1.Difference b/w python and shell scripting

* *Shell scripting has simpler notations for I/O redirection.*
* *It is simpler to create pipelines out of existing programs in shell.*
* *Shell scripting reuses entire programs.*
* *Shell is universally available (on anything like Unix) - Python is not necessarily installed.*

*In any case, these works can be done in python also, but it requires more complex functionalities and complex procedures in python.*

2.Difference b/w list and set

|  |  |
| --- | --- |
| ***List*** | ***Set*** |
| *Ordered collection* | *Unordered Collections* |
| *Allows you to do operations such as intersection, union, difference, and symmetric difference* | *are really variable-length arrays, not Lisp-style linked lists. In lists the elements are accessed by indexes.* |
| *In order to find an element in a set, a hash lookup is used* | *No hash lookup is required* |
|  |  |

3.what is \_\_module\_\_?

*Importing functions from one module to another is a common occurrence. The \_\_module\_\_ property is intended for retrieving the module where the function was defined, either to read the source code or sometimes to re-import it in a script.*

8.how to write multiple stmt on single line?

for i in range(10): print "i equals 9" if i==9 else "i equals 9"

11.what is recursion function? Example?

*Recursive Function:*

*Recursive function is a function which goes on until the terminating condition is satisfied*

*Example:*

def factorial(n):

if n == 0:

return 1

else:

return n \* factorial(n - 1)

12.what is command line arguments? (getopt module and sys module)

*Python provides a getopt module that helps you parse command-line options and arguments. The Python sys module provides access to any command-line arguments via the sys.argv. This serves two purposes −*

*sys.argv is the list of command-line arguments.*

*len(sys.argv) is the number of command-line arguments.*

import sys

print 'Number of arguments:', len(sys.argv), 'arguments.'

print 'Argument List:', str(sys.argv)

*Run the program:*

$ python test.py arg1 arg2 arg3

*Output:*

Number of arguments: 4 arguments.

Argument List: ['test.py', 'arg1', 'arg2', 'arg3']

16.What is \_\_init\_\_.py used for?

*The \_\_init\_\_.py files are required to make Python treat the directories as containing packages; this is done to prevent directories with a common name, such as string, from unintentionally hiding valid modules that occur later (deeper) on the module search path. In the simplest case, \_\_init\_\_.py can just be an empty file, but it can also execute initialization code for the package or set the \_\_all\_\_ variable, described later.*

20.how to remove whitespace in string?

*To remove leading and ending spaces, use* [str.strip()](http://docs.python.org/2/library/stdtypes.html" \l "str.strip):

sentence = ' hello apple'

sentence.strip()

>>> 'hello apple'

*To remove leading and ending spaces, use* [str.strip()](http://docs.python.org/2/library/stdtypes.html" \l "str.strip):

sentence = ' hello apple'

sentence.strip()

>>> 'hello apple'

*To remove all spaces, use* [str.replace()](http://docs.python.org/2/library/stdtypes.html" \l "str.replace):

sentence = ' hello apple'

sentence.replace(" ", "")

>>> 'helloapple'

*To remove duplicated spaces, use* [str.split()](http://docs.python.org/2/library/stdtypes.html" \l "str.split):

sentence = ' hello apple'

" ".join(sentence.split())

>>> 'hello apple'

33.what is pickle?

*The pickle module can take almost any python object and convert it to a string. This process is known as pickling. And the process of reconstructing the object from the string representation is called unpickling. Over the precess of pickling and unpickling, the string representation is stored either in a file or sent over a network connection to some other machine.*

29.Differnce b/w range and xrange?

*range() creates a list, so if you do range(1, 10000000) it creates a list in memory with 9999999elements.*

*xrange() is a sequence object and it does not produce list as in list, it yields the num as and when needed, thus making system faster.*

*Only in Python 2.7, not in Python 3*

35. differnce b/w two lists?

