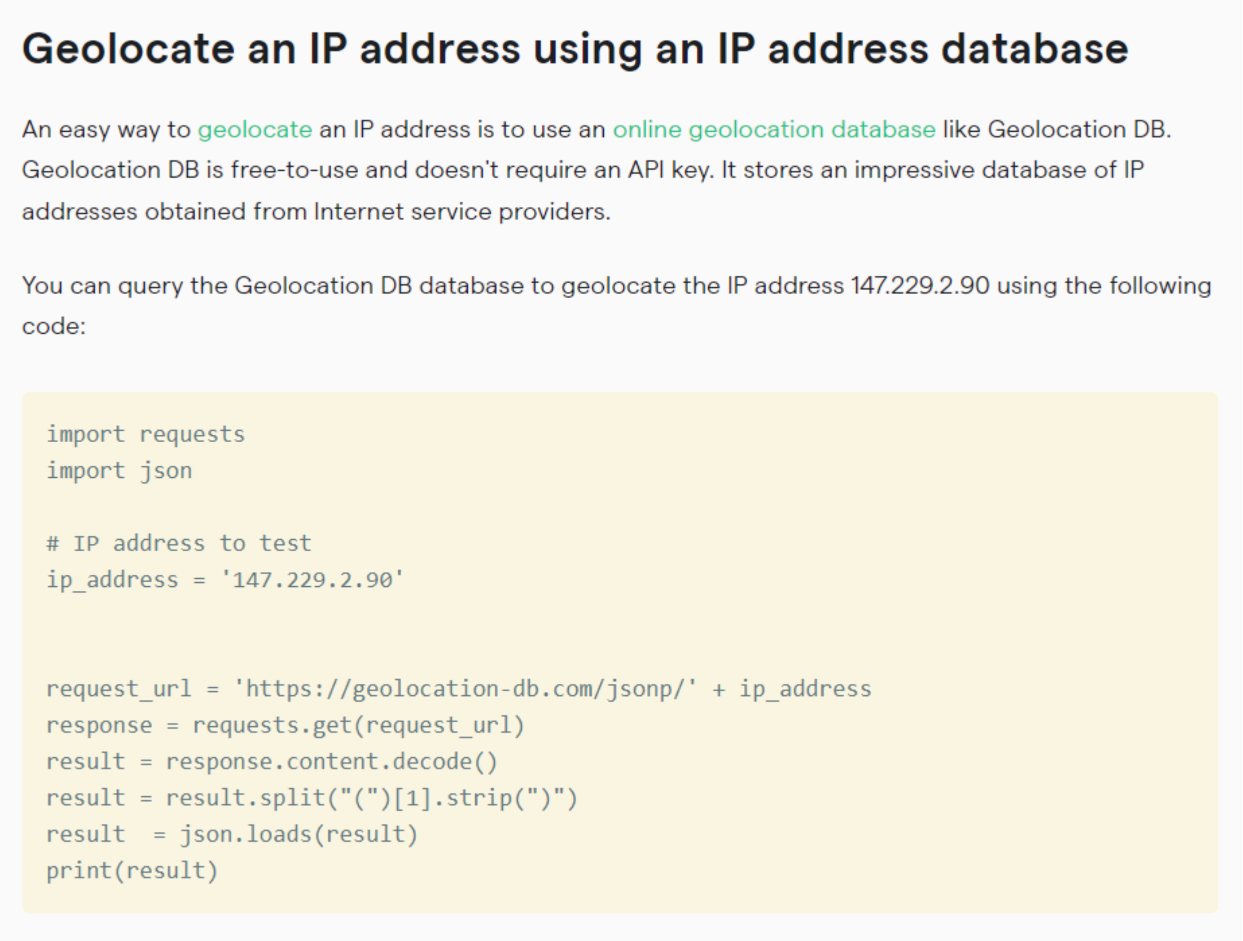
```python
import time

for i in range(5):
    print(i)
    time.sleep(2)  # Makes Python wait for 2 seconds in each iteration

# Output:
# 0
# (After a 2 second delay, the program prints...)
# 1
# (After a 2 second delay, the program prints...)
# 2
# (After a 2 second delay, the program prints...)
# 3
# (After a 2 second delay, the program prints...)
# 4
```

In this example, we have a loop that iterates five times. In each iteration, it prints the current iteration number and then waits for 2 seconds before proceeding to the next iteration. This way, we manage to introduce a delay in each iteration.

Ramuglia, Gabriel. “Python Wait: Pausing with Time.Sleep() and More.” *Linux Dedicated Server Blog*, 10 Sept. 2023, ioflood.com/blog/python-wait/.

Introduction

Researching how to look up IP country information



Geolocate an IP address using an IP address database

An easy way to geolocate an IP address is to use an online geolocation database like Geolocation DB. Geolocation DB is free-to-use and doesn't require an API key. It stores an impressive database of IP addresses obtained from Internet service providers.

