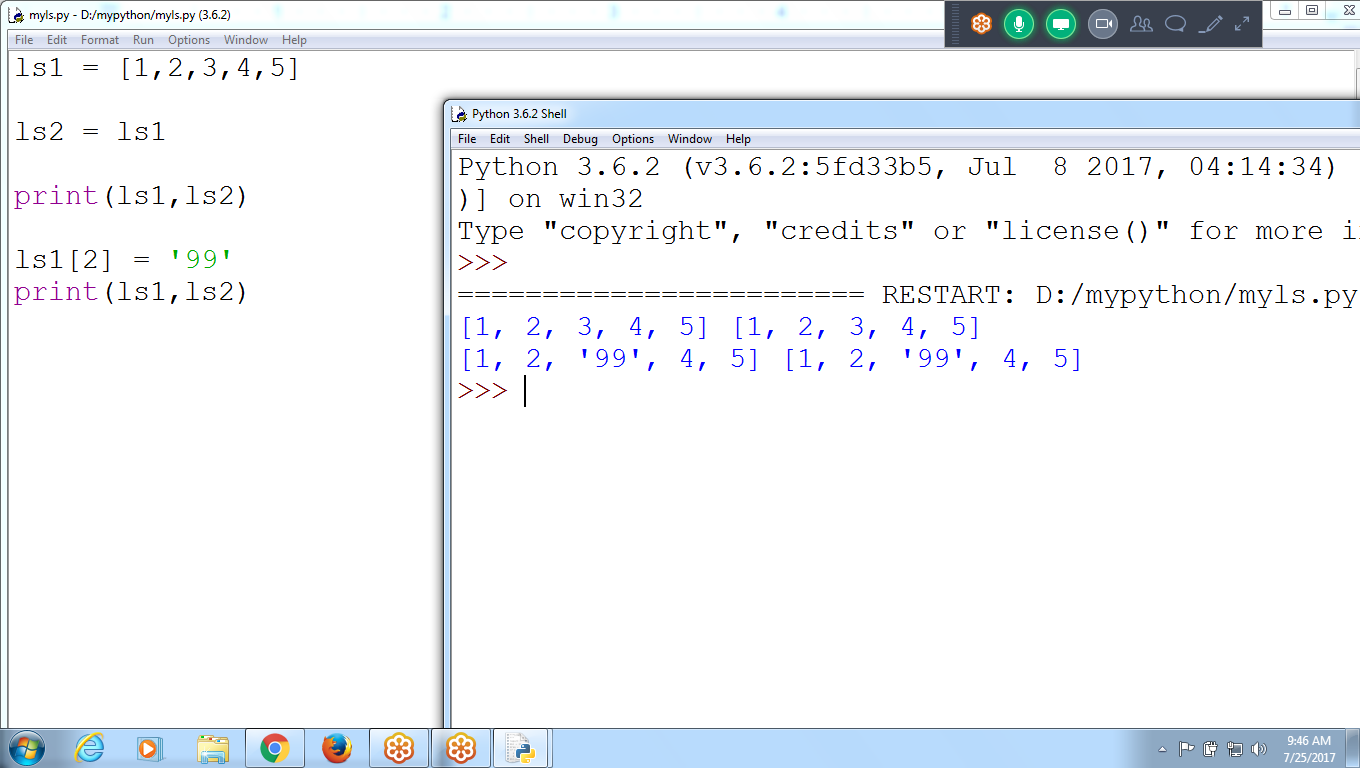
Ls1 address (reference) copied to ls2

On doing change in ls1, it effets ls2 structure



ls1 = [1,2,3,4,5]

ls2 = ls1

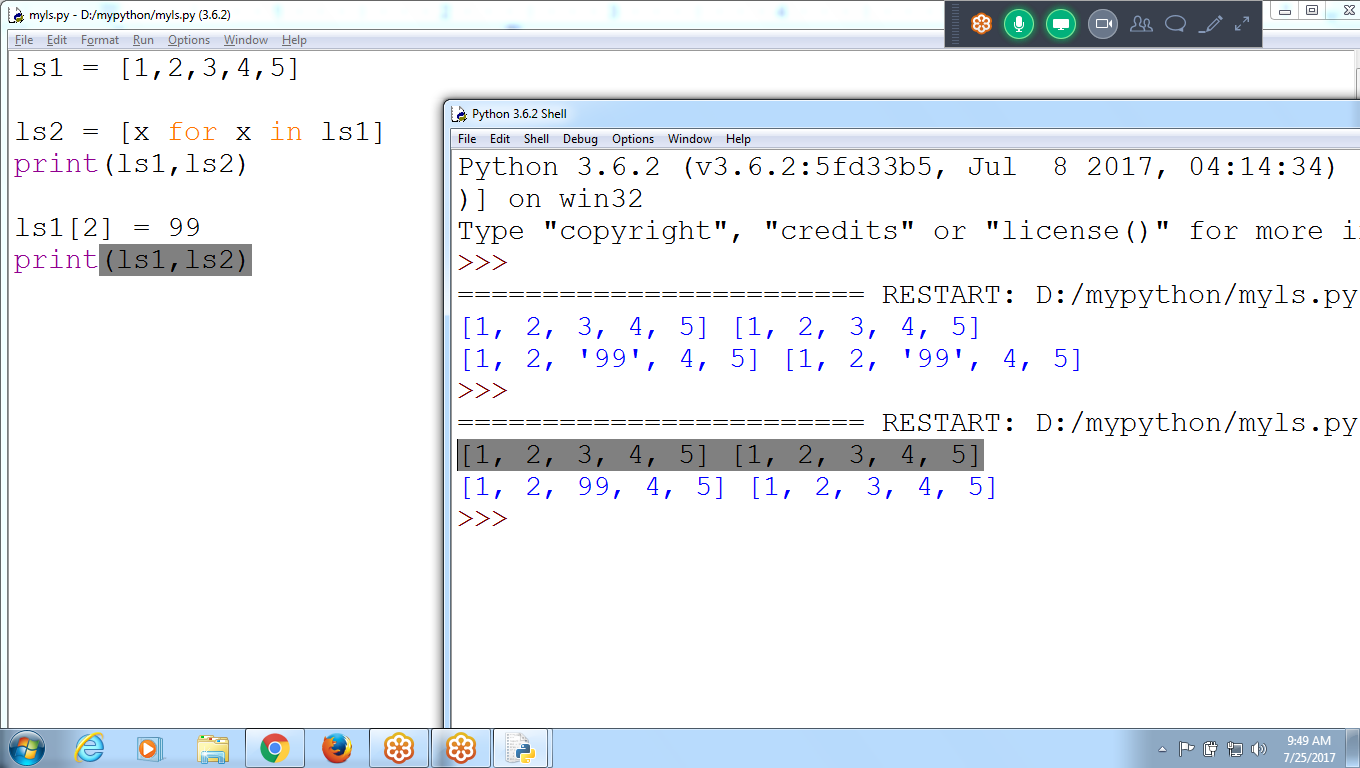
print(ls1,ls2)

ls1[2] = '99'

print(ls1,ls2)

Copying values from ls1 to ls2

On doing changes in ls1, it is not effecting in ls2



ls1 = [1,2,3,4,5]

