**LIST : Collection of Strings, integer, floats, Sub list**

**>>> print(' Hello ')**

**Hello**

**>>> a = 24**

**>>> b = 3.5**

**>>> type(a)**

**<class 'int'>**

**>>> type(b)**

**<class 'float'>**

**>>> st = 'python program '**

**>>> type(st)**

**<class 'str'>**

**>>>**

**>>> ls =['raju', 'rani', 'siva', '123', 'hello']**

**>>> ls**

**['raju', 'rani', 'siva', '123', 'hello']**

**>>>**

**>>>**

**>>> ls.append('python')**

**>>> ls**

**['raju', 'rani', 'siva', '123', 'hello', 'python']**

**>>>**

**>>>**

**>>>**

**>>> ls.insert(2, 'ramu')**

**>>> ls**

**['raju', 'rani', 'ramu', 'siva', '123', 'hello', 'python']**

**>>>**

**>>>**

**>>> ls.remove('ramu')**

**>>> ls**

**['raju', 'rani', 'siva', '123', 'hello', 'python']**

**>>>**

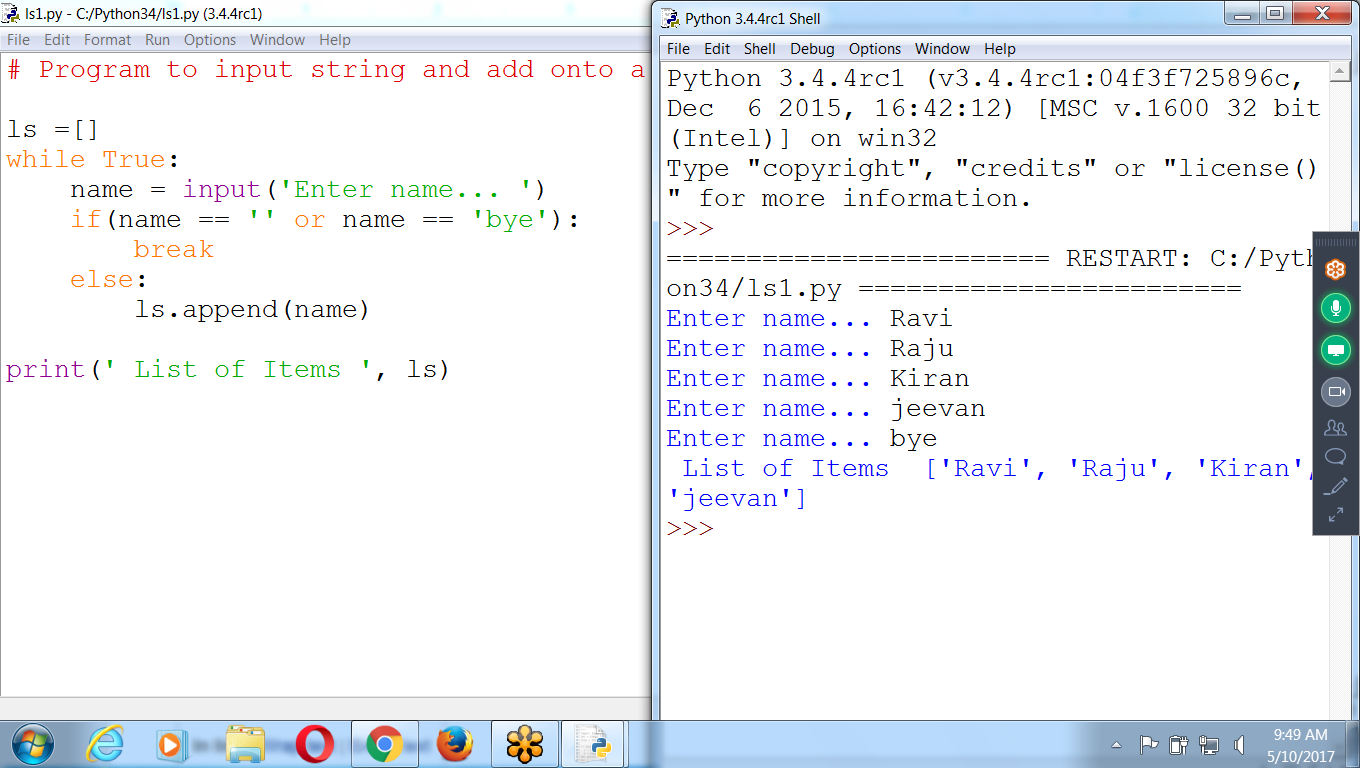
**>>> ls.index('hello')**

**4**

**Program :: Accept Name Until User press Enter Key and**

**append to list**

append() : To add element at the end of the list



**# Accept name and store onto a list variable until user press enter key**

# Program to input string and add onto a list

ls =[]

while True:

name = input('Enter name... ')

if(name == '' or name == 'bye'):