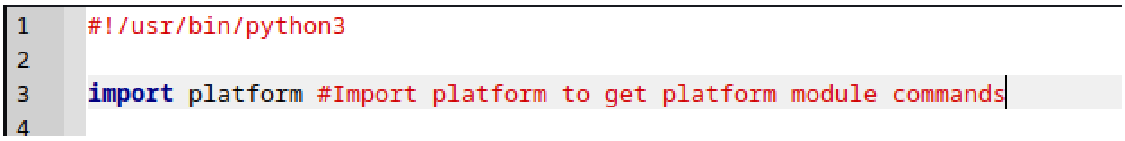
You can query the Geolocation DB database to geolocate the IP address 147.229.2.90 using the following code:

```
import requests
import json

# IP address to test
ip_address = '147.229.2.90'


request_url = 'https://geolocation-db.com/jsonp/' + ip_address
response = requests.get(request_url)
result = response.content.decode()
result = result.split("(")[1].strip(")")
result  = json.loads(result)
print(result)
```

Shipton, Elizabeth (Lizzie). “Python IP Geolocation: How to Geolocate an IP Address with Python.” *Python IP Geolocation: How to Geolocate an IP Address with Python*, AbstractAPI, 7 Aug. 2023, www.abstractapi.com/guides/ip-geolocation/how-to-geolocate-an-ip-address-in-python.

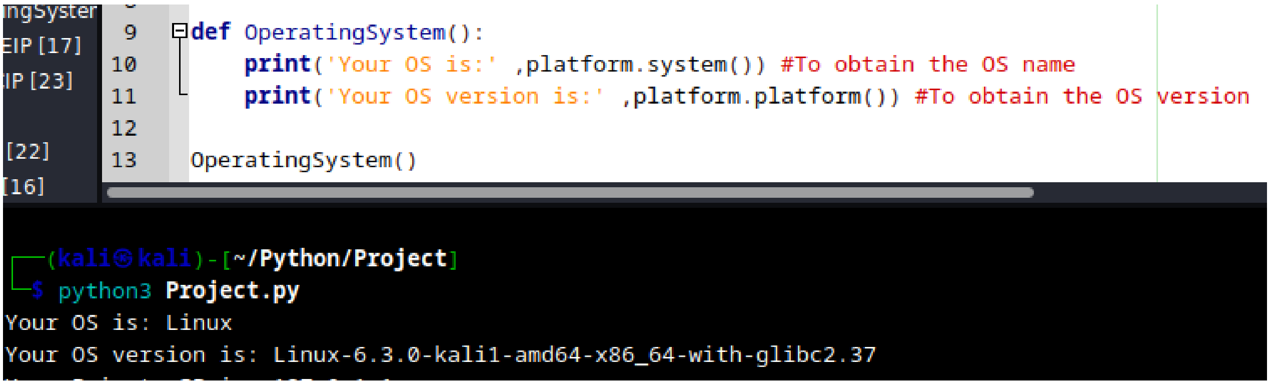
Displaying OS version

By importing the platform module, Python can utilize its functionality



```
#!/usr/bin/python3

import platform #Import platform to get platform module commands
```

The output in Linux

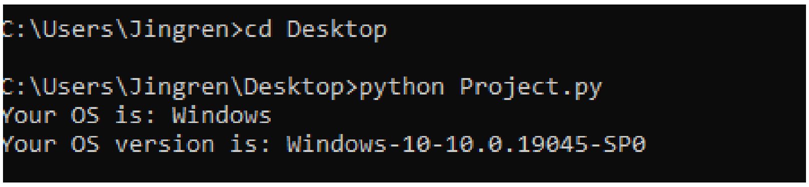


```
def OperatingSystem():
    print('Your OS is:' ,platform.system()) #To obtain the OS name
    print('Your OS version is:' ,platform.platform()) #To obtain the OS version

OperatingSystem()
```

```
┌──(kali㉿kali)-[~/Python/Project]
└─$ python3 Project.py
Your OS is: Linux
Your OS version is: Linux-6.3.0-kali1-amd64-x86_64-with-glibc2.37
```

Using platform.system() and platform.platform() to find the OS name and OS version.

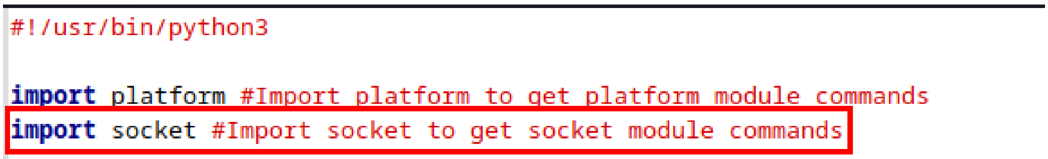
The output in Windows



```
C:\Users\Jingren>cd Desktop

C:\Users\Jingren\Desktop>python Project.py
Your OS is: Windows
Your OS version is: Windows-10-10.0.19045-SP0
```

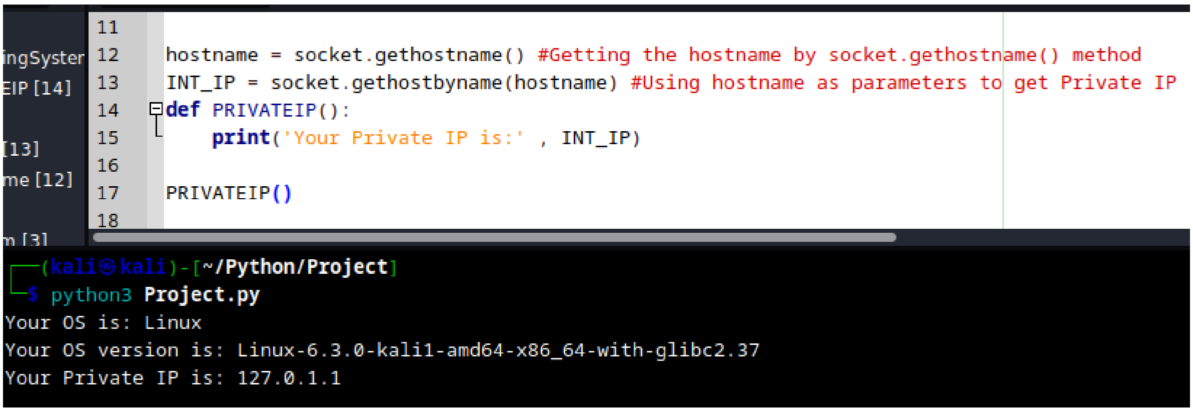
Displaying Private IP Address

By importing the socket module, Python can utilize its functionality



```
#!/usr/bin/python3

import platform #Import platform to get platform module commands
import socket #Import socket to get socket module commands
```