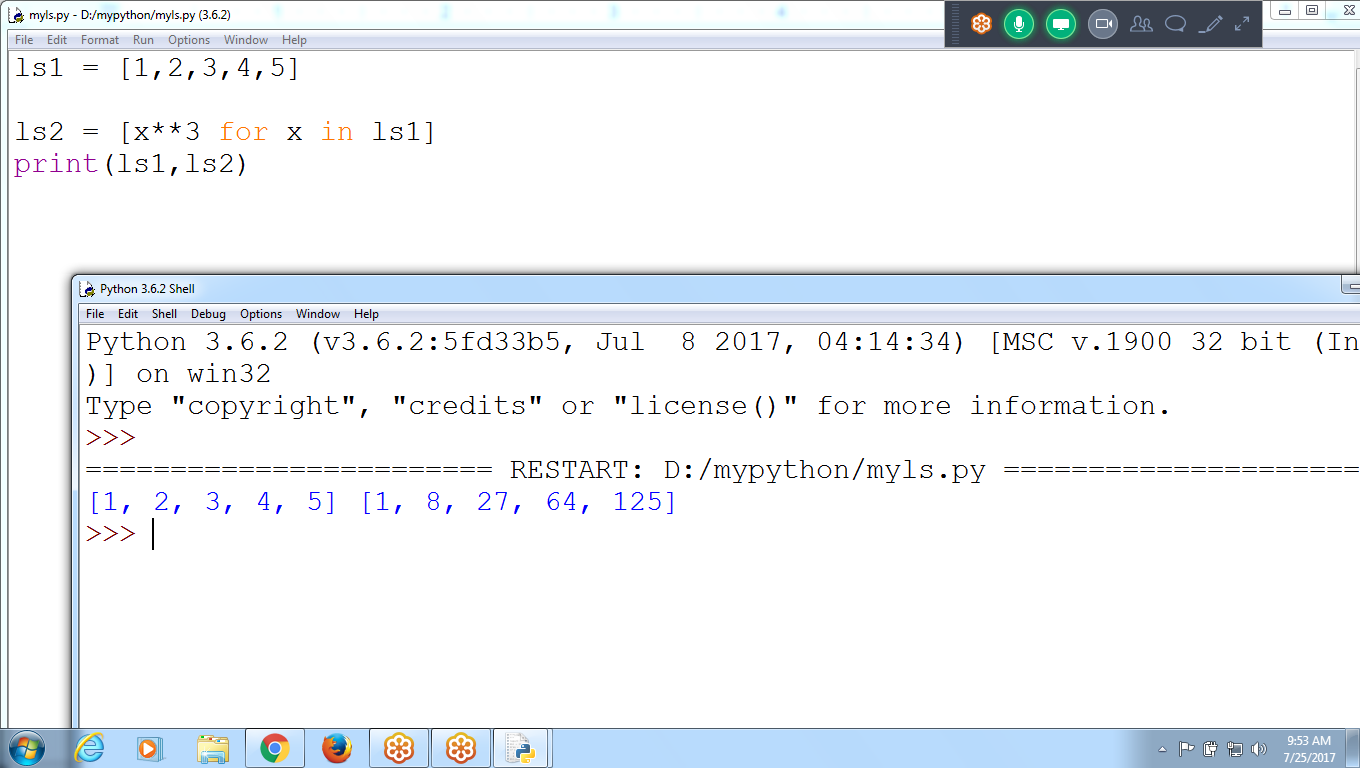
ls2 = [x for x in ls1]

print(ls1,ls2)

ls1[2] = 99

print(ls1,ls2)