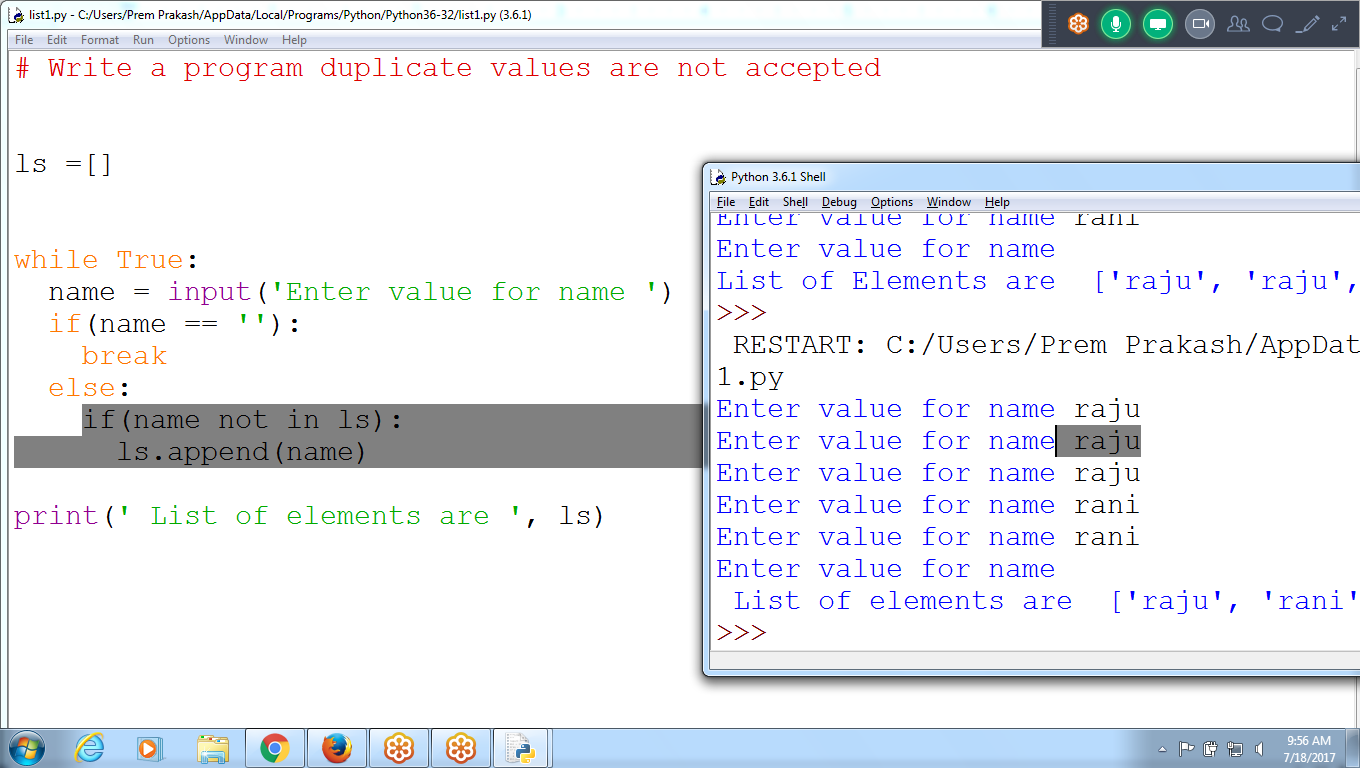
break

else:

ls.append(name)

print(' List of Items ', ls)

Remove Duplicates ::: Using “not in” Operator



# Input Name until user press Enter Key

ls1 = []

while True:

name = input('Enter Any name ')

if(name == '' or name == 'bye'):

break

else:

ls1.append(name)

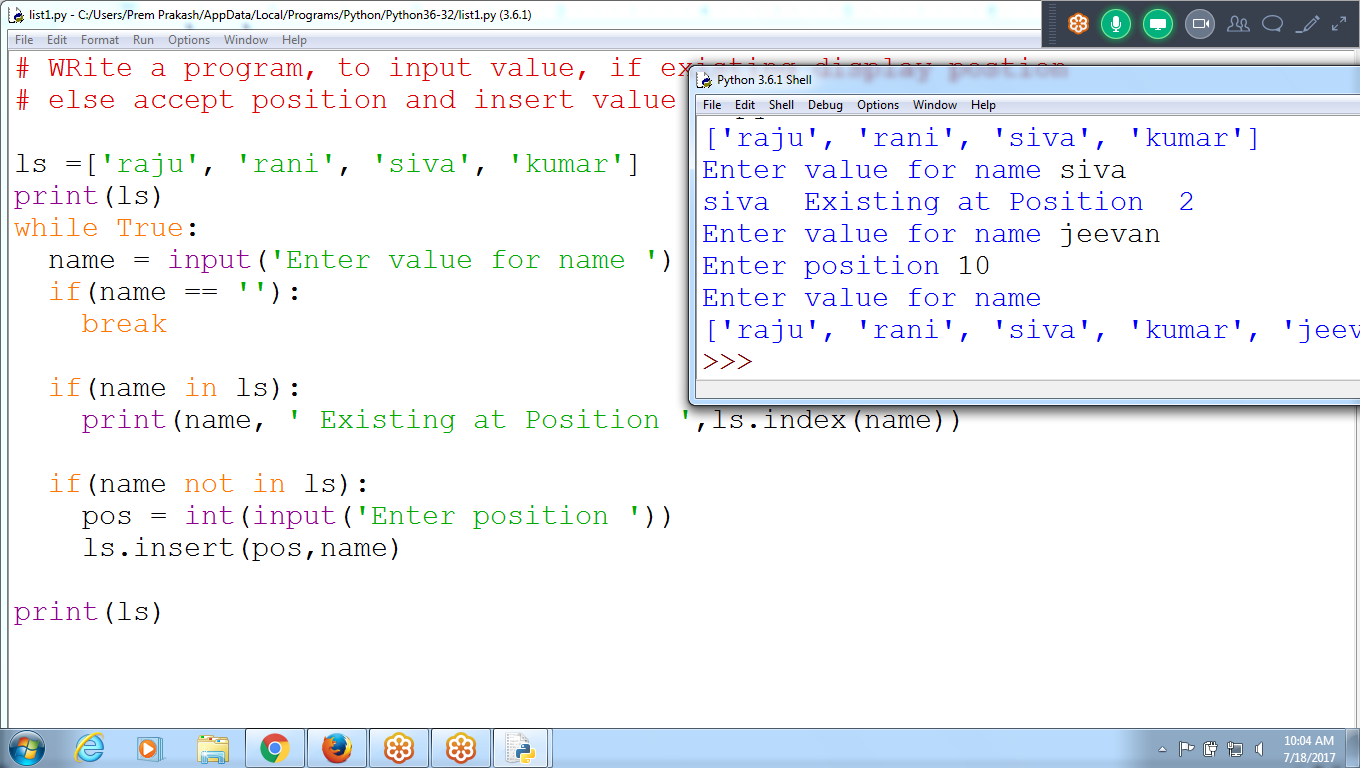
print(ls1)

**Example : INput name, if name not existing, insert at given position**

**If existing display Position**

If position not existing inserts at last i.e append

.index() :: returns first position found in the list



**# input name, if name not existing insert at specified position**

ls = ['ramu', 'ravi', 'raja', 'kiran']

while True:

name = input('Enter Name ')

if(name == ''):

break

if name in ls:

print(name, ' Existing at position ', ls.**index**(name))

if name not in ls:

pos = int(input('Enter Position to Insert value '))