The output in Linux



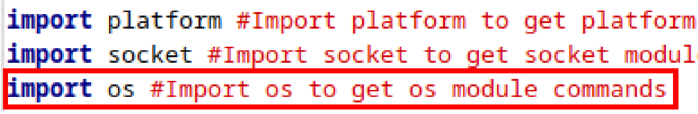
```

hostname = socket.gethostname() #Getting the hostname by socket.gethostname() method
INT_IP = socket.gethostbyname(hostname) #Using hostname as parameters to get Private IP
def PRIVATEIP():
    print('Your Private IP is:' , INT_IP)

PRIVATEIP()

```

```
┌──(kali㉿kali)-[~/Python/Project]
└─$ python3 Project.py
Your OS is: Linux
Your OS version is: Linux-6.3.0-kali1-amd64-x86_64-with-glibc2.37
Your Private IP is: 127.0.1.1
```

By using hostname = socket.gethostname() we ensure that we can later use it as a parameter to obtain the private IP address. The private IP address can be retrieved using socket.gethostbyname(hostname).

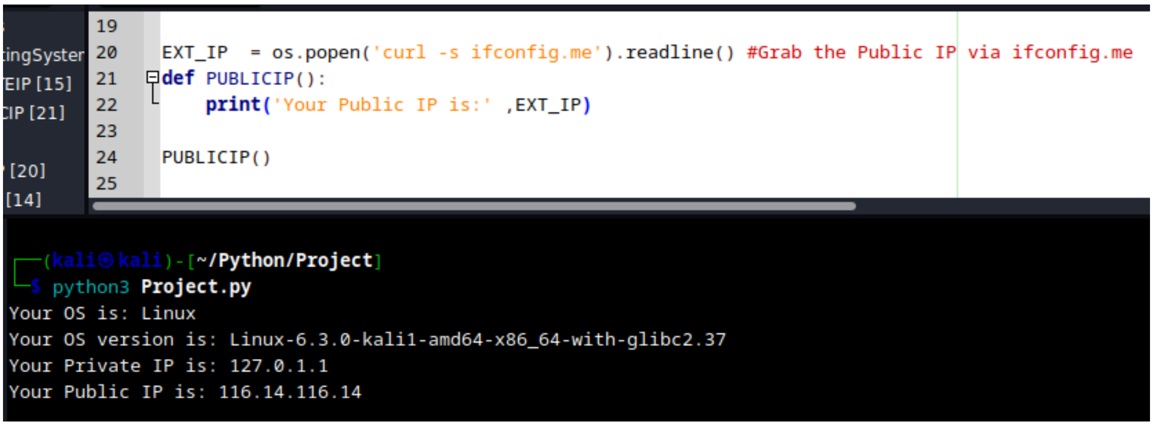
Displaying Public IP Address

By importing the os module, Python can use code from the os module



```
import platform #Import platform to get platform module commands
import socket #Import socket to get socket module commands
import os #Import os to get os module commands
```

The output in Linux



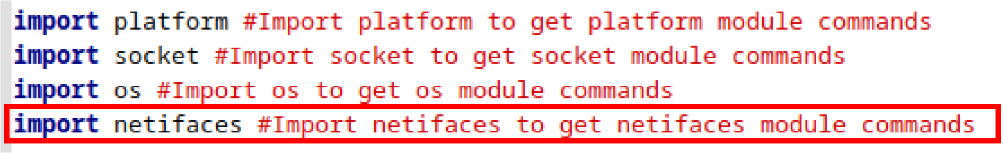
```

EXT_IP  = os.popen('curl -s ifconfig.me').readline() #Grab the Public IP via ifconfig.me
def PUBLICIP():
    print('Your Public IP is:' ,EXT_IP)

PUBLICIP()

```

```
┌──(kali㉿kali)-[~/Python/Project]
└─$ python3 Project.py
Your OS is: Linux
Your OS version is: Linux-6.3.0-kali1-amd64-x86_64-with-glibc2.37
Your Private IP is: 127.0.1.1
Your Public IP is: 116.14.116.14
```

By using popen, it opens a command to curl ifconfig.me.