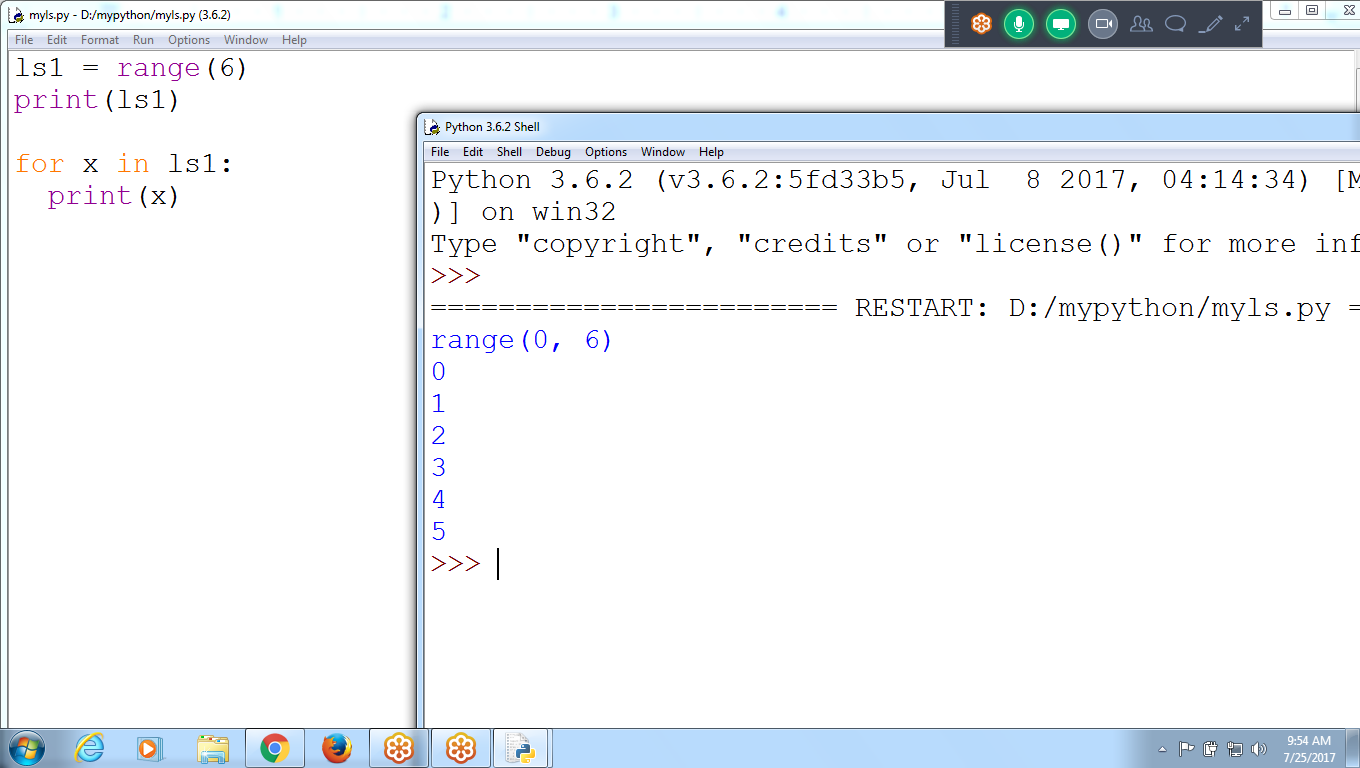
**List Comprehension :**

****

**ls1 = [1,2,3,4,5]**

**ls2 = [x\*\*3 for x in ls1]**

**print(ls1,ls2)**

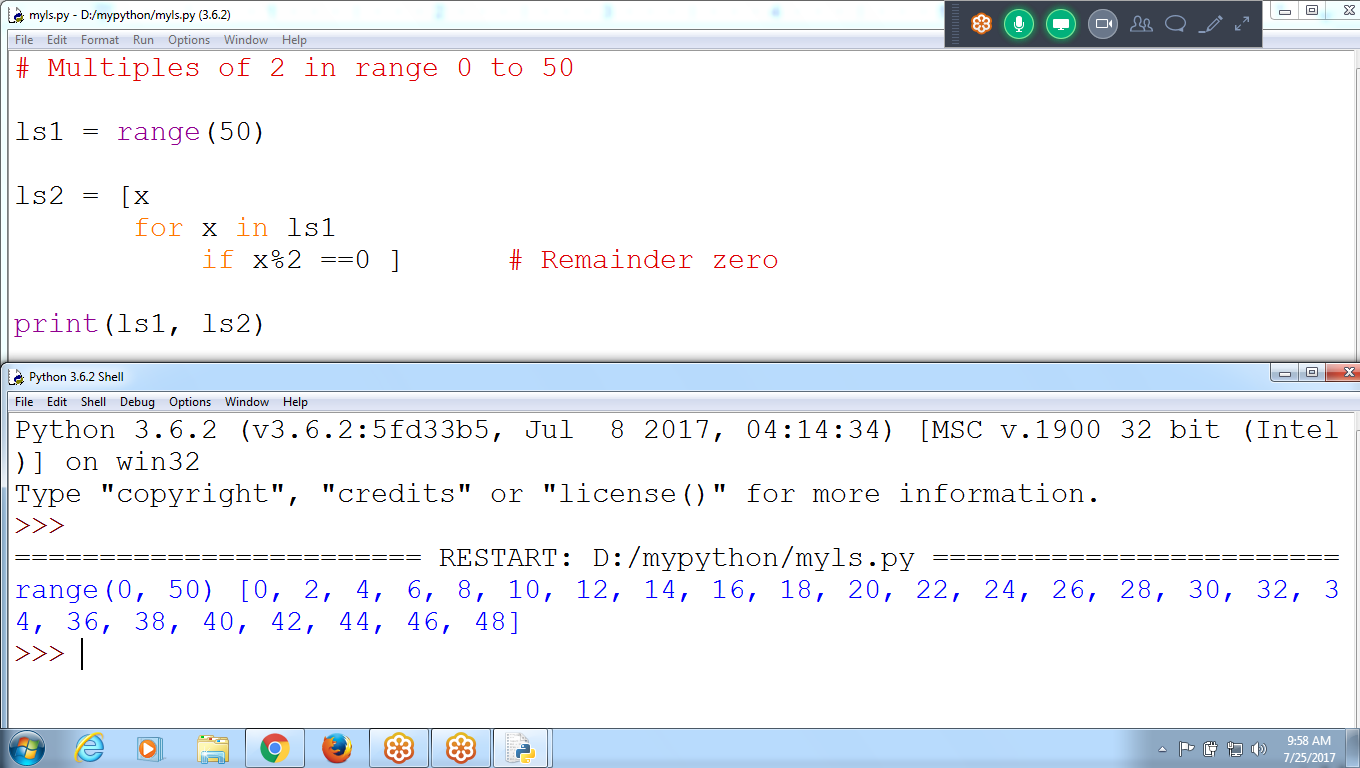
****

**ls1 = range(6)**

**print(ls1)**

**for x in ls1:**

**print(x)**

****

**# Multiples of 2 in range 0 to 50**

**ls1 = range(50)**

**ls2 = [x**

**for x in ls1**

**if x%2 ==0 ] # Remainder zero**

**print(ls1, ls2)**

**>>> ls = [x\*\*2 for x in range(20) ] # power of 2 from 0 to 19**

**>>> lt = [x**

**for x in ls**

**if x % 2 == 0]**

**>>> print(lt)**

**[0, 4, 16, 36, 64, 100, 144, 196, 256, 324]**

**>>>**

1 to 100

Find list of prime numbers in n= 100 to 200

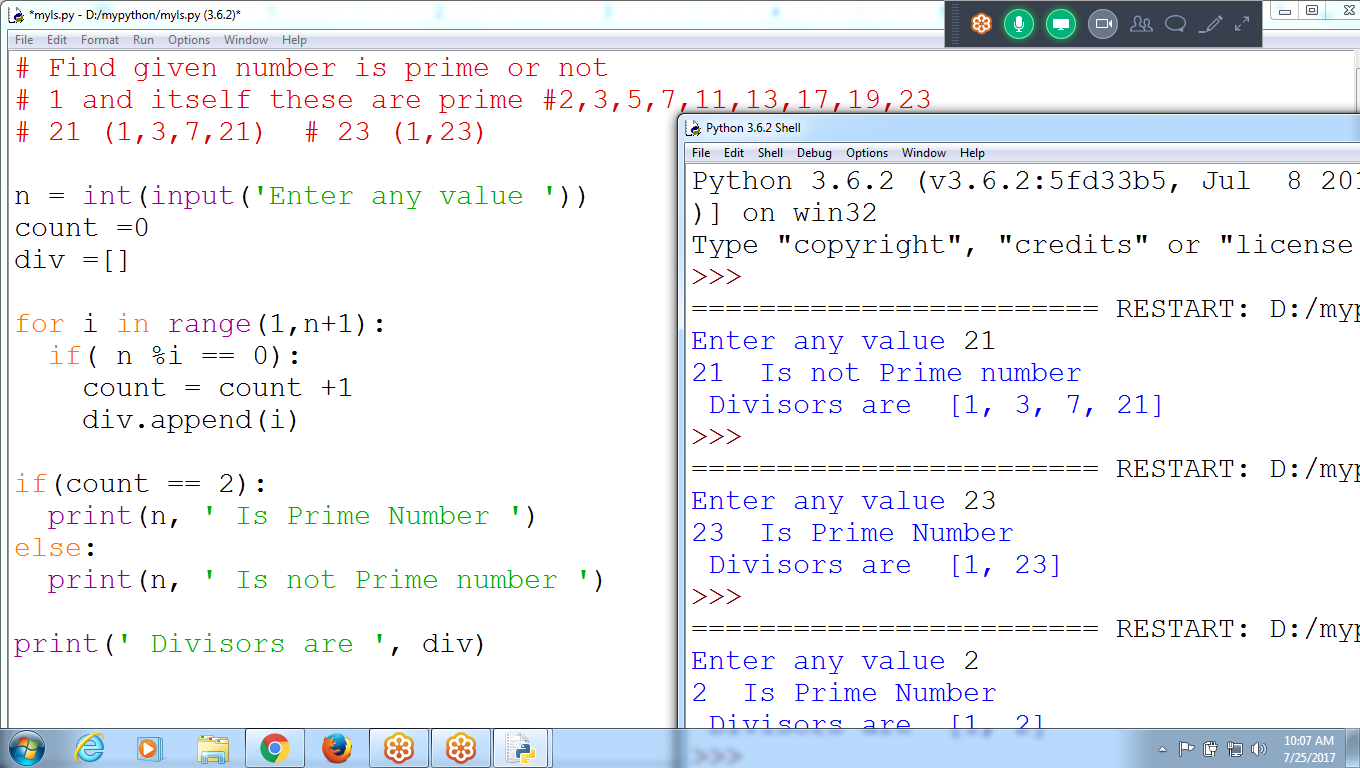
N =100, 101, 102,

N value must divide with all numbers from 1 to 100

If count = 2

Then n is prime

**Find Given Number is Prime or not**



# Find given number is prime or not

# 1 and itself these are prime #2,3,5,7,11,13,17,19,23

# 21 (1,3,7,21) # 23 (1,23)

n = int(input('Enter any value '))

count =0

div =[]

for i in range(1,n+1):

if( n %i == 0):

count = count +1

div.append(i)

if(count == 2):