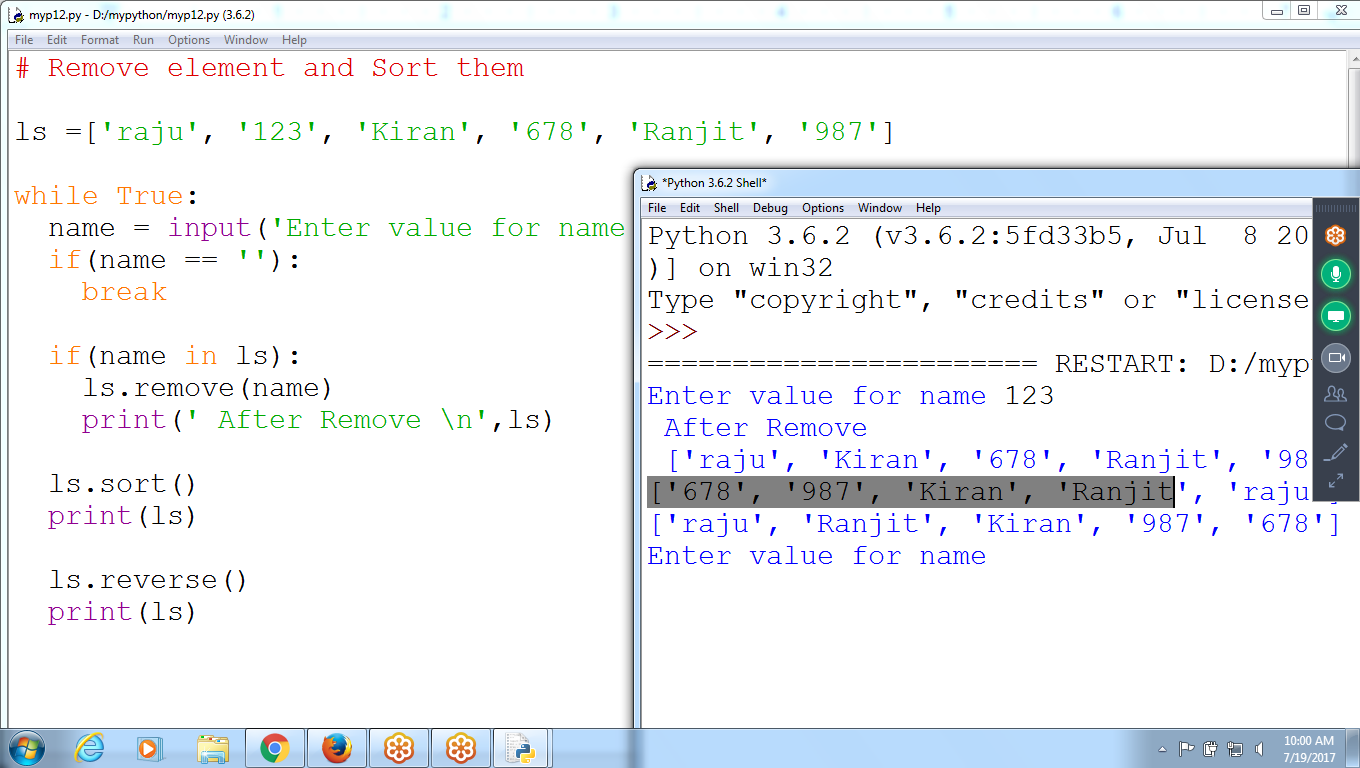
ls.insert(pos,name)

print(ls)

**Example : To Delete and Sort elements in LIST**

ls.sort() : Ascending Order i.e increasing order

ls.reverse() : Descending Order



# Remove element and Sort them

ls =['raju', '123', 'Kiran', '678', 'Ranjit', '987']

while True:

name = input('Enter value for name ')

if(name == ''):

break

if(name in ls):

ls.remove(name)

print(' After Remove \n',ls)

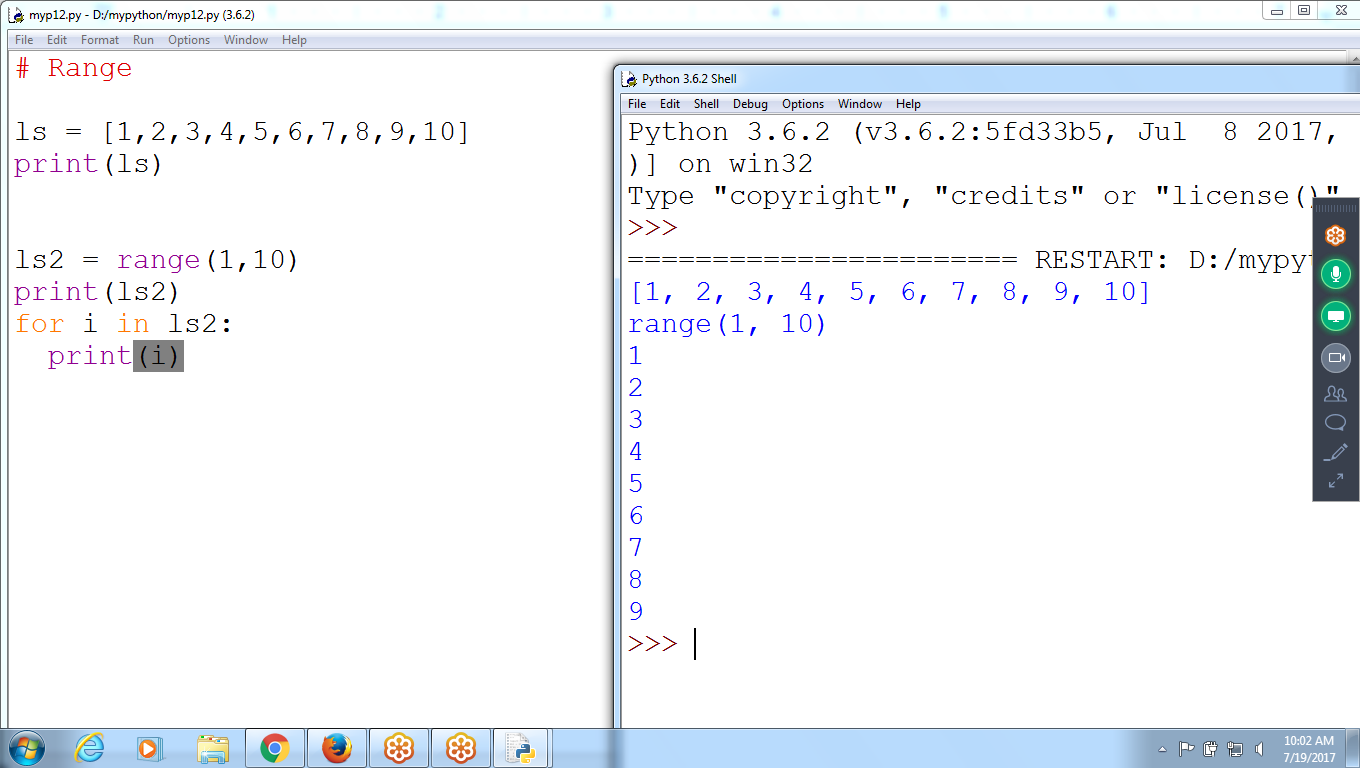
ls.sort()

print(ls)

ls.reverse()

print(ls)

Using Range()



# Range

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

print(ls)