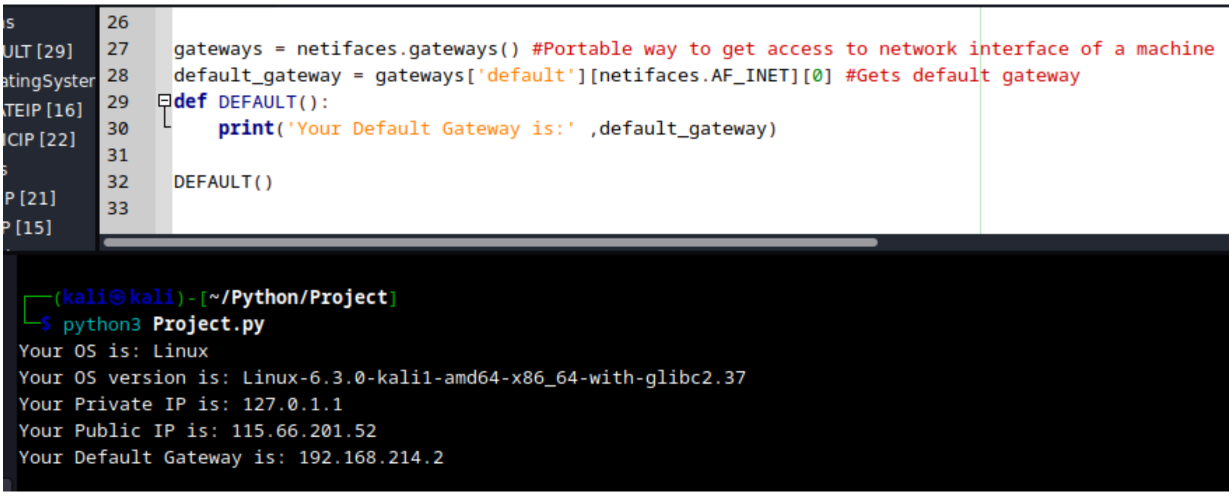
Displaying Default Gateway

By importing the netifaces module, Python can use code from the netifaces module



```
import platform #Import platform to get platform module commands
import socket #Import socket to get socket module commands
import os #Import os to get os module commands
import netifaces #Import netifaces to get netifaces module commands
```

The output in Linux



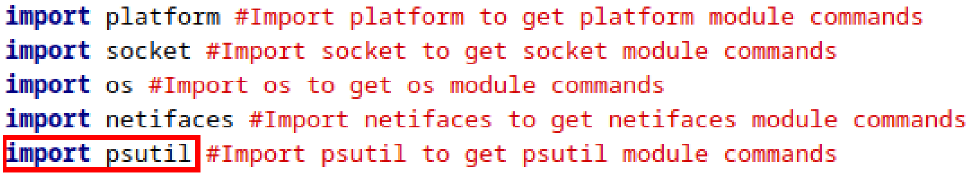
```

gateways = netifaces.gateways() #Portable way to get access to network interface of a machine
default_gateway = gateways['default'][netifaces.AF_INET][0] #Gets default gateway
def DEFAULT():
    print('Your Default Gateway is:' ,default_gateway)

DEFAULT()

```

```
┌──(kali㉿kali)-[~/Python/Project]
└─$ python3 Project.py
Your OS is: Linux
Your OS version is: Linux-6.3.0-kali1-amd64-x86_64-with-glibc2.37
Your Private IP is: 127.0.1.1
Your Public IP is: 115.66.201.52
Your Default Gateway is: 192.168.214.2
```

The netifaces.gateways() function allows Python to retrieve portable network interface information. Using gateways['default'][netifaces.AF_INET][0] allows Python to retrieve the default gateway using the gateways variable stored earlier.

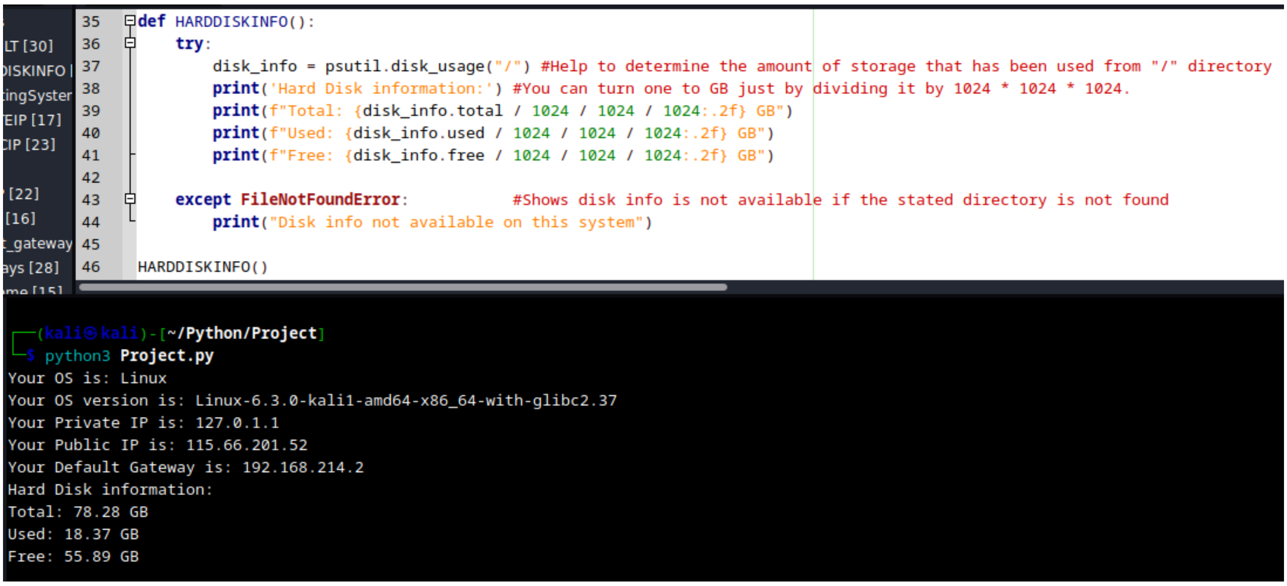
Displaying Hard Disk Information

By importing the psutil module, Python can utilize code from the psutil module



```
import platform #Import platform to get platform module commands
import socket #Import socket to get socket module commands
import os #Import os to get os module commands
import netifaces #Import netifaces to get netifaces module commands
import psutil #Import psutil to get psutil module commands
```