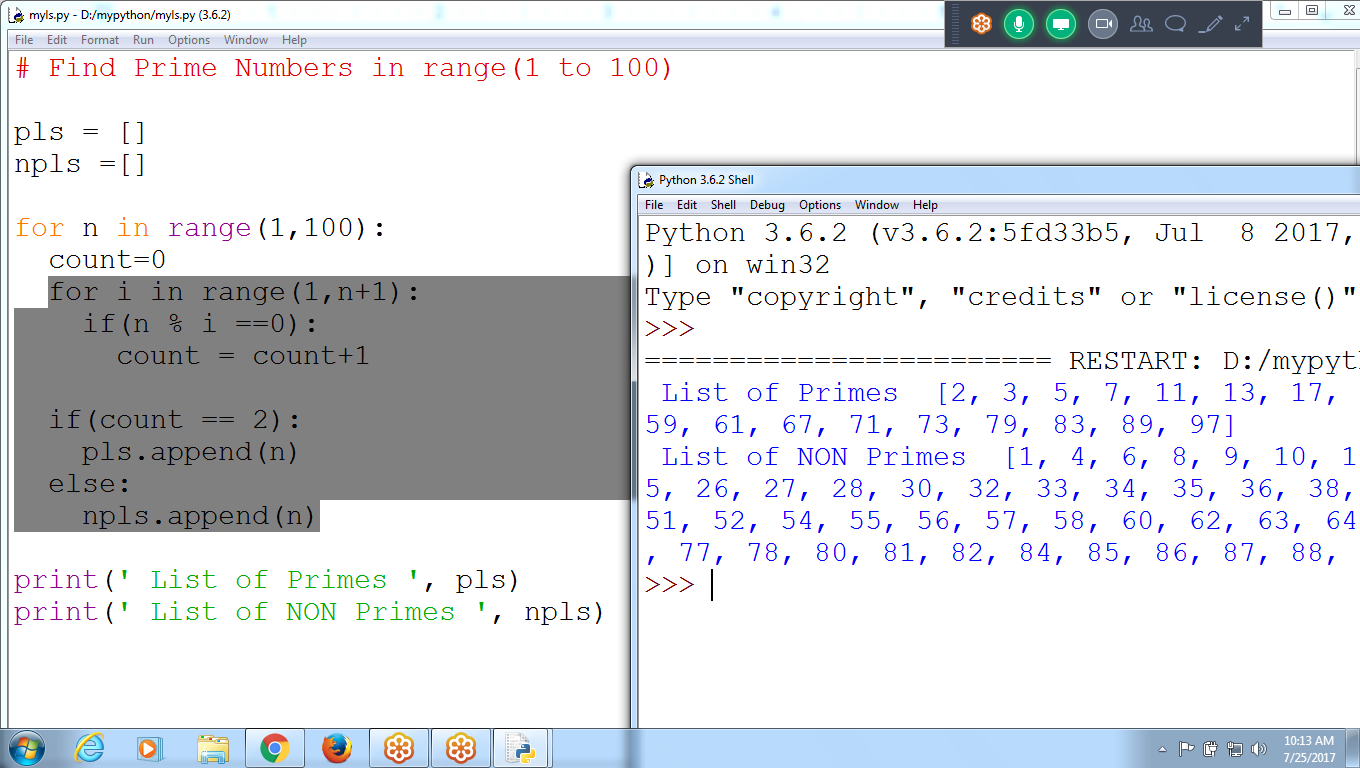
print(n, ' Is Prime Number ')

else:

print(n, ' Is not Prime number ')

print(' Divisors are ', div)

**Find List of PRIME numbers between range 1, 100**



# Find Prime Numbers in range(1 to 100)

pls = []

npls =[]

for n in range(1,100):

count=0

for i in range(1,n+1):

if(n % i ==0):

count = count+1

if(count == 2):

pls.append(n)

else:

npls.append(n)

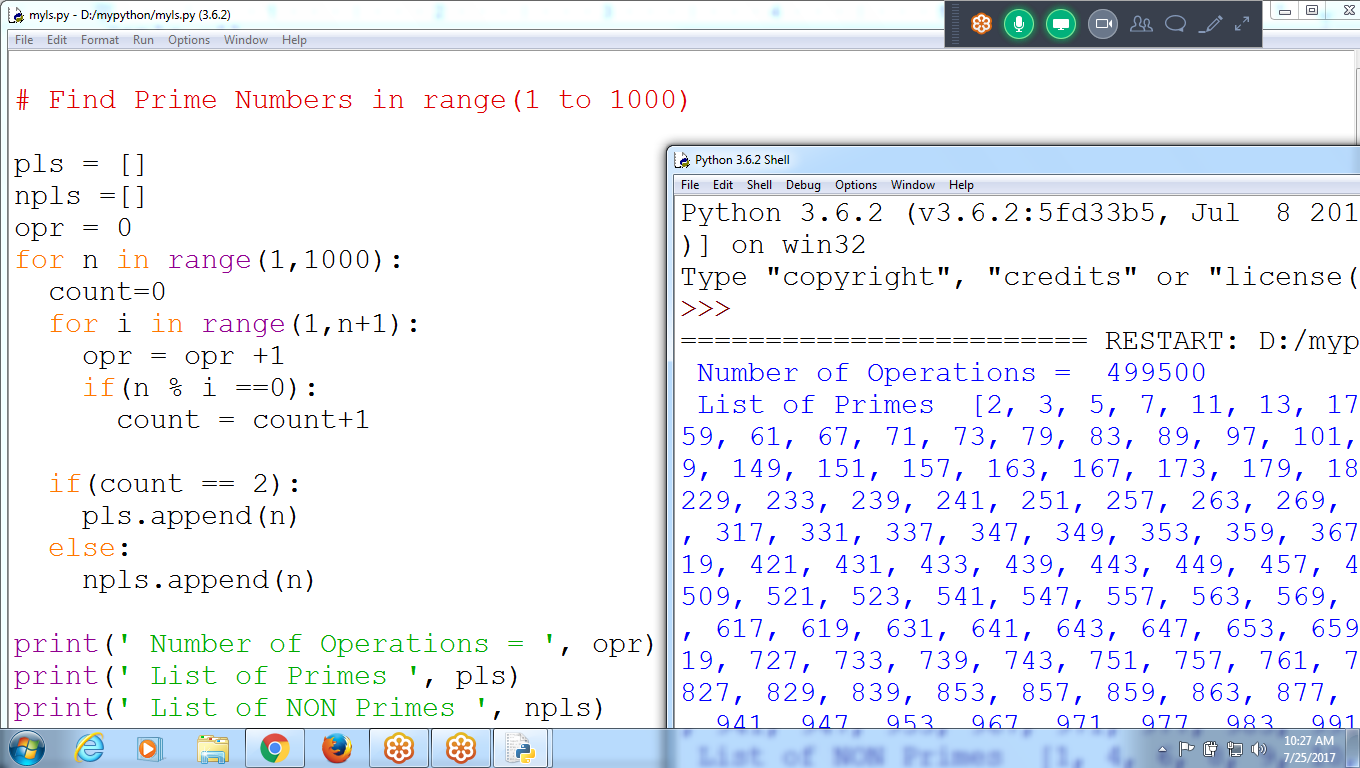
print(' List of Primes ', pls)

print(' List of NON Primes ', npls)