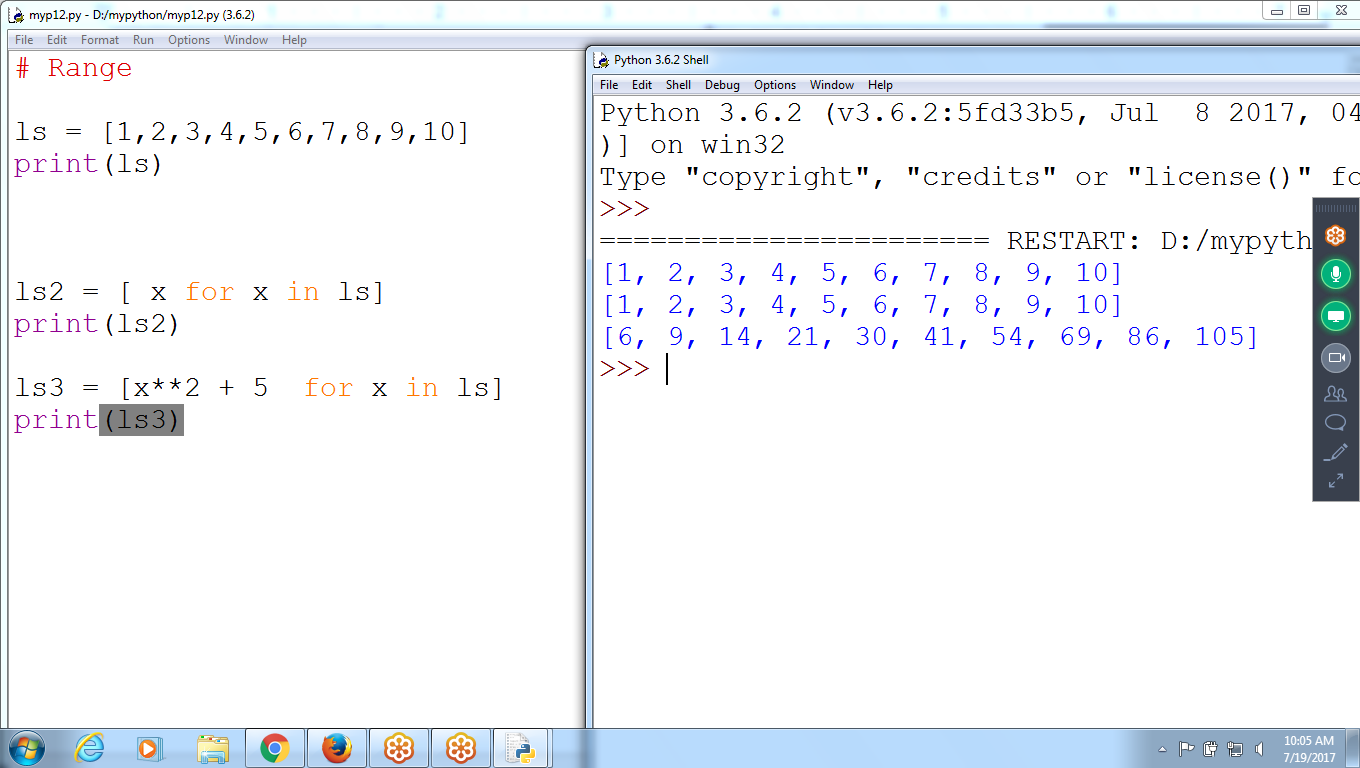
ls2 = range(1,10)

print(ls2)

for i in ls2:

print(i)

List Example



# Range

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

print(ls)

ls2 = [ x for x in ls]

print(ls2)

ls3 = [x\*\*2 + 5 for x in ls]

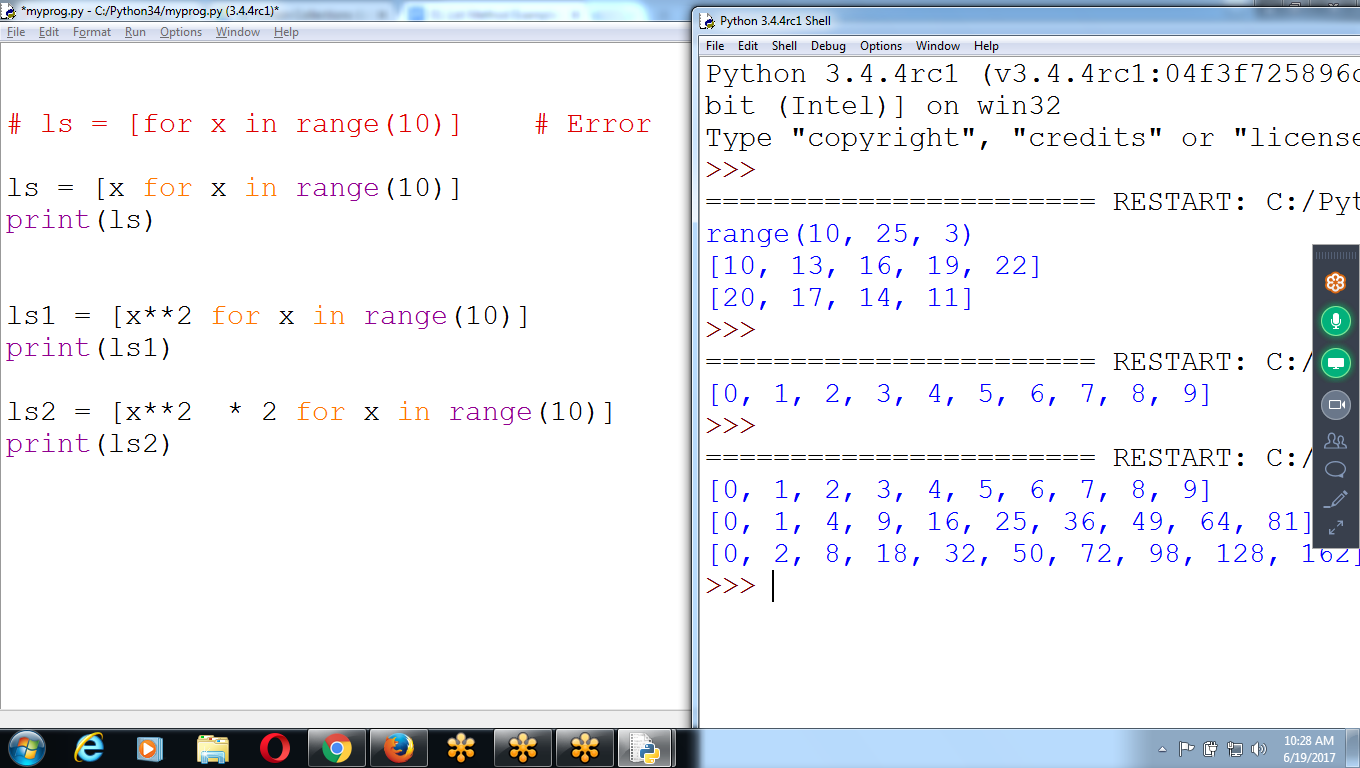
print(ls3)

