

# Python Project: Python OS INFO

**ARUN DASSE - s13 - cfc2407**  
**Lecturer name: James**

## Objective: Create automation to display the operating system information.

```
print('PROJECT: Python OS Info')
print('ARUN DASSE - s13')
print('Lecturer: JAMES')
```

```
(arun@kali) - [~/pf/pyproject]
$ python py_project.py
PROJECT: Python OS Info
ARUN DASSE - s13
Lecturer: JAMES
```

**Part 1:** Display the OS version - if windows, display the windows details; if executed on Linux, display the Linux details. [Reference credit: <https://www.w3resource.com/python-exercises/python-basic-exercise-43.php>]

### Steps:

- Importing the Module = python codes that was already created. The codes can be imported and used in your current environment / script.
- For the part 1: Two modules were imported: os and platform;
- The OS module in Python provides functions for interacting with the operating system.
- The platform module in Python is used to access the underlying platform's data, such as, hardware, operating system, etc.
- The commands `os.name()`, `platform.system()` and `platform.release()` helps to display the name of the operating system and its version.

```
import os
import platform

print("\nSection 1 displays the Linux Details:")
print("\nName of the operating system:",os.name)
print("\nName of the OS system:",platform.system())
print("\nVersion of the operating system:",platform.release())
```

Section 1 displays the Linux Details:

Name of the operating system: posix

Name of the OS system: Linux

Version of the operating system: 5.18.0-kali5-amd64

## **Part 2:** Display the private IP address, public IP address, and the default gateway.

### **Steps:**

- With the help of the command `os.system(<repective linux command to display the ip address>)`.
- This sections display the private IP, public IP address and default gateway.
- Using the Linux commands 'grep' and 'awk' (inside `os.system`) to display the exact info needed.

```
print("\nSection 2 displays the IP addresses and Deafult Gateway:")

print("\nThe private IP address of the system is:")
print(os.system("ifconfig | grep broadcast | awk '{print $2}'"))

print("\nThe public IP address of the system is:")
print(os.system("curl ifconfig.io") )

print("\nThe Default Gateway is:")
print(os.system("route | grep default | awk '{print $2}'"))
```

Section 2 displays the IP addresses and Deafult Gateway:

The private IP address of the system is:  
192.168.216.128

The public IP address of the system is:  
137.132.212.131

The Default Gateway is:  
192.168.216.2

**Part 3:** Display the hard disk size, free and used space. [Reference credits: [https://www.geeksforgeeks.org/python-shutil-disk\\_usage-method/#:~:text=disk\\_usage\(\)%20method%20in%20Python,attributes%20total%2C%20used%20and%20free.](https://www.geeksforgeeks.org/python-shutil-disk_usage-method/#:~:text=disk_usage()%20method%20in%20Python,attributes%20total%2C%20used%20and%20free.)]

### Steps:

- Shutil module in Python provides many functions of high-level operations on files and collections of files.
- Defining the path, in this case, I am provide my current working directory path.
- With the help of the command 'shutil.disk\_usage(path)' stored to a variable stat - we can get the info of hard disk size, used and free space.
- Print the variable stat to display the info.

```
print("\nSection 3 displays the Hard disk size, used and free space:")

import shutil

# Path
path = "/home/arun/pf/pyproject"

# Get the disk usage statistics
# about the given path
stat = shutil.disk_usage(path)

# Print disk usage statistics
print("\nDisk usage statistics:")
print(stat)
```

Section 3 displays the Hard disk size, used and free space:

Disk usage statistics:  
usage(total=84053143552, used=14256721920, free=65479700480)

## Part 4: Display the top five directories and their size

### Steps:

- The first command used in this section is `os.getcwd()` to print the current working directory
- Before displaying the specific directory/files, using the command `os.listdir()` to list out all the files/subfolders inside the current working directory.
- Saving all the files/folders to a variable `dir_items`.
- Initiating the for loop to be executed for every item (file or folder) saved to the variable `dir_items`.
- Next step is to check the if condition: if the size of any file/folder inside `dir_items` is greater than 15000 bytes.
- Then it will display only the folders satisfies the condition set.

```
print("\nSection 4 displays the top five directories and their sizes:")
print("\nThe current working directory is: ",os.getcwd())
print("\nDirectories in current working directory is: ",os.listdir())
print("\nThe top five directories in the current working folder is: ")
dir_items=os.listdir('/home/arun/pf/pyproject')
for eachitem in dir_items:
    if os.path.getsize(eachitem) > 15000:
        print(os.path.getsize(eachitem),'bytes', eachitem)
```

```
Section 4 displays the top five directories and their sizes:
```

```
The current working directory is: /home/arun/pf/pyproject
```

```
Directories in current working directory is: ['Dir 10', 'testcases', 'Dir2', 'Dir3', 'pythonfolder', 'lab04', 'Dir1', 'NR', 'Dir4', 'Dir 8', 'Dir 9', 'Dir5', 'Dir7', 'socproject', 'lab05', 'Dir6', 'py_project.py', 'nrproject', 'SOC', 'LF']
```

```
The top five directories in the current working folder is:
```

```
20480 bytes Dir 10
20480 bytes testcases
20480 bytes Dir3
20480 bytes Dir5
20480 bytes LF
```

**Part 5:** Display the CPU usage; refresh every 10 seconds [Reference Credits:  
<https://www.geeksforgeeks.org/python-script-that-is-executed-every-5-minutes/>]

### Steps:

- Importing psutil module which helps to keep track of various resources utilization in the system. Usage of resources like CPU, memory, disks, network, sensors can be monitored.
- The function psutil.cpu\_percent() provides the current system-wide CPU utilization in the form of a percentage.
- The given interval in the time.sleep() function is 10 seconds and make while loop is true.
- The function will sleep for the given time interval. After that, it will start executing and display the CPU utilization info.
- pip install psutil - Before import psutil; install the psutil in Linux/Ubuntu if the module not found.

```
import psutil
import time
print("\nSection 5 displays the CPU usage for every 10 seconds:")
while(True):
    print(psutil.cpu_percent())
    time.sleep(10)
```

Press ctrl+c to end the operation.

```
Section 5 displays the CPU usage for every 10 seconds:
0.0
2.2
1.9
1.2
1.1
1.2
1.2
1.1
1.2
1.4
1.1
1.0
1.1
0.8
1.3
```