**HOW to Know how many operation performed**

**For 1000 primes taking :: 499500 Operations**

**Opr = opr +1**

****

**# Find Prime Numbers in range(1 to 1000)**

**pls = []**

**npls =[]**

**opr = 0**

**for n in range(1,1000):**

**count=0**

**for i in range(1,n+1):**

**opr = opr +1**

**if(n % i ==0):**

**count = count+1**

**if(count == 2):**

**pls.append(n)**

**else:**

**npls.append(n)**

**print(' Number of Operations = ', opr)**

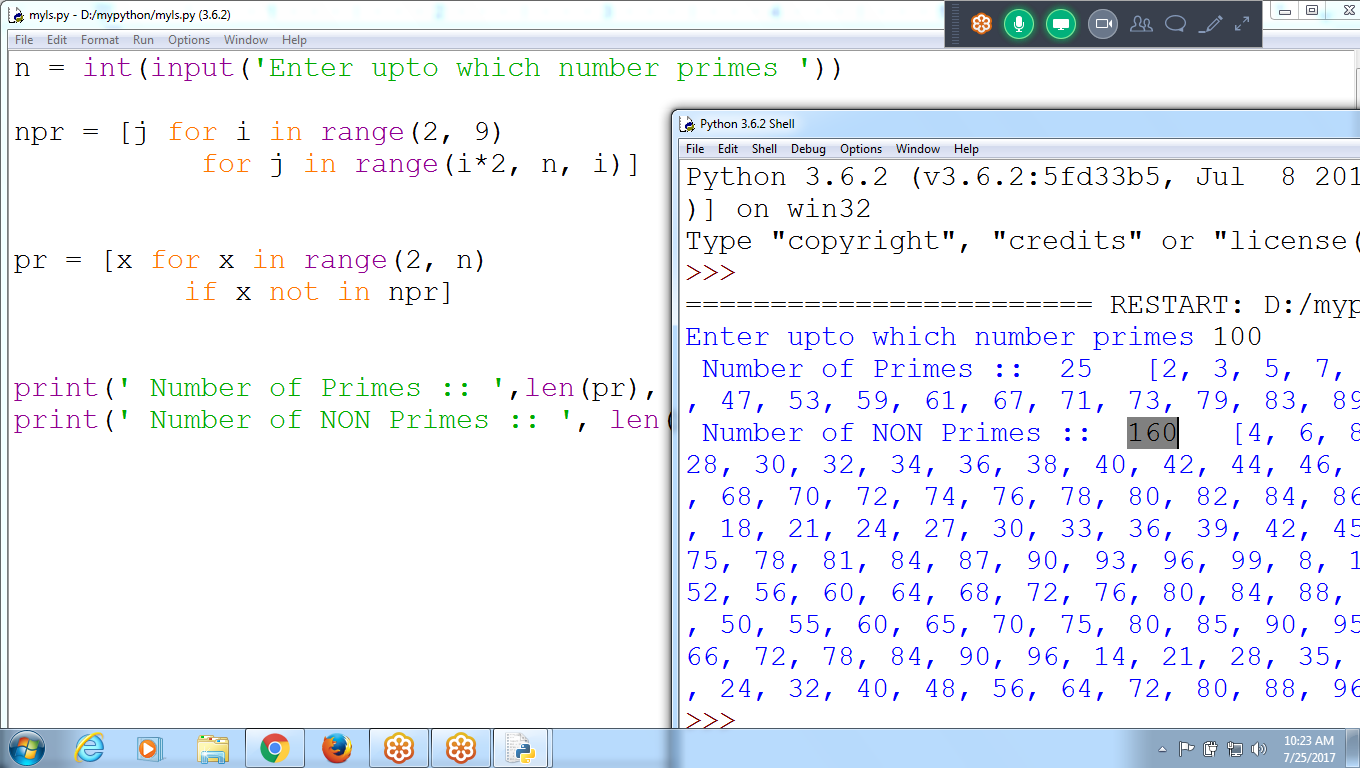
**print(' List of Primes ', pls)**

**print(' List of NON Primes ', npls)**

**Find Number of Prime Numbers with in Range :: using List**

First find list of Non Primes

Identify list of primes



Total Number of Operations : 185 (Earlier 4950)  
Non Primes : 160

Primes : 25

n = int(input('Enter upto which number primes '))

npr = [ j for i in range(2, 9)

for j in range(i\*2, n, i)]

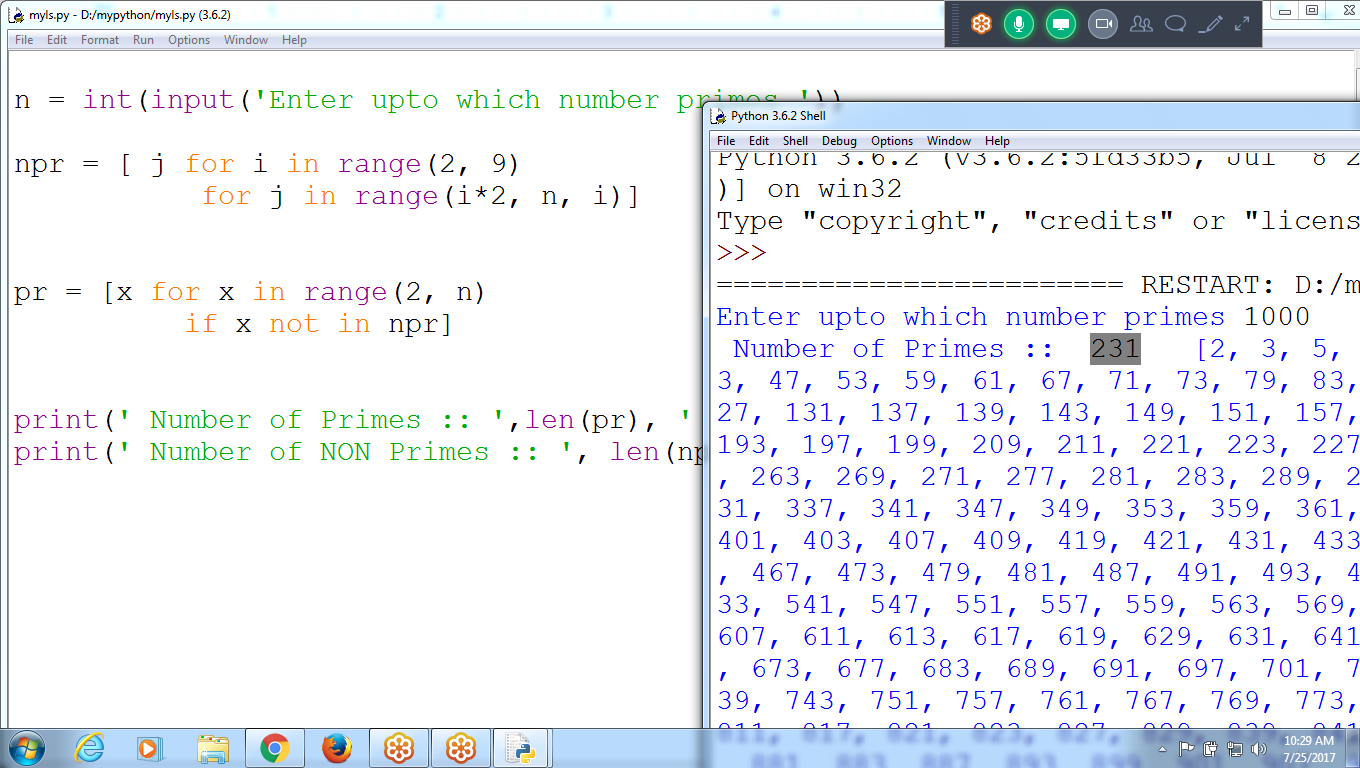
pr = [x for x in range(2, n)