**>>> ls = [x\*\*2 for x in range(10)] # Exponent starts from 0 to 9**

>>> ls

[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]

>>>



# ls = [for x in range(10)] # Error

ls = [x for x in range(10)]

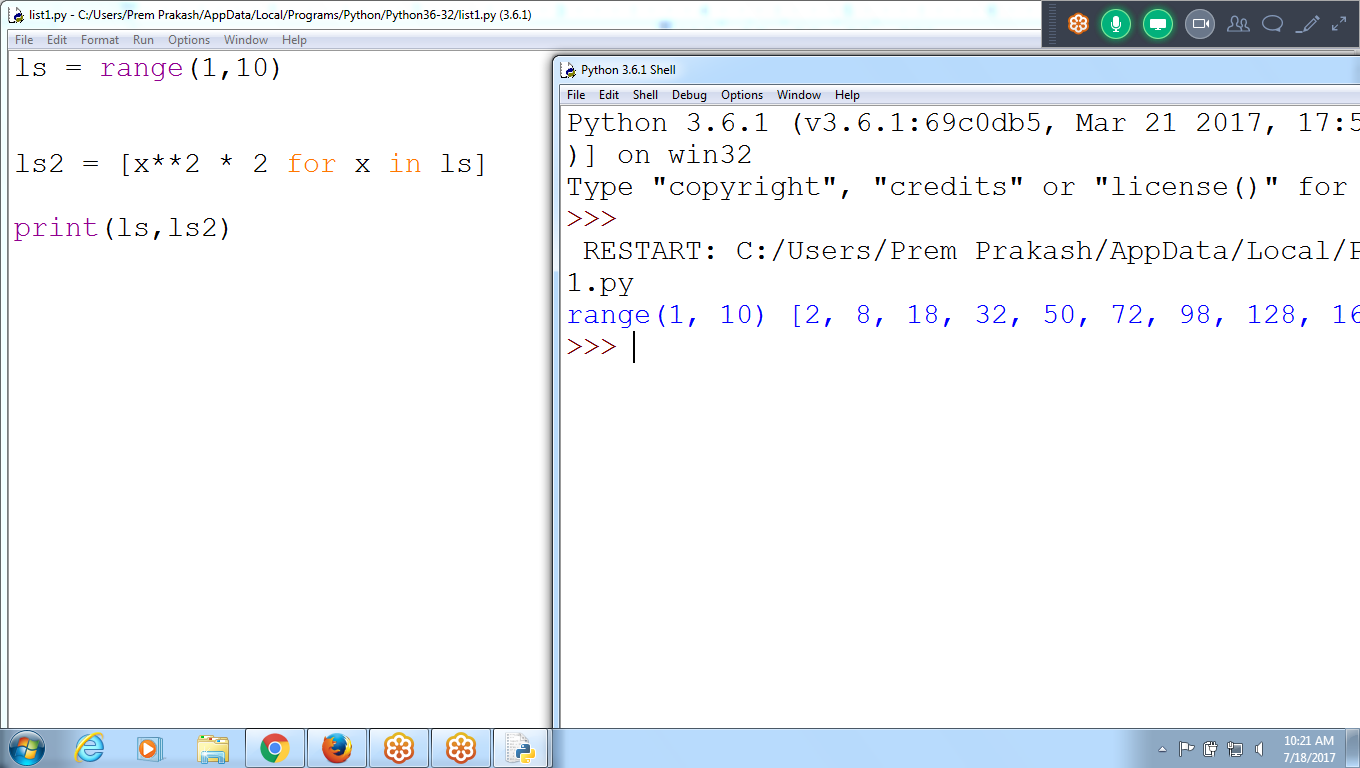
print(ls)

ls1 = [x\*\*2 for x in range(10)]

print(ls1)

ls2 = [x\*\*2 \* 2 for x in range(10)]

print(ls2)

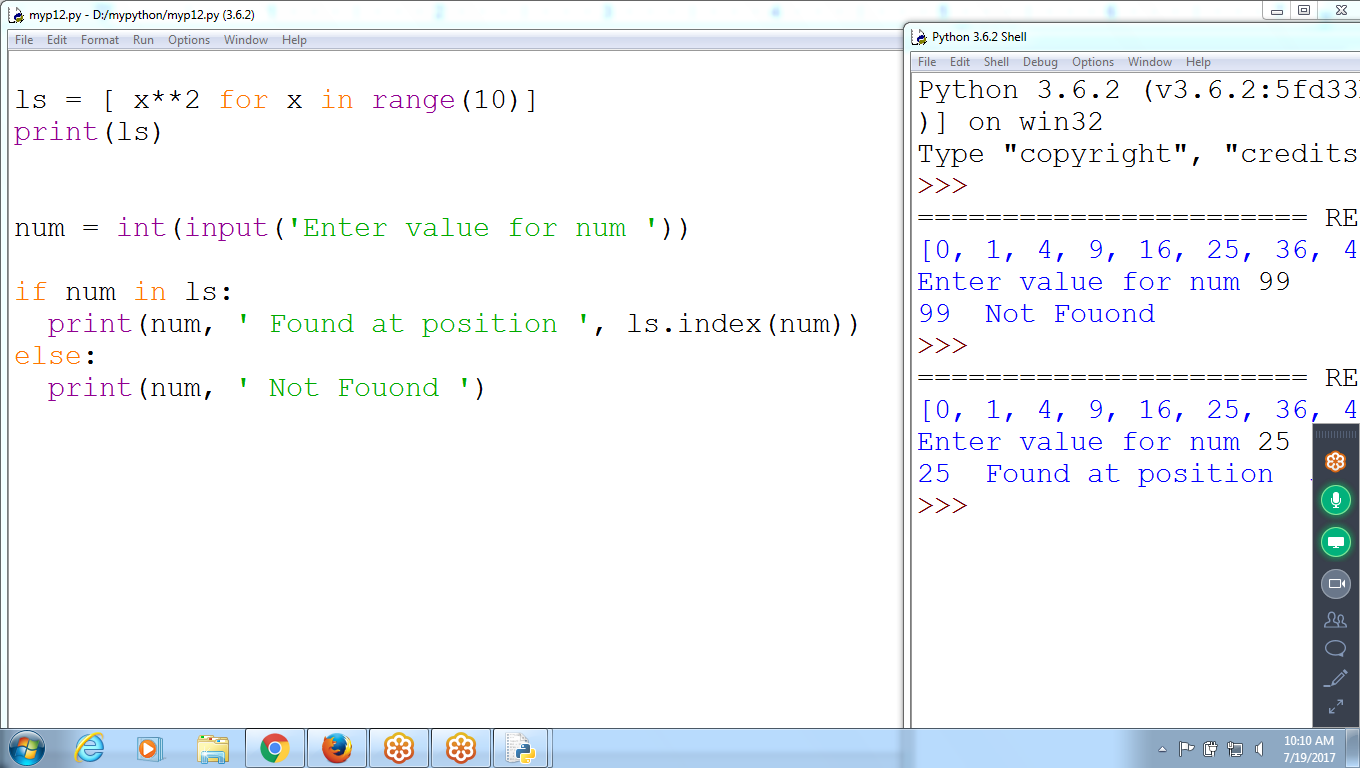


ls = range(1,10)

ls2 = [x\*\*2 \* 2 for x in ls]

print(ls,ls2)

