Python Session 1

==============================================================

# Script to print statements

print("Hello Python")

print('How r u')

print("Let's go for lunch")

print('He said,"I am busy today"')

print('He said,"Let\'s go for lunch"')

print("He said,\"Let's go for lunch\"")

======================================================================

Single line comments start with #

Multiple line comments can start with ''' and end with '''

(or) start """ and end with """

=========================================================================

#Adddition and Concatination

print(2+5)

print("Hello"+"Python")

print("Hi"+"5")

print("hi"+str(5))

print("8"+"5")

print("8"+str(5))

print(int("8")+5)

print("Hello","Python")

print("Hi",5)

=======================================================================

#if conditions

#Program to find the smallest of 3 numbers

a = 10

b = 20

c= 5

if a < b and a < c:

print("a is the smallest ")

elif b < c:

print("b is the smallest")

else:

print("c is the smallest")

print("Program finished")

========================================================================

Loops

=================

#Program to display the even numbers between 1-10

i = 2

while i <= 10:

print(i)

i = i + 2

==========================================================================

#Program to display the 2 tables

i = 1

while i <= 10:

print(2,"\*",i,"=",2\*i)

i = i + 1

===========================================================================

Python Session 2

===========================================================================

=================================================================

Python Session 2

================================================================

For Loops

----------------

a=[1,4,6,9,10]

for i in a:

print(i)

tools=["Docker","Jenkins","Git","Ansible"]

for x in tools:

print(x)

===============================================================

#To display the sum of numbers in a list

numbers=[10,50,90,40,60]

sum = 0

for i in numbers:

sum = sum + i

print("The total sum of numbers is: ",sum)

===================================================================

#To find the max number in a list

numbers=[10,50,90,40,60]

max = 0

for i in numbers:

if max < i:

max = i

print("The maximum number in the list is :",max)

==================================================================

#Program to find the number of even and odd

numbers=[1,2,3,4,5,6,7,8,9,10,11]

even = 0

odd = 0

for i in numbers:

if i % 2 == 0:

even = even + 1

print("Even :",i)

else:

odd = odd + 1

print("Odd :"i)

print("The even number count is :",even)

print("The odd number count is :",odd)

========================================================================

Read Write operations on Files

=======================================

#To create a new file and write into it

text ="\nThis is a python session"

file=open("C:/Users/gandh/OneDrive/Desktop/file1.txt","w")

file.write(text)

#To append we can open in append mode using 'a'

text ="\nThis is a python session"

file=open("C:/Users/gandh/OneDrive/Desktop/file1.txt","a")

file.write(text)

#To read all the content of a file

file=open("C:/Users/gandh/OneDrive/Desktop/file1.txt","r")

text = file.read()

print(text)

#To read the content line by line

file=open("C:/Users/gandh/OneDrive/Desktop/file1.txt","r")

text = file.readlines()

for x in text:

print(x)

#To copy all the content of a file into another file

file1=open("C:/Users/gandh/OneDrive/Desktop/file1.txt","r")

file2=open("C:/Users/gandh/OneDrive/Desktop/file2.txt","w")

text = file1.read()

file2.write(text)

#To copy alternate lines of code from one file to another

file1=open("C:/Users/gandh/OneDrive/Desktop/file1.txt","r")

file3=open("C:/Users/gandh/OneDrive/Desktop/file3.txt","a")

text=file1.readlines()

i = 0

while i < len(text):

file3.write(text[i])

i = i + 2

==========================================================================

Python Session 3

==========================================================================

Functions in Python

Functions are used to create reusable code

def add(x,y):

print(x+y)

def sub(x,y):

print(x-y)

def mul(x,y):

print(x\*y)

def div(x,y):

print(x/y)

add(7,8)

add(100,200)

mul(5,2)

=========================================================================

User defined function to copy the content of a file to another file

def copy\_file(srcfile,destfile):

file1=open(srcfile,'r')

file2=open(destfile,'w')

data = file1.read()

file2.write(data)

copy\_file("C:\\Users\\gandh\\OneDrive\\Desktop\\linux.txt","C:\\Users\\gandh\\OneDrive\\Desktop\\newlinux.txt")

=============================================================================

Classes and Objects

----------------------------

class Calculator:

def add(self,a,b):

print(a+b)

def sub(self,a,b):

print(a-b)

def mul(self,a,b):

print(a\*b)

def div(self,a,b):

print(a/b)

def exp(self,a,b):

print(a\*\*b)

c = Calculator()

c.mul(5,4)

c.sub(10,9)

==============================================================================

Modules in Python

------------------------

Modules are code libraries

import math

import calendar

a = math.sqrt(16)

print(a)

b = math.factorial(5)

print(b)

c = calendar.isleap(2020)

print(c)

d = calendar.month(2021,2)

print(d)

==================================================================

Creating customised modules

-------------------------------

Create a python program "dimesnsions.py"

def area(length,breadth):

print(length\*breadth)

def perimeter(length,bredth):

print(2\*(length+bredth))

To use the above module in another python program

import dimensions

dimensions.area(10,5)

dimensions.perimeter(10,5)

===============================================================

Program to accept a password and check if it is 6-12 characters long

It should have lower case and upper case alphabets

It should have number and the special character @ # $

import re

p = input("Enter some password: ")

x = True

while x:

if (len(p)<6 or len(p)>12):

break

elif not re.search("[a-z]",p):

break

elif not re.search("[A-Z]",p):

break

elif not re.search("[0-9]",p):

break

elif not re.search("[$#@]",p):

break

else:

print("Valid password")

x= False

if x:

print("Invalid password")