if x not in npr]

print(' Number of Primes :: ',len(pr), ' ', pr)

print(' Number of NON Primes :: ', len(npr), ' ', npr)

Using List Comprehension for (1,1000) prime



What are primes :: 1 and itself :: 2, 3, 5, 7,11, 13, 17....

10 to 100

10 :: Divide from 1 to 10, get divisors : if they are 2

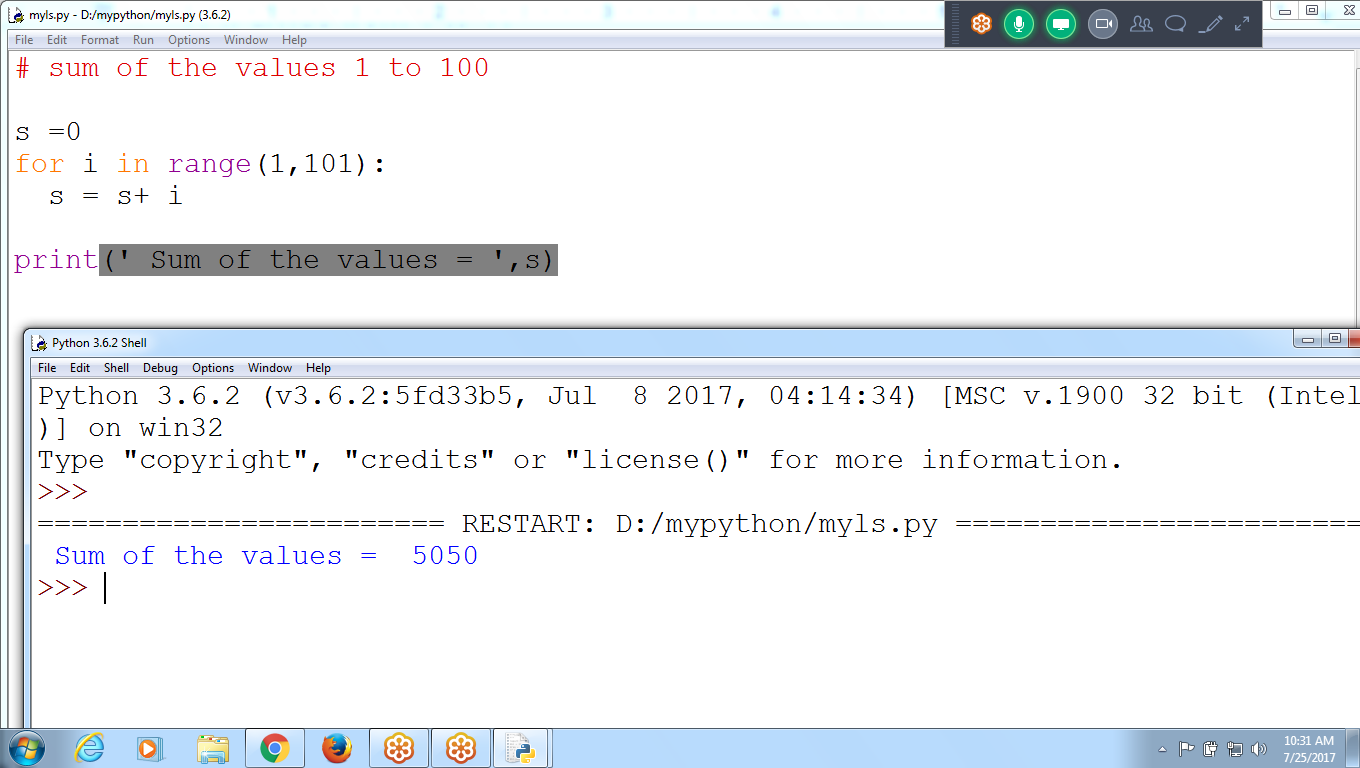
:: 10 is prime

11 :: divide from 1 to 11

90 :: divide from 1 to 90

'''

Sum of the values 1 to 100



# sum of the values 1 to 100

s =0

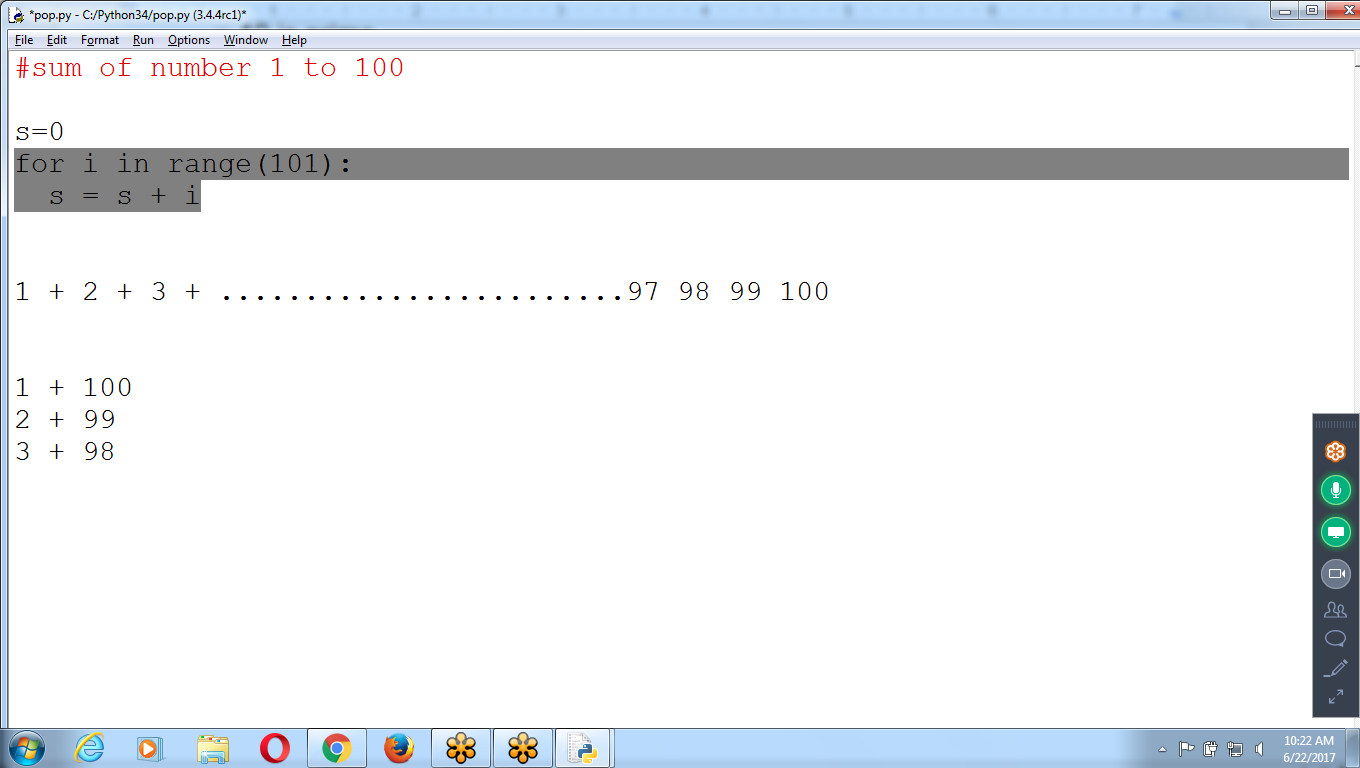
for i in range(1,101):