*Convert the lists to sets and run the usual*[*set operations*](http://docs.python.org/2/library/stdtypes.html#set)*such as difference or symmetric difference.*

*Example:*

a = [1,2,3,4,5,6]

b = [1,2,3,4,5,6,7,8,9]

set(b) - set(a)

*Output:*

set([7, 8, 9])

38.how to delete duplicate values in python in list?

*(by maintaining order)*

list1=[1,5,4,2,5,8,6,8,9,4,9]

list2=[]

for i in range(len(list1)):

if list1[i] not in list2:

list2.append(list1[i])

print list2

[1, 5, 4, 2, 8, 6, 9]

*(by not maintaining order)*

t = [1, 2, 3, 1, 2, 5, 6, 7, 8]

t

[1, 2, 3, 1, 2, 5, 6, 7, 8]

list(set(t))

[1, 2, 3, 5, 6, 7, 8]

s = [1, 2, 3]

list(set(t) - set(s))

[8, 5, 6, 7]

43.how to delete duplicate values in tuple?

*You cannot. Tuple is immutable*

47.what is lambda keyword?

*Python supports a style of programming called*functional programming*where we can pass functions to other functions to meet our requirements*.

44.how to find element in list?

list1= [‘a’, ‘b’, ‘c’, ‘e’, ‘g’, ‘y’, ‘r’]

*If we want to find ‘b’, we will do in the following way:*

list1.index(‘b’)

45.how to find element in tuple?

a= [(1,2),(1,4),(3,5),(5,7)]

*If we just want the first number to match we can do it like this:*

[item for item in a if item[0] == 1]

*If you are just searching for tuples with 1 in them:*

[item for item in a if 1 in item]

49.what is list compreshion?

*Python supports a concept called "list comprehensions". It can be used to construct lists in a very natural, easy way,*

S = [x\*\*2 for x in range(10)]  
>>> V = [2\*\*i for i in range(13)]  
>>> M = [x for x in S if x % 2 == 0]  
>>>   
>>> print S; print V; print M

|  |
| --- |
| [0, 1, 4, 9, 16, 25, 36, 49, 64, 81] [1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096] [0, 4, 16, 36, 64] |

50.what is mutable and immutable?

*The following are immutable objects:*

* *Numeric types: int, float, complex*
* *string*
* *tuple*
* *frozen set*
* *bytes*

*The following objects are mutable:*

* *list*
* *dict*
* *set*
* *byte array*

52.How to write sum of startin 1to 100?

sum = 0

for i in range(1,101):

sum += i

53.how to concatenate list and string?

*You can use this code,*

fruits = ['banana', 'apple', 'plum', 'pineapple', 'cherry']

mystr = 'i like the following fruits: '

print (mystr + ', '.join(fruits))

*The above code will return the output as below:*

i like the following fruits: banana, apple, plum, pineapple, cherry

54.how to split string?

## *Description*

*The method split() returns a list of all the words in the string, using str as the separator (splits on all whitespace if left unspecified), optionally limiting the number of splits to num.*

## *Syntax*

*Following is the syntax for split() method −*

str.split(str="", num=string.count(str)).

## *Parameters*

* *str -- This is any delimeter, by default it is space.*
* *num -- this is number of lines to be made*

## *Return Value*

*This method returns a list of lines.*

## *Example*

*The following example shows the usage of split() method.*

#!/usr/bin/python

str = "Line1-abcdef \nLine2-abc \nLine4-abcd";

print str.split( )

print str.split(' ', 1 )

*When we run above program, it produces following result −*

['Line1-abcdef', 'Line2-abc', 'Line4-abcd']

['Line1-abcdef', '\nLine2-abc \nLine4-abcd']

39.how to convert list to tuple?

l = [4,5,6]

tuple(l)

(4, 5, 6)

40.how to convert tuple to list?

tup = (1,2,3,4,5,6)

list(tup)

[1, 2, 3, 4, 5, 6]

41.how to convert tuple to dictionary?

t = ((1, 'a'),(2, 'b'))

dict((y, x) for x, y in t)

{'a': 1, 'b': 2}

46.how to count elements in list? len()

list1=[1,2,3,4,5,6,7,8]

len(list1)

8