The output in Linux



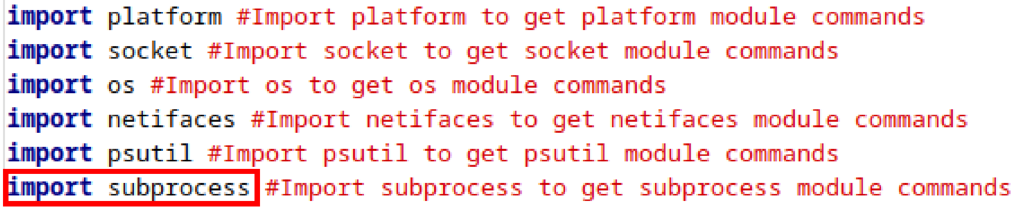
```
def HARDDISKINFO():
    try:
        disk_info = psutil.disk_usage("/") #Help to determine the amount of storage that has been used from "/" directory
        print('Hard Disk information:') #You can turn one to GB just by dividing it by 1024 * 1024 * 1024.
        print(f"Total: {disk_info.total / 1024 / 1024 / 1024:.2f} GB")
        print(f"Used: {disk_info.used / 1024 / 1024 / 1024:.2f} GB")
        print(f"Free: {disk_info.free / 1024 / 1024 / 1024:.2f} GB")

    except FileNotFoundError:          #Shows disk info is not available if the stated directory is not found
        print("Disk info not available on this system")

HARDDISKINFO()
```

```
┌──(kali㉿kali)-[~/Python/Project]
└─$ python3 Project.py
Your OS is: Linux
Your OS version is: Linux-6.3.0-kali1-amd64-x86_64-with-glibc2.37
Your Private IP is: 127.0.1.1
Your Public IP is: 115.66.201.52
Your Default Gateway is: 192.168.214.2
Hard Disk information:
Total: 78.28 GB
Used: 18.37 GB
Free: 55.89 GB
```

Using psutil.disk_usage allows the system to retrieve hard disk information. When using ("/"), it calculates the storage starting from the root directory. To simplify the method, I opted to display the storage in gigabytes (GB), where 1GB equals 1024 * 1024 * 1024.

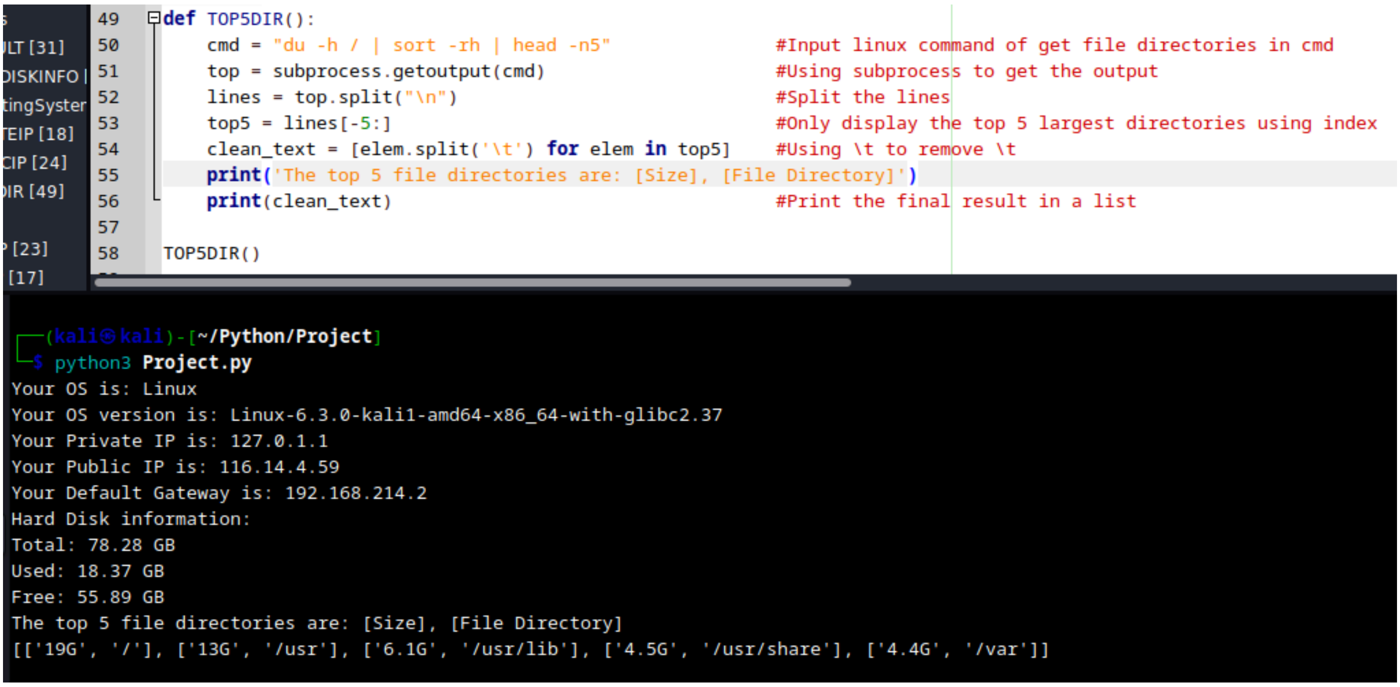
Displaying Top 5 Directories Information

