Python 1 - Activity

Test below questions (Q1-Q20) in your python shell

Q1. How to test python version in command line?

C:\Users\User>python -V

Python 3.7.1

C:\Users\User>python --version

Python 3.7.1

root@krosum:~# python -V

Python 2.7.3

root@krosum:~# python --version

Python 2.7.3

root@krosum:~#

root@krosum:~# python3 -V

Python 3.2.3

root@krosum:~# python3 --version

Python 3.2.3

root@krosum:~#

Q2. Using print() function , display your name and working city name

Ex: Hi myself karthikeyan from chennai.

>>> print("My self karthikeyan from chennai")

My self karthikeyan from chennai

>>>

>>> print('My self karthikeyan from chennai')

My self karthikeyan from chennai

>>>

Q3. How to use single line comment and multiline comment in python?

```
 # single line comment
 >>> '''
... This is multiline
... comment
... in python
... programming
... '''
>>> """
... This is multiline
... comment
... in python
... programming
... """
```

Q4. Write a python program

Step 1: declare a variable name called **server**, initialize value as your working operating system name.

Step 2: using **print()** function - display following message

Hi, my working kernel is Linux.

```
>>> server="Linux"
>>> print("Hi, My working kernel is:",server)
Hi, My working kernel is: Linux
```

Q5. How to combine below 3 string values into single string?

S1="My Server Name:"

S2="SunOs"

S3="Version is 6.5"

Note: use string concatenation operator

```
>>> S1="My Server Name:"
>>> S2="SunOs"
>>> S3="Version is 6.5"
>>> S1+S2+S3
'My Server Name:SunOsVersion is 6.5'
>>> print(S1+S2+S3)
My Server Name:SunOsVersion is 6.5
```

Q6. How to write multiline string statements in python? (Write all possible ways)

```
>>> print("""String1
... String2
... String3
... String4
... String5""")
String1
String2
String3
String4
String5
>>> print("String1\nString2\nString3\nString4\nString5")
String1
String2
String3
String4
String5
>>>
```

Q7. How to use escape characters (\n\t) in python?

```
>>> print("List of servers\nLinux\nSunOs\nAix\tHpux\nWinx")
List of servers
Linux
SunOs
Aix     Hpux
Winx
```

Q8. Using single print() function display your shell name, kernel name, login name details line by line.

Note: use \n characters.

```
>>> print("My shell name is:/bin/bash\nKernel name is:Linux\nLogin Name is:root")
My shell name is:/bin/bash
Kernel name is:Linux
Login Name is:root
```

Q9. How to calculate below expressions?

A) V="100"

F=200

V+F

```
>>> V="100"
>>> F=200
>>> type(V)
<class 'str'>
>>> type(F)
<class 'int'>
>>> int(V)+F
300
>>> print("Total value:",int(V)+F)
Total value: 300
```

B) Size="356\n"

Total=1000.35

Size+Total

```
>>> Size="356\n"
>>> Total=1000.35
>>> int(Size)+Total
1356.35
>>> print("Total:",int(Size)+Total)
Total: 1356.35
```

Q10. How to convert int, float data types in to string type?

```
>>> n=50
>>> F=1002.456
>>> type(n)
<class 'int'>
>>> type(F)
<class 'float'>
>>> str(n)
'50'
>>> str(F)
'1002.456'
>>> str(n)   '50'
>>> str(F)   '1002.456'
```

Q11. How to determine python data type?

```
>>> type (variable) (or) type(value)
```

Q12. What is difference between / and // operator in python?

// is the floored-division operator

/ Classic division operator

```
>>> 456/6
76.0
>>> 456//6
76
>>>
>>> 456.789/4
114.19725
>>>
>>> 456.789//4
114.0
>>>
```

Q13. Using string repetition operator (*) how to display lines in below format.

```
    ----------------------------------------

      ****************************************

    ----------------------------------------
```

```
>>> s='-'*45
>>> a='*'*50
>>> a='-'*45
>>> b='*'*50
>>> c='-'*45
>>> print(a)
---------------------------------------------
>>> print(b)
**************************************************
>>> print(c)
---------------------------------------------
>>>
>>> print(a,"\n",b,"\n",c)
      ---------------------------------------------
 **************************************************
      ---------------------------------------------
```

Q14. Given variable is **port=3036**

How to validate port number range is above 3000?

(say True/False)

>>> port=3036

>>> port >3000

True

Q15. **name="root"**

How to test user login is root user or not?

(say True/False)

>>> name= "root"

>>> name == "root"

True

Q16. Identify the error messages:-

a) Fname="p1.log"

print("File name is:",FNAME)

FNAME – NameError - undefined variable

print(“File name is:”,Fname)

b) print("File name is:"Fname)

, (comma) is missing

print(“File name is:”,Fname)

c) print(File name is:Fname)

Syntax Error

print(“File name is:”,Fname)

d) print ("File name is:Fname')

SyntaxError: EOL while scanning string literal

print(“File name is:Fname”)

e) Fsize=1234

print("File name is:",Fname,"FileSize is:"FSize)

Syntax Error: invalid syntax , is missing

>>> print("File name is:",Fname,"File Size is:",Fsize)

File name is: p1.log File Size is: 1234

f) print("File Size is:"+Fsize)

TypeError: can only concatenate str (not "int") to str

>>> print("File Size is:"+str(Fsize))

File Size is:1234

g) F="13.4567"

print("F value is:",int(F))