Search for a Key in a List



**# Accept integer key and search it is existing or not**

ls = [ x\*\*2 for x in range(10)]

print(ls)

num = int(input('Enter value for num '))

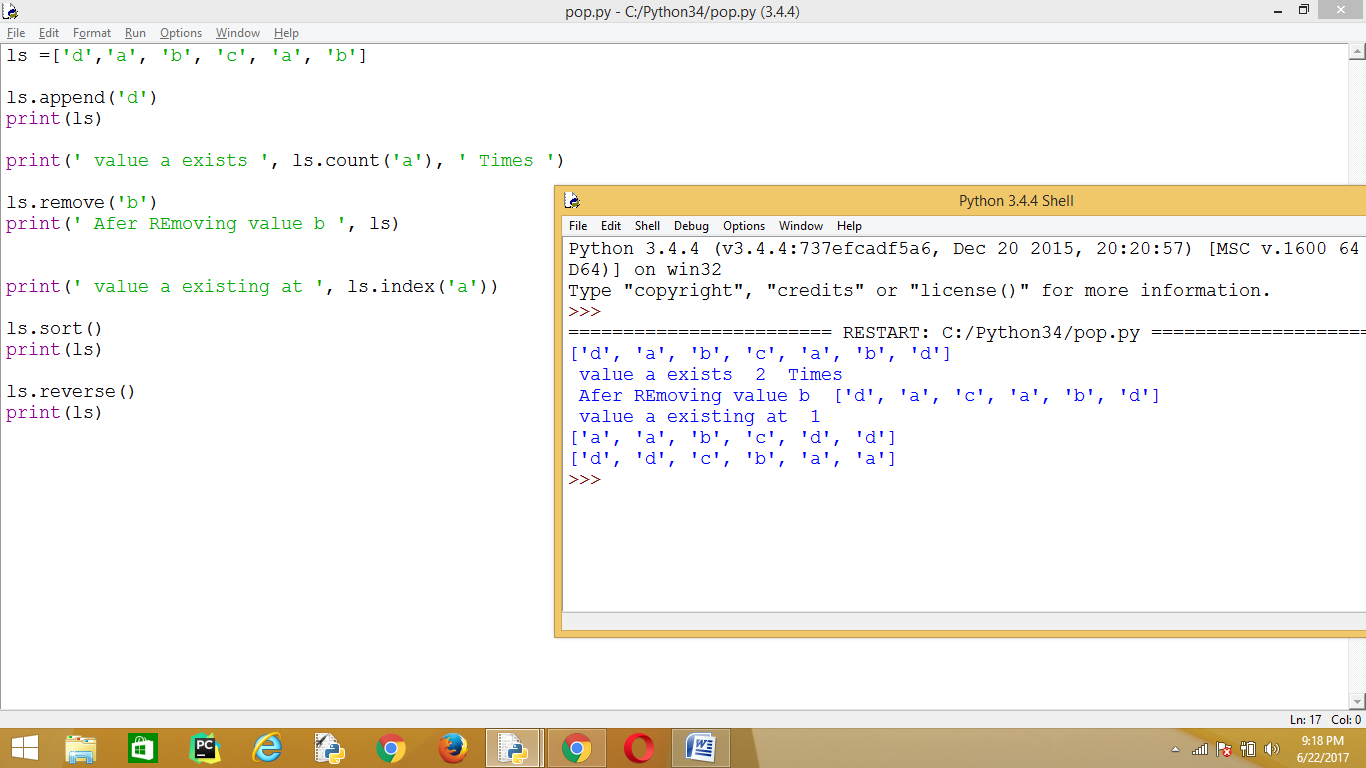
if num in ls:

print(num, ' Found at position ', ls.index(num))

else:

print(num, ' Not Fouond ')

**List Methods**



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

ls.append('d')

print(ls)

print(' value a exists ', ls.count('a'), ' Times ')

ls.remove('b')

print(' Afer REmoving value b ', ls)

print(' value a existing at ', ls.index('a'))

ls.sort()

print(ls)

ls.reverse()

print(ls)

**Python List Methods**

append() - Add an element to the end of the list

extend() - Add all elements of a list to the another list

insert() - Insert an item at the defined index

remove() - Removes an item from the list

pop() - Removes and returns an element at the given index

clear() - Removes all items from the list

index() - Returns the index of the first matched item

count() - Returns the count of number of items passed as an argument

sort() - Sort items in a list in ascending order

reverse() - Reverse the order of items in the list

copy() - Returns a shallow copy of the list

**Built-in Functions with List Function Description**

all() Return True if all elements of the list are true (or if the list is empty).

any() Return True if any element of the list is true. If the list is empty, return False.

enumerate() Return an enumerate object.

It contains the index and value of all the items of list as a tuple.

len() Return the length (the number of items) in the list.

list() Convert an iterable (tuple, string, set, dictionary) to a list.