s = s+ i

print(' Sum of the values = ',s)

Instead of 100 operations

Can be minimized to 50 Operations



## 

## 

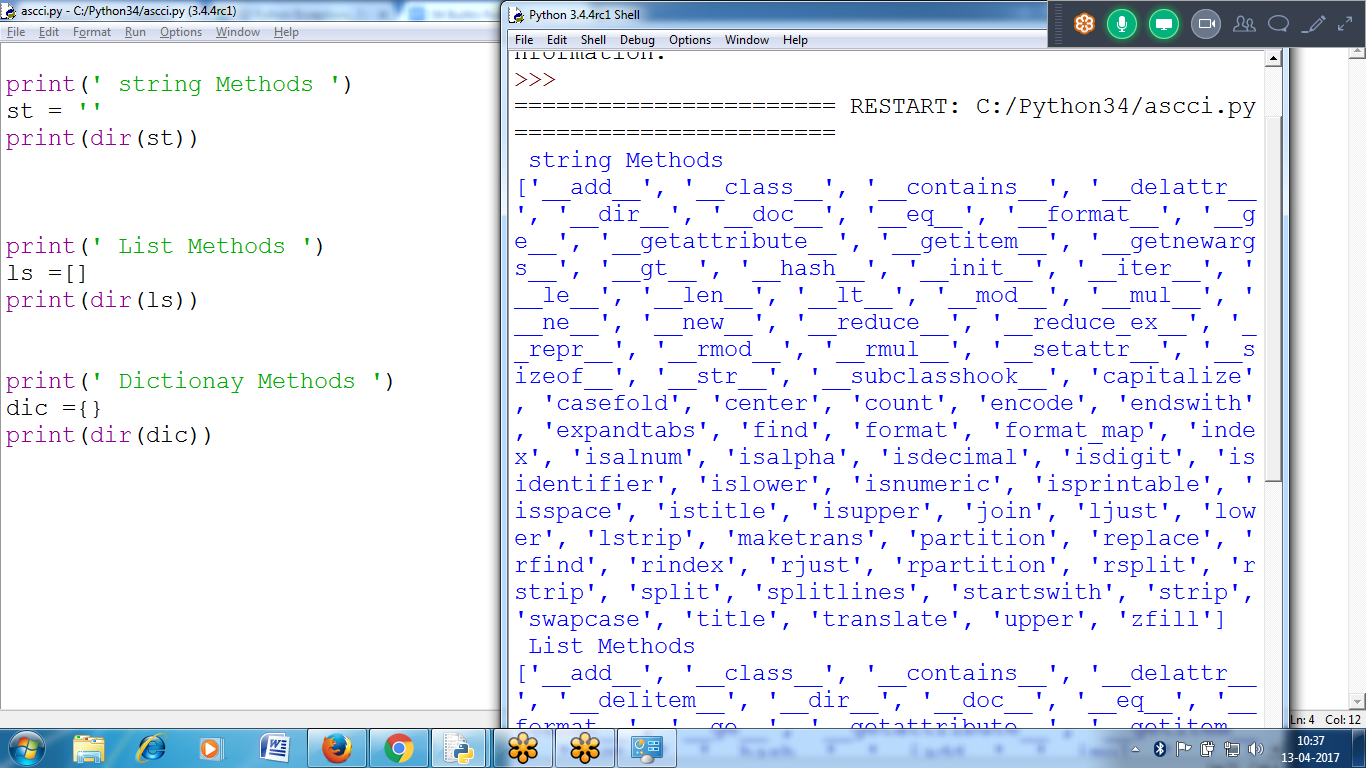
## 

## 

## 

## 

## 

**dir(**)

print(' Methods in List ')

li = []

print (dir(li) )

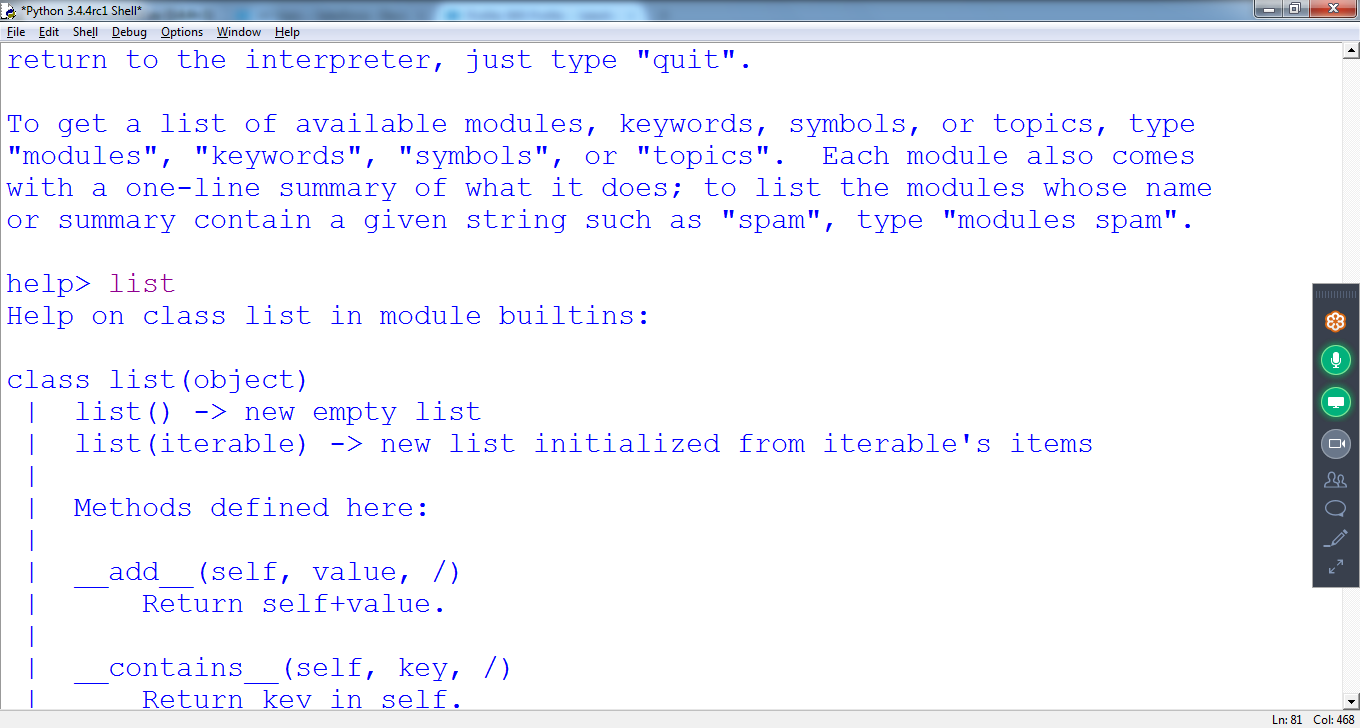
print(' Methods in Dictionary ')