By using import subprocess module, it will allow python to use the code that is from subprocess module



```
import platform #Import platform to get platform module commands
import socket #Import socket to get socket module commands
import os #Import os to get os module commands
import netifaces #Import netifaces to get netifaces module commands
import psutil #Import psutil to get psutil module commands
import subprocess #Import subprocess to get subprocess module commands
```

The output in Linux



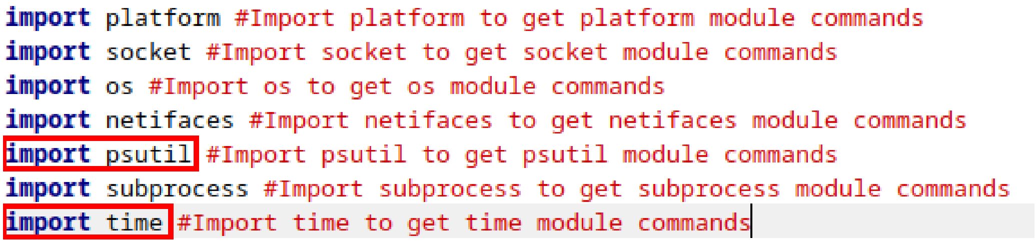
```
def TOP5DIR():
    cmd = "du -h / | sort -rh | head -n5"            #Input linux command of get file directories in cmd
    top = subprocess.getoutput(cmd)                  #Using subprocess to get the output
    lines = top.split("\n")                          #Split the lines
    top5 = lines[-5:]                                #Only display the top 5 largest directories using index
    clean_text = [elem.split('\t') for elem in top5] #Using \t to remove \t
    print('The top 5 file directories are: [Size], [File Directory]')
    print(clean_text)                                #Print the final result in a list

TOP5DIR()
```

```
┌──(kali㉿kali)-[~/Python/Project]
└─$ python3 Project.py
Your OS is: Linux
Your OS version is: Linux-6.3.0-kali1-amd64-x86_64-with-glibc2.37
Your Private IP is: 127.0.1.1
Your Public IP is: 116.14.4.59
Your Default Gateway is: 192.168.214.2
Hard Disk information:
Total: 78.28 GB
Used: 18.37 GB
Free: 55.89 GB
The top 5 file directories are: [Size], [File Directory]
[['19G', '/'], ['13G', '/usr'], ['6.1G', '/usr/lib'], ['4.5G', '/usr/share'], ['4.4G', '/var']]
```

Using subprocess as a module allows Python to execute Linux commands to retrieve file directory information. The command du -h is used to print file sizes in a human-readable format. Adding head -n5 ensures the command only displays the top 5 largest file directories. By using subprocess.getoutput(cmd), Python executes the command. I used top.split to remove lines and specified [-5:] to get the top 5. To eliminate \t between size and file directory, I used a for-loop with elem.split('\t') as listed above. The final result is stored in a list for readability.

