Python Programming

Basic python programming

1. Execute the below on python IDLE/python shell and analyze the output for yourself!

**Duration:** less than a minute

>>> import this

1. Check this link :

**Duration:** read at your leisure

http://stackoverflow.com/questions/4504487/the-zen-of-python-distils-the-guiding-principles-for-python-into-20-aphorisms-bu

1. Identify what is there at the end of the output

**Duration:** less than a minute

>>> a = 10 \*\* 100

>>> a

That’s “L” which indicates that it’s a long integer. Python will automatically handle it as a long. So how long can the “int” be? Execute the below to get that information

>>> import sys

>>> sys.maxint

The output is the max “int”. Which basically says that your number range is from –(sys.maxint-1) to sys.maxint

1. Consider **str** to be a variable containing string. **len(str)** gives the length of a string. Describe the valid indices range of string in terms of len. Choose the right answer

**Duration:** less than a minute

1. –ve len(str) to len(str) -1
2. –ve len(str) -1 to len(str) -1
3. –ve len(str) to len(str)
4. Execute the following

**Duration:** less than a minute

>>> a = "Hello Bangalore“

>>> a[0]= “h”

1. Execute the following. What do you observe?

**Duration:** less than a minute

>>> a = "Hello Bangalore“

>>> a.replace(‘H’, ‘h’)

>>> a

1. Execute the following. What do you observe?

**Duration:** less than a minute

>>> a = "Hello Bangalore“

>>> a.replace(‘a’, ‘e’)

>>> a

>>> a.replace(‘a’,’e’,1)

1. Execute the following and understand that the start, end , step are all optional

# ignoring the end index

>>> a[0:]

'0123456789'

# ignoring the start index

>>> a[:5]

'01234'

# ignoring start, end and step

>>> a[::]

'0123456789'

# ignoring start, end but providing a stepper

>>> a[::2]

'02468'

# reverses a string

>>> a[::-1]

'9876543210'

**Duration:** 5 minutes

1. Execute the following and find out that negative indices also work

**Duration:** less than a minute

>>> a=['a','b','c','d']

>>> del a[-1]

>>> a

>>> a.pop(-1)

1. index() function on list will give the index of an item. Remove the item ‘b’ using index and remove function. Deliberate on why this should not be done!

**Duration:** less than a minute

>>> a=['a','b','c','d']

1. Get your string back. Note that the function join is on the string and not on list. Str(a) will not give you what you wanted! Try out different strings with join and arrive at a comma separated string from a list of items

**Duration** : 2 minutes

>>> mystr = 'abcdef'

>>> a = list(mystr)

>>> a

>>> ''.join(a)

>>> str(a)

1. In place list reversal

**Duration:** less than a minute

>>> a = ['a', 'b', 'c', 'd', 'e', 'f']

>>> a.reverse()

1. Execute the following and understand how different it is to create a tuple with only one element.

**Duration: two minutes**

>>> a\_tuple = 't1','t2','t3'

>>> a\_tuple

>>> b\_tuple = a\_tuple[0:1]

>>> b\_tuple

>>> type(b)

>>> a = ('hi')

>>> a

>>> type(a)

>>> a = ('hi',)

>>> a

>>> type(a)

1. Execute the below and be surprised and wonder how it made sense

**Duration:** less than a minute

>>> a = ['ab','bc','cd','de']

>>> dict(a)

(Hint : Remember a string is actually a sequence of characters?)

1. Use the list() function on a dictionary and find out the result

**Duration:** less than a minute

>>> list(a\_d)

>>> list(a\_d.items())

1. Execute the following. get() is more graceful and it allows you to return a default value

**Duration:** 2 minutes

>>> a\_d={'city': 'Bangalore', 'name': 'Aditya S P', 'email': 'sp.aditya@gmail.com'}

>>> a\_d['country']

>>> a\_d.get('country', 'Not Found')

1. Loop through the dictionary below and print both the keys and values of the same

**Duration:** 2 minutes

>>> a\_d={'city': 'Bangalore', 'name': 'Aditya S P', 'email': 'sp.aditya@gmail.com'}

1. Execute the below and understand why it worked. (hint : check question # 15 )

**Duration:** 1 minute

>>> a='somestr'

>>> for itr in a:

print itr

1. Write a function with takes a mandatory single argument and any number of arguments. In other words a function which takes in 1 or more parameters

**Duration:** 5 minutes

1. Execute the below function and deliberate on what you find. **Duration:** 2 minutes

globals()

locals()

1. Create a module and use it

Duration: 15 minutes

Step1 : create a file called mymodule.py and store it in a folder of your choice

Step2 : Let the contents of the file be as follows

import sys

print “arguments passed”, sys.argv

Step3 : Open command prompt and navigate to the folder where you have stored the above mymodule.py file

Step4 : type

C:\Users\aditya\Desktop>first.py

OR

C:\Users\aditya\Desktop>python first.py

1. Execute the below and find out what is tell() doing

>>> while True:

mytext=fh.read(chunk)

print fh.tell()

if not mytext:

break

1. range() only works with integers. Create a generator which returns float . let the default start be 0.5 and let it take and end and an epsilon value

for eg. float\_gen(start=0.5,end=0.6,epsilon=0.01)

1. Create a generator which always returns 10 random number between 1 to 2 and use this generator to create a list using comprehension. Below is the usage of random

>>> import random as r

>>> r.random()

0.1744863583896109

>>> r.uniform(1,2)

1.5645935380899947