d = {}

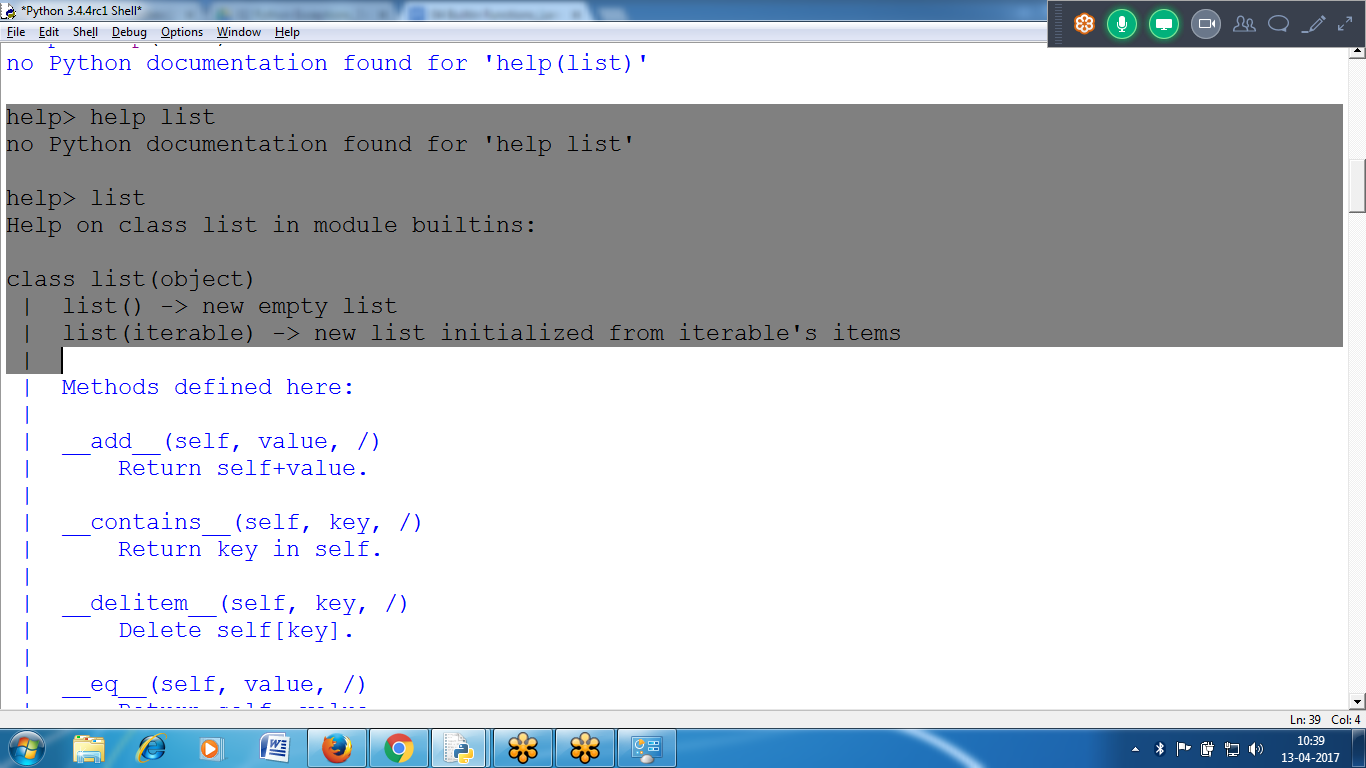
print(dir(d) )

>>> help()

help> list



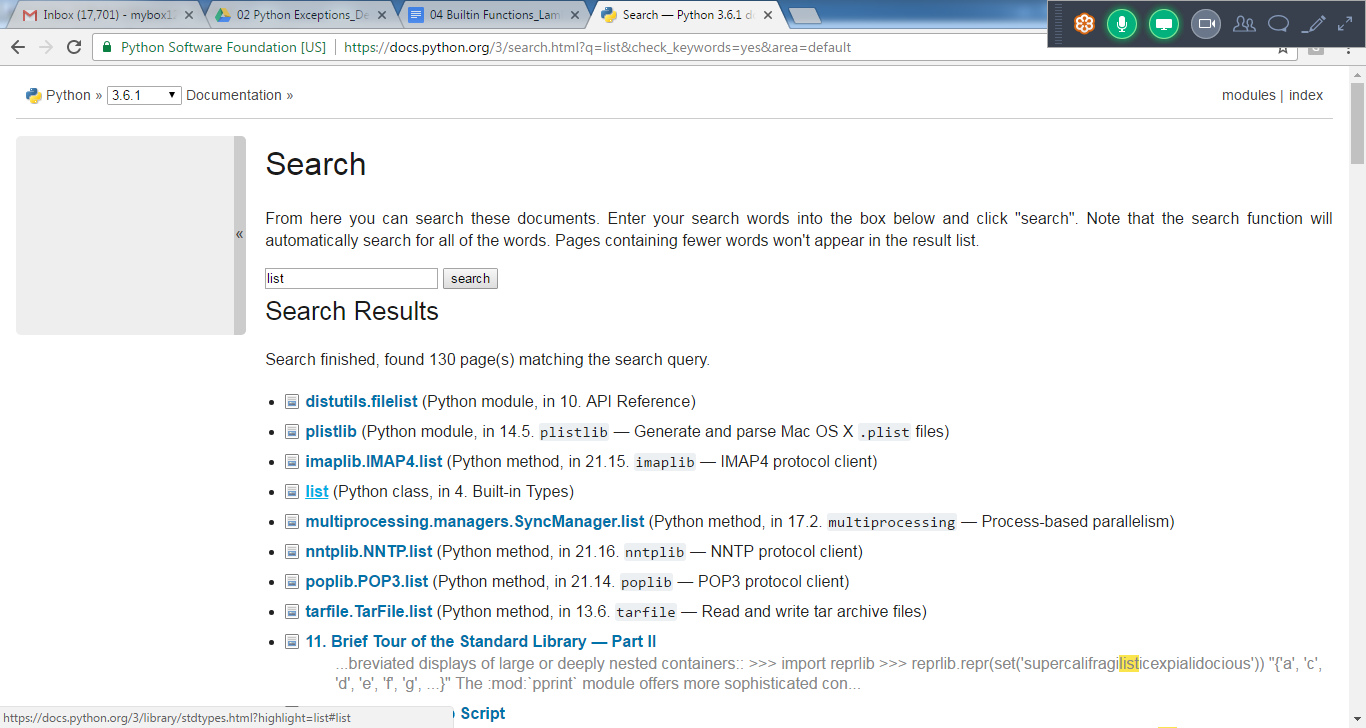
help>> list