===========================================================================

Python Session 4

===========================================================================

Python docker integrations

==================================

Python script to sownload any image

import subprocess

image = input("Enter the image name: ")

subprocess.call("docker pull %s"%(image),shell=True)

============================================================

Python script to delete any container

import subprocess

container = input("Enter the name of the container: ")

subprocess.call("docker rm -f %s"%(container),shell=True)

====================================

Python script to create a container based on interactive inputs

import subprocess

image=input("Enter the imagename: ")

name=input("Enter some name for container: ")

ports=input("Enter the ports to be mapped: ")

detach=input("Do you want to run in detached mode y/n: ")

if detach=='y':

subprocess.call("docker run --name %s -d -p %s %s"%(name,ports,image),shell=True)

elif detach=='n':

subprocess.call("docker run --name %s -p %s %s"%(name,ports,image),shell=True)

else:

print("Invalid option")

=====================================

Python script to delete only tomcat containers

import subprocess

subprocess.call('docker rm -f $(docker container ls | grep tomcat | cut -d " " -f 1)',shell=True)

===================================

Python script ot create multiple containers

import subprocess

image=input("Enter the imagename: ")

count=input("Enter the number of containers that should be created: ")

i = 1

while i <= int(count):

subprocess.call("docker run --name container%d -P -d %s"%(i,image),shell=True)

i = i + 1

========================================================================

Python Session 5

=========================================================================

Python script to delete all docker images

import subprocess

subprocess.call('docker images | cut -d " " -f 1 > file1',shell=True)

images = open("file1",'r').readlines()

i = 1

while i < len(images):

img = images[i]

subprocess.call("docker rmi %s"%(img),shell=True)

i = i + 1

==========================================================================

Python script to create a multi container docker architecture

import subprocess

subprocess.call("docker network create --driver bridge intelliqit",shell=True)

subprocess.call("docker run --name mydb -d -e MYSQL\_ROOT\_PASSWORD=intelliqit --network intelliqit mysql:5",shell=True)

subprocess.call("docker run --name mywordpress -d -p 8989:80 --network intelliqit wordpress",shell=True)

~

~

================================================================================

Python script to create 5 networks and create 5 nginx containers

on those networks

import subprocess

i = 1

while i <= 5:

subprocess.call("docker network create --driver bridge intelliqit%d"%(i),shell=True)

subprocess.call("docker run --name nginx%d --network intelliqit%d -d -P nginx"%(i,i),shell=True)

i = i + 1

~

==================================================================================

Interactive program in python to create services in docer swarm

import subprocess

image=input("Enter the image name: ")

service=input("Enter some name for service: ")

port=input("Enter the ports to be mapped: ")

replicas=input("Enter the number of replicas: ")

subprocess.call("docker service create --name %s -p %s --replicas %s %s"%(service,port,replicas,image),shell=True)

=========================================================================

PythonSession 6

Python script to perfrom autoscalling of services in swarm based

on available free memory

vim script.py

import subprocess

subprocess.call("free -m | grep Mem | awk '{print $4}' > file2",shell=True)

free=int(open("file2",'r').read())

if free >= 150 and free < 250:

subprocess.call("docker service scale webserver=3",shell=True)

elif free >=250 and free < 500:

subprocess.call("docker service scale webserver=6",shell=True)

else:

subprocess.call("docker service scale webserver=10",shell=True)

Set this script in crontab

crontab -e

\* \* \* \* \* /root/script.py

================================================================================

Python automation on Jenkins

--------------------------------

1 Create an ubuntu20 instance and install Jenkins

2 Install pip

sudo apt-get install python3-pip

3 Install python-jenkins

sudo pip3 install python-jenkins

Script to see info about all jobs in jenkins

========================================================

import jenkins

j = jenkins.Jenkins("http://localhost:8080","admin","admin")

print(j.get\_jobs())

===========================================================================

Script to run a jenkins job

-----------------------------------------------------------------

import jenkins

j = jenkins.Jenkins("http://localhost:8080","admin","admin")

print(j.build\_job("Development"))

=============================================================

Script to create a jenkins job

import jenkins

j = jenkins.Jenkins("http://localhost:8080","admin","admin")

j.create\_job("Sample1",jenkins.EMPTY\_CONFIG\_XML)

-----------------------------------------------------------------

Script to delete a jenkins job

======================================

import jenkins

j = jenkins.Jenkins("http://localhost:8080","admin","admin")

j.delete\_job("NewDevelopment")

=====================================================================

Script to create multpiple jenkins jobs

----------------------------------------------------------

import jenkins

j = jenkins.Jenkins("http://localhost:8080","admin","admin")

i = 1

while i <= 10:

j.create\_job("Sample%d"%i,jenkins.EMPTY\_CONFIG\_XML)

i = i + 1

---------------------------------------------------------------------------

Script to take backup of multiple jenkins jobs

import jenkins

j = jenkins.Jenkins("http://localhost:8080","admin","admin")

i = 1

while i <= 10:

j.copy\_job("Sample%d"%i,"NewSample%d"%i)

i = i + 1

===============================================================================

Script to take delete multiple jenkins jobs

import jenkins

j = jenkins.Jenkins("http://localhost:8080","admin","admin")

i = 1

while i <= 10:

j.delete\_job("Sample%d"%i)

i = i + 1

~

======================================================================