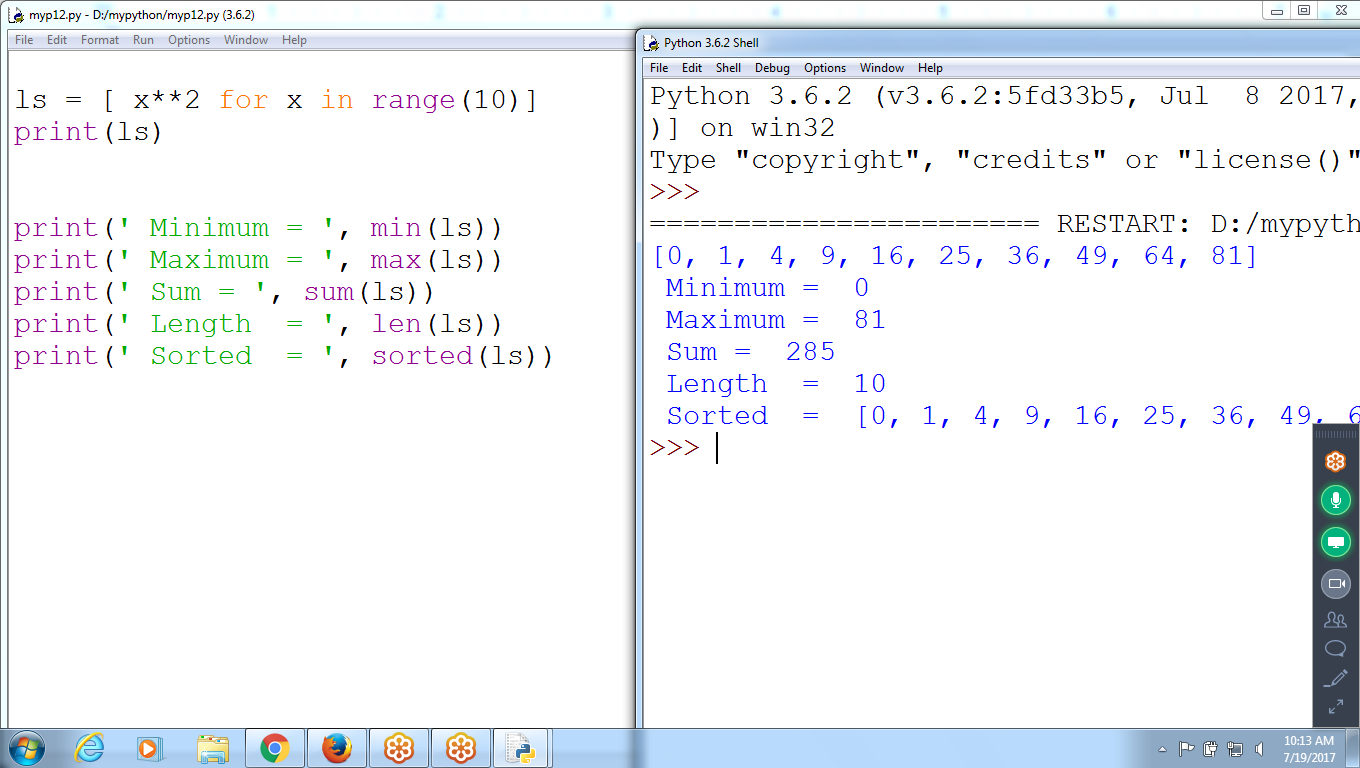
max() Return the largest item in the list.

min() Return the smallest item in the list

sorted() Return a new sorted list (does not sort the list itself).

sum() Return the sum of all elements in the list.

**Python List functions**

****

**ls = [ x\*\*2 for x in range(10)]**

**print(ls)**

**print(' Minimum = ', min(ls))**

**print(' Maximum = ', max(ls))**

**print(' Sum = ', sum(ls))**

**print(' Length = ', len(ls))**

**print(' Sorted = ', sorted(ls))**

**Append takes as ITEM in Sub list format**

>>> ls = ['raj', 'ravi', 'kiran', 'john', 'ranjit']

>>> ls

['raj', 'ravi', 'kiran', 'john', 'ranjit']

>>> len(ls)

5

>>> ls.append(['a','b','c'])

>>> ls

['raj', 'ravi', 'kiran', 'john', 'ranjit', **['a', 'b', 'c']]**

>>> len(ls)

6

>>>

**EXTEND Each Element as Separate Item as Concatenation**

>>> ls2 = ['raj', 'ravi', 'kiran', 'john', 'ranjit']

>>> ls2.extend(**['a','b','c']**)

>>> ls2

['raj', 'ravi', 'kiran', 'john', 'ranjit', **'a', 'b', 'c']**

>>>

>>> len(ls2)

8

**# Difference Between Append, Extend, Insert**

**>>> ls = ['raju', 'ran', 'siva']**

**>>> len(ls)**

**3**

**>>> ls.append(['a','b','c'])**

**>>> len(ls)**

**4**

**>>>**

**>>>**

**>>>**

**>>>**

**>>> ls = ['raju', 'ran', 'siva']**

**>>> len(ls)**

**3**

**>>> ls.extend(['a','b','c'])**

**>>> len(ls)**

**6**

**>>> ls**

**['raju', 'ran', 'siva', 'a', 'b', 'c']**

# **References**

* Assigning a list to a variable, means assigning a list *reference* to the variable
* A **reference** is a value that points to some bit of data,
* A **list reference** is a value that points to a list

**MUTABLE (or) Modify (or) References (or) Ex: List, Dictionary**

**Immutable (or) can NOT Modify (or) value type ex: Integer, string, Tuples**

* Python uses references whenever variables must store values of **mutable data** types, such as **lists or dictionaries**
* For values of immutable data types such as strings, integers, or tuples, Python variables will store the **value** itself.
* **variables** are like boxes that contain values.
* strings and integer values, variables simply contain the string or integer value
* **list variables** contain *references* to list values
* Copies only the list reference in name1 to name2, not the list value.
* The values stored in name1 and name2 now both refer to the same list.
* modify the first element of name2 , modifies the same list that name1 refers to