ValueError: invalid literal for int() with base 10: '13.4567'

>>> print("F value is:",int(float(F)))

F value is: 13

>>> float(F)

13.4567

>>> int(float(F))

13

Q17. Predict the output of below codes?

a) s="Linux\nWinx\tSunOs\nAix\tUnix"

print(type(s))

print(s)

>>> s="Linux\nWinx\tSunOs\nAix\tUnix"

>>> print(type(s))

<class 'str'>

>>> print(s)

Linux

Winx SunOs

Aix Unix

b) s1="admin"

s1 != "root"

>>> s1 != "root"

False

c) Tag="Test"

print("Tag Name is:"+Tag+str(1)+"\n")

print("Tag Name is:"+Tag+str(2)+"\n")

```
>>> Tag="Test"
>>> "Tag Name is:"+Tag+str(1)+"\n"
'Tag Name is:Test1\n'
>>> "Tag Name is:"+Tag+str(2)+"\n"
'Tag Name is:Test2\n'
>>> print("Tag Name is:"+Tag+str(1)+"\n")
Tag Name is:Test1
>>> print("Tag Name is:"+Tag+str(2)+"\n")
Tag Name is:Test2
```

d) IP='127.0.0.1'

port=8000

print("Running server name:"+IP+"\tport number is:"+str(port))

```
>>> IP='127.0.0.1'
>>> port=8000
>>>
>>> print("Running server name:"+IP+"\t Port number is:"+str(port))
Running server name:127.0.0.1      Port number is:8000
```

Q18) Say Yes/No

a) is python dynamic type programming? **- YES**

b) is python portable? **- YES**

c) is python not supported winx? **- NO (**python is portable language**)**

d) is python not supported floating point operations? **- NO**

e) is python is case sensitive language? **- YES**

Q19) using **type()** function determine below types?

a) type(str(10+20))
<class 'str'>

b) type(10+20.0)
<class 'float'>

c) type('')
<class 'str'>

d) type(100>200)
<class 'boolean'>

e) type (int("46")+int(10.335))
<class 'int'>

Q20) In python which function is used to read input from keyboard and what will be that function default return type?

input() – function is interface to keyboard.

input() – default return is string(str)

Example:

>>> type(input())

<class 'str'>

****

Task - create a new python file and run a program

Q1. Write a python program (create a filename p1.py)

Step 1: read an employee name from <STDIN> (keyboard)

read an employee ID from <STDIN> (keyboard)

read an employee Cost from <STDIN> (keyboard)

Step 2: determine input data type and display value to monitor.

Note: understand difference between type() vs print() usages.

root@krosum:~# cat -n E1.py

```
1  ename=input("Enter a emp name:")
2  eid=input("Enter a emp id:")
3  ecost=input("Enter a emp cost:")
4  print(type(ename),ename)
5  print(type(eid),eid)
6  print(type(ecost),ecost)
```

root@krosum:~# python3 E1.py

Enter a emp name:**Vishnu**

Enter a emp id:**345**

Enter a emp cost:**12345.678**

<class 'str'> Vishnu

<class 'str'> 345

<class 'str'> 12345.678

Q2. Write a python program (create a filename p2.py)

Step 1: read an any two disk name from <STDIN>*(ex: /dev/sda1,/dev/sda2)*

Step 2: read an any two disk size from <STDIN>*(ex: 100 200)*

Step 3: calculate sum of 2 disks.

Step 4: display individual disk name disk size to monitor.

Step 5: display total disk size to monitor.

```
root@krosum:~# cat -n E2.py
 1  dp1=input("Enter a partition:")
 2  ds1=input("Enter {} partition size:".format(dp1))
 3  dp2=input("Enter a partition:")
 4  ds2=input("Enter {} partition size:".format(dp2))
 5
 6  total=int(ds1)+int(ds2)
 7
 8  print("Disk Name:\tSize:")
 9  print("-"*35)
10  print(dp1,ds1)
11  print(dp2,ds2)
12  print("-"*35)
13  print("Total Disk Size:",total)
14
```

root@krosum:~# python3 E2.py

Enter a partition:**/dev/sda1**

Enter /dev/sda1 partition size:**120**

Enter a partition:**/dev/sda2**

Enter /dev/sda2 partition size:**200**

Disk Name: Size:

/dev/sda1 120

/dev/sda2 200

Total Disk Size: 320

Q3. Write a python program (create a filename p3.py)

STEP 1: import an os module into your local file (***import os***)

STEP 2: read a system command from <STDIN>

STEP 3: using **os.system ()** function; execute your input command in python.

Note: system () function return value

```
root@krosum:~# cat -n E3.py
     1  import os
     2  v=input("Enter a system command:")
     3  os.system(v)
root@krosum:~# python3 E3.py
Enter a system command: uptime
 15:35:51 up 18 min,  1 user,  load average: 0.27, 0.24, 0.23
root@krosum:~# python3 E3.py
Enter a system command: uname
Linux
root@krosum:~# python3 E3.py
Enter a system command: ps -f
UID        PID  PPID  C STIME TTY          TIME CMD
root      3032  2979  0 15:24 pts/0    00:00:00 su -
root      3040  3032  0 15:24 pts/0    00:00:00 -su
root      3221  3040  1 15:35 pts/0    00:00:00 python3 E3.py
root      3222  3221  0 15:35 pts/0    00:00:00 sh -c ps -f
root      3223  3222  0 15:35 pts/0    00:00:00 ps -f
```