

1. Strings in Python are immutable sequences containing only characters. A Python list is also a sequence collection, but a list can contain items of any type (e.g., character, strings, integers, floats, other lists, etc.). Lists are represented by comma-separated values enclosed in square brackets ('[', ')'). Operations on lists (**or any sequence collection**, e.g., strings) include:

Operation	Operator	Explanation	Example myList=[5,6,7,8] ListB=[8,9]	Result of Example
Indexing	[<index>]	Access the element specified by the index	myList[2]	7
Slicing	[:]	Extract a part of the list	myList[1:3]	[6, 7]
Concatenation	+	Combine lists together	myList + ListB	[5, 6, 7, 8, 8, 9]
Repetition	*	Concatenate a repeated number of times	ListB * 3	[8, 9, 8, 9, 8, 9]
Membership	in	Ask whether an item is in a list	3 in myList	False
Length	len(list)	How many items are in the list?	len(myList)	4

For the following lists, predict the results:

cheer = [2, 4, 6, 8, 'who', 'do', 'we', 'appreciate']

rhyme = [1, 2, 'buckle', 'your', 'shoe']

Expression	Predicted Result	Actual Result
cheer[4]		
cheer[2:6]		
rhyme[:4]		
cheer[1:4] + rhyme[-2:]		
cheer[:2] * 3		
6 in rhyme		
len(cheer)		
[cheer[2:4]*4]		

2. Lists in Python are mutable, i.e., they can be changed by assigning individual elements or slices new values.

For the following lists, predict the resulting lists:

Initial List Value	Expression	Result
temp = ['a', 'b', 'c', 'd']	temp[1] = 99	
temp = ['a', 'b', 'c', 'd']	temp[1] = 'cat'	
temp = ['a', 'b', 'c', 'd']	temp[1] = ['cat', 'dog']	
temp = ['a', 'b', 'c', 'd']	temp[1:3]=[6, 7, 8, 9]	
temp = ['a', 'b', 'c', 'd']	temp[1:2] = 5	

3. The following methods are provided by Lists in Python:

Method	Usage	Explanation
append	<code>myList.append(item)</code>	Adds item to the end of myList
extend	<code>myList.extend(otherList)</code>	Extends myList by adding all items in otherList to myList's end
insert	<code>myList.insert(i, item)</code>	Insert item in myList at index i
pop	<code>myList.pop()</code>	Remove and return the last item in myList
pop(i)	<code>myList.pop(i)</code>	Remove and return the ith item in myList
del	<code>del myList[i]</code>	Deletes the item in the ith position of myList
remove	<code>myList.remove(item)</code>	Removes the first occurrence of item in myList
index	<code>myList.index(item)</code>	Returns the index of the first occurrence of item in myList
count	<code>myList.count(item)</code>	Returns the number of occurrences of item in myList
sort	<code>myList.sort()</code>	Modifies myList to be sorted
reverse	<code>myList.reverse()</code>	Modifies myList to be in reverse order

What would be the value of myList and temp after each of the following Python statements?

```
myList = range(5)
```

```
temp = range(10, 7, -1)
```

```
last = myList.pop( )
```

```
myList.extend(temp)
```

```
myList.insert(4, 3)
```

```
item = myList.pop(5)
```

```
del myList[2]
```

4. Below is a summary of the important file-system functions from the `os` module in Python.

os Module File-system Functions	
General syntax	Description
<code>getcwd()</code>	Returns the complete path of the current working directory
<code>chdir(path)</code>	Changes the current working directory to path
<code>listdir(path)</code>	Returns a list of the names in directory named path
<code>mkdir(path)</code>	Creates a new directory named path and places it in the current working directory
<code>rmdir(path)</code>	Removes the directory named path from the current working directory
<code>remove(path)</code>	Removes the file named path from the current working directory
<code>rename(old, new)</code>	Renames the file or directory named old to new

os.path Module File-system Functions	
General syntax	Description
<code>exists(path)</code>	Returns True if path exists and False otherwise
<code>isdir(path)</code>	Returns True if path is a directory and False otherwise
<code>isfile(path)</code>	Returns True if path is a file and False otherwise
<code>getsize(path)</code>	Returns the size in bytes of the object named path

Write a program to display the current working directory's content. Clearly state whether the entry is a file or directory. For files, display their size.

a) Write a high-level English-like algorithm for the main function.

b) Write the main function?

5. Suppose you have a small business and are maintaining customer information in a text file: `customerData.txt`. Each customer record is on a single line with 12 fields separated by commas(','). The order of the fields on a line is: First Name, Middle Initial, Last Name, Street Address, City, State, Zip Code, Country, Email Address, Telephone Number, Gender, and Birthday.

Write a function `generateList` to read each line of text about a customer and generate a list of fields for that customer. Insert the list of fields for each customer into a `customers` list. Thus, the `customers` list is a list-of-lists where each item contains the information about a single customer. The general approach you should take is to:

- read each line from the file (contains all the information about a customer) as a string,
- split the line into a list of strings for the fields,
- insert this list of strings into the `customers` list.

```
def main():
    """ Open's file, reads customer information into a list, closes the file"""
    custFile = open('customerData.txt', 'r')
    customerList = generateList(custFile)

    # Echo first customer in the customerList
    print "customerList[0]:", customerList[0]
    custFile.close()

def generateList(custFile):
    """ Reads customer data from custFile and returns a list of customers"""
```