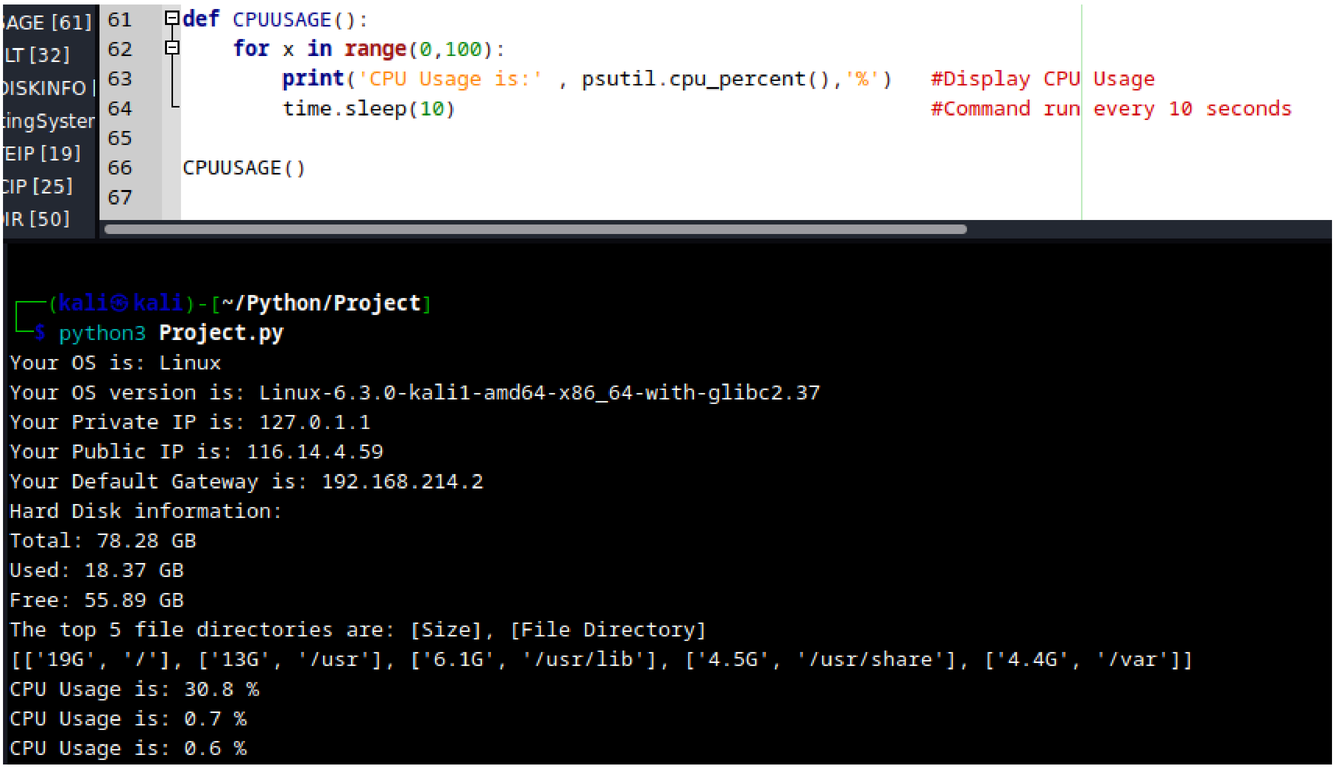
Displaying CPU Usage Information

By importing the time module, Python can utilize code from the time module. Additionally, for this section, we are also using psutil



```
import platform #Import platform to get platform module commands
import socket #Import socket to get socket module commands
import os #Import os to get os module commands
import netifaces #Import netifaces to get netifaces module commands
import psutil #Import psutil to get psutil module commands
import subprocess #Import subprocess to get subprocess module commands
import time #Import time to get time module commands
```

The output in Linux



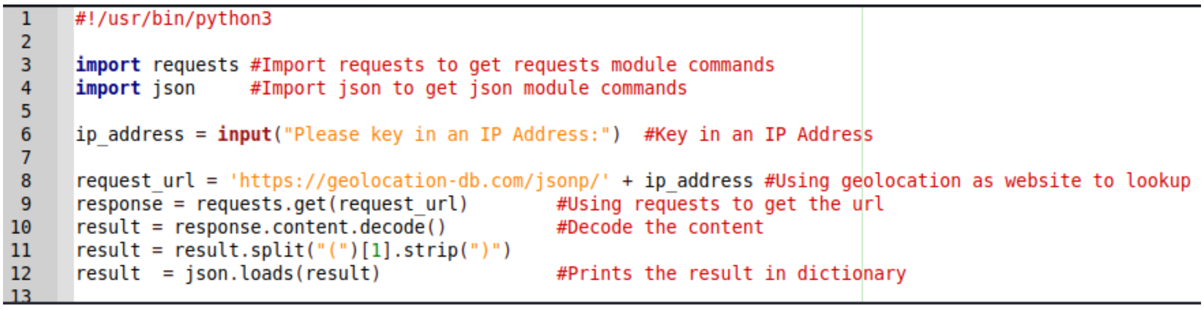
```
def CPUUSAGE():
    for x in range(0,100):
        print('CPU Usage is:' , psutil.cpu_percent(),'%')    #Display CPU Usage
        time.sleep(10)                                       #Command run every 10 seconds

CPUUSAGE()
```

```
┌──(kali㉿kali)-[~/Python/Project]
└─$ python3 Project.py
Your OS is: Linux
Your OS version is: Linux-6.3.0-kali1-amd64-x86_64-with-glibc2.37
Your Private IP is: 127.0.1.1
Your Public IP is: 116.14.4.59
Your Default Gateway is: 192.168.214.2
Hard Disk information:
Total: 78.28 GB
Used: 18.37 GB
Free: 55.89 GB
The top 5 file directories are: [Size], [File Directory]
[['19G', '/'], ['13G', '/usr'], ['6.1G', '/usr/lib'], ['4.5G', '/usr/share'], ['4.4G', '/var']]
CPU Usage is: 30.8 %
CPU Usage is: 0.7 %
CPU Usage is: 0.6 %
```

Using a for loop allows the script to continuously check CPU usage percentage. To display the percentage, use psutil.cpu_percent(). To refresh the command every 10 seconds, use time.sleep(10) to display CPU usage percent at regular intervals.

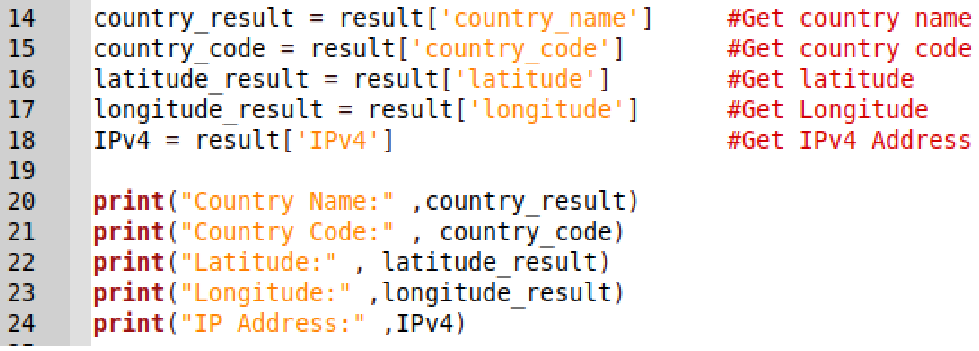
Looking Up The IP Country Information

Commands on looking up IP country information



```
#!/usr/bin/python3

import requests #Import requests to get requests module commands
import json     #Import json to get json module commands

ip_address = input("Please key in an IP Address:")  #Key in an IP Address

request_url = 'https://geolocation-db.com/jsonp/' + ip_address #Using geolocation as website to lookup
response = requests.get(request_url)         #Using requests to get the url
result = response.content.decode()           #Decode the content
result = result.split("(")[1].strip(")")
result  = json.loads(result)                 #Prints the result in dictionary