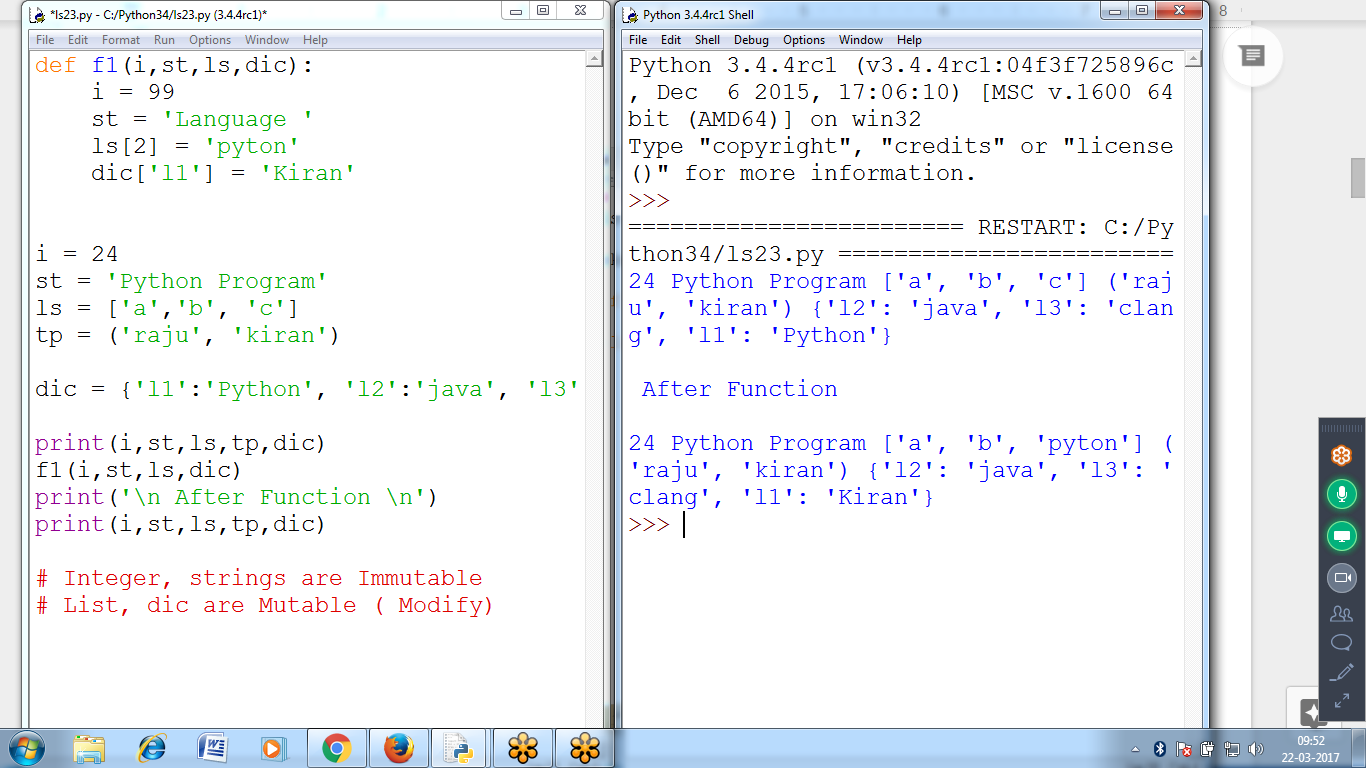
## **Working with References**

how arguments get passed to functions ::

When a function is called, the values of the arguments are copied to the parameter variables

**Integers and strings can’t Modify ::: Immutable**

**List and Dictionary :: Mutable :: Can modify**



#Diff bet String, list, tuple and Dict

# integer, string and Tuples are Immutable can't be Modify,

# list and Dictionary can be MODIFIED are MUTABLE object

def f1(i,st,ls,dic):

i = 99

st = 'Language '

ls[2] = 'pyton'

dic['l1'] = 'Kiran'

i = 24

st = 'Python Program'

ls = ['a','b', 'c']

tp = ('raju', 'kiran')

dic = {'l1':'Python', 'l2':'java', 'l3':'clang'}

print(i,st,ls,tp,dic)

**f1(i,st,ls,dic)**

print('\n After Function \n')

print(i,st,ls,tp,dic)

# Integer, strings are Immutable

# List, dic are Mutable ( Modify)

**Strings are Immutable (can NOT Modify)**

Lists are Mutable (can Modify values)

>>> st = 'python lang'

>>>

>>>

>>> **st[2] = 'r'**

Traceback (most recent call last):

File "<pyshell#36>", line 1, in <module>

st[2] = 'r'

TypeError: 'str' **object does not support item assignment**

>>>

>>>

>>>

>>>

>>> ls = ['raju', 'ran', 'siva', 'a', 'b', 'c']

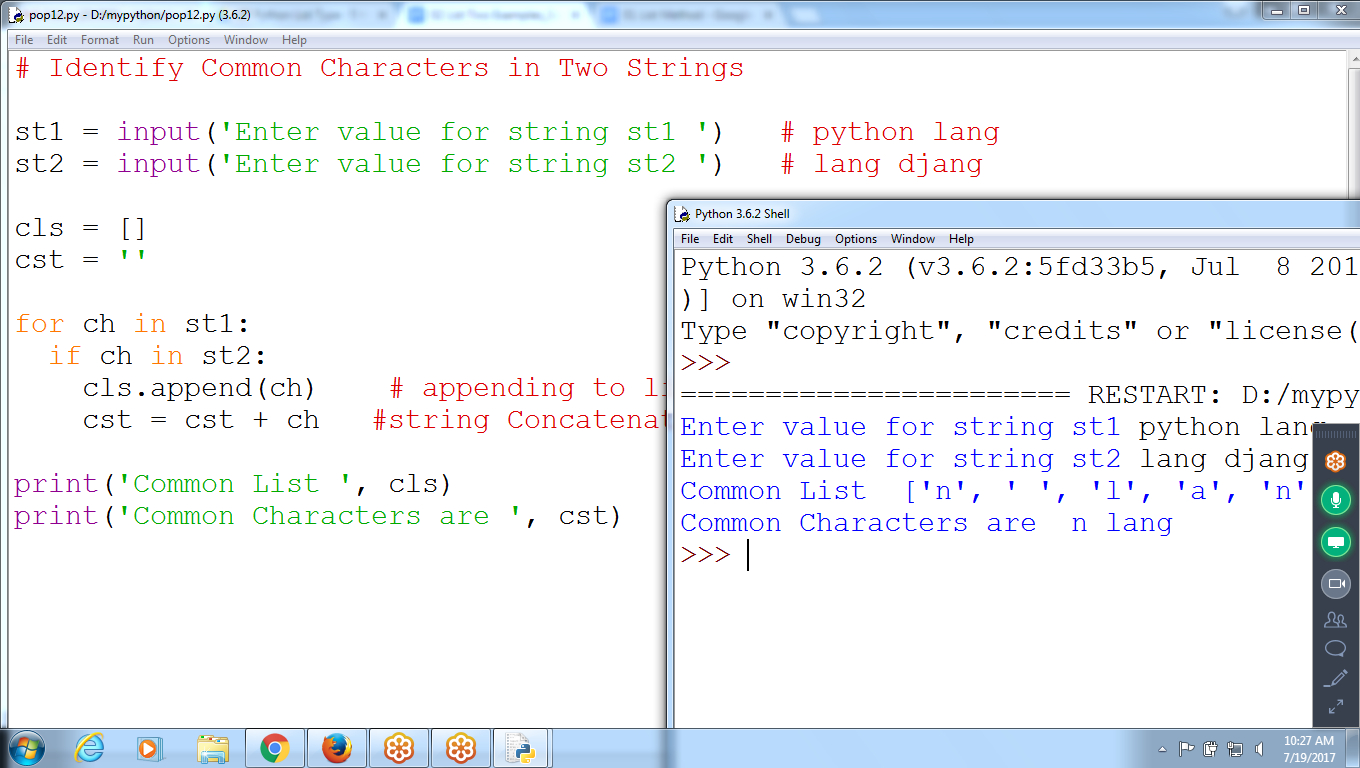
>>> ls[2] = 'ranjeet'

>>> ls

['raju', 'ran', 'ranjeet', 'a', 'b', 'c']

>>>

Common Characters in a List



# Identify Common Characters in Two Strings

st1 = input('Enter value for string st1 ') # python lang

st2 = input('Enter value for string st2 ') # lang djang

cls = []

cst = ''

for ch in st1:

if ch in st2:

cls.append(ch) # appending to list

cst = cst + ch #string Concatenation

print('Common List ', cls)

print('Common Characters are ', cst)