```

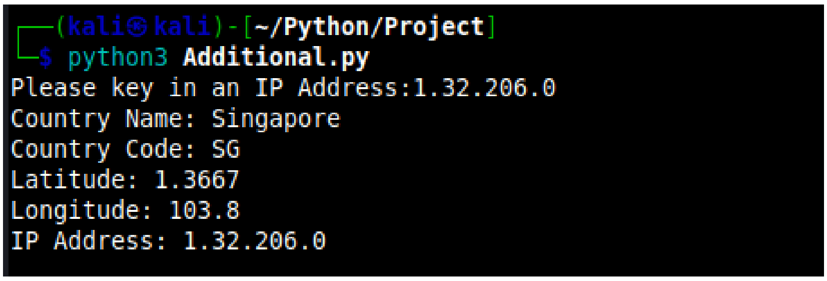
To run the script, we need to import modules from `requests` and `json`. For the IP address section, I ask the user to input an IP address for lookup. `request_url` is used to access a geolocation website to query the country database. The `response` variable stores the requests made to the geolocation service. The resulting content will be decoded and stored in JavaScript Object Notation (JSON) format to sort the objects. Finally, it will print the result as a dictionary.



```
country_result = result['country_name']     #Get country name
country_code = result['country_code']       #Get country code
latitude_result = result['latitude']        #Get latitude
longitude_result = result['longitude']      #Get Longitude
IPv4 = result['IPv4']                       #Get IPv4 Address

print("Country Name:" ,country_result)
print("Country Code:" , country_code)
print("Latitude:" , latitude_result)
print("Longitude:" ,longitude_result)
print("IP Address:" ,IPv4)
```

Using the dictionary result, I will print the country name, country code, latitude, longitude, and IP address separately instead of clustering them together in a single line.

The output in Linux



```
┌──(kali㉿kali)-[~/Python/Project]
└─$ python3 Additional.py
Please key in an IP Address:1.32.206.0
Country Name: Singapore
Country Code: SG
Latitude: 1.3667
Longitude: 103.8
IP Address: 1.32.206.0
```

References

nkmk. “Get the OS Name and Version in Python.” *Nkmk Note*, 5 Feb. 2024, note.nkmk.me/en/python-platform-system-release-version/.

Elance, Pradeep. “Python Program to Find the IP Address of the Client.” *Tutorialspoint*, 25 Aug. 2023, www.tutorialspoint.com/python-program-to-find-the-ip-address-of-the-client.

Cit. “Getting a Machine’s External IP Address with Python.” *Stack Overflow*, 24 Nov. 2017, stackoverflow.com/questions/2311510/getting-a-machines-external-ip-address-with-python.

Tutorials Point. “Python Os.Popen() Method.” *Tutorialspoint*, www.tutorialspoint.com/python/os_popen.htm#:~:text=Description,as%20in%20open()%20function.

Admin. “Python Get Default Gateway IP Using Netifaces: A Step Guide - Python Tutorial.” *Tutorial Example*, 1 Sept. 2020, www.tutorialexample.com/python-get-default-gateway-ip-using-netifaces-a-step-guide-python-tutorial/.

Khue. “Python: Checking System RAM/CPU/Disk Usage.” *Sling Academy*, 19 May 2023, www.slingacademy.com/article/python-checking-system-ram-cpu-disk-usage/.

Lobito, Pedro. “Finding the Biggest Folder within a Directory Tree.” *Stack Overflow*, 14 Nov. 2019, stackoverflow.com/questions/58855112/finding-the-biggest-folder-within-a-directory-tree.

dlatikay. “How to Find the Largest Files in Linux.” *Virtono Community*, 12 Apr. 2023, www.virtono.com/community/tutorial-how-to/how-to-find-the-largest-files-in-linux/#:~:text=The%20%E2%80%9Cdu%20%2Dah%20%2F%E2%80%9D,files%20appear%20at%20the%20top.

Kalra, Sukrit. “Python 2: Attributeerror: ‘list’ Object Has No Attribute ‘Strip.’” *Stack Overflow*, 19 July 2013, stackoverflow.com/questions/17751322/python-2-attributeerror-list-object-has-no-attribute-strip.

Goyal, Pranjal. “How to Get Current CPU and RAM Usage in Python?” *GeeksforGeeks*, GeeksforGeeks, 18 Oct. 2022, www.geeksforgeeks.org/how-to-get-current-cpu-and-ram-usage-in-python/.

Ramuglia, Gabriel. “Python Wait: Pausing with Time.Sleep() and More.” *Linux Dedicated Server Blog*, 10 Sept. 2023, ioflood.com/blog/python-wait/.

Shipton, Elizabeth (Lizzie). “Python IP Geolocation: How to Geolocate an IP Address with Python.” *Python IP Geolocation: How to Geolocate an IP Address with Python*, AbstractAPI, 7 Aug. 2023, www.abstractapi.com/guides/ip-geolocation/how-to-geolocate-an-ip-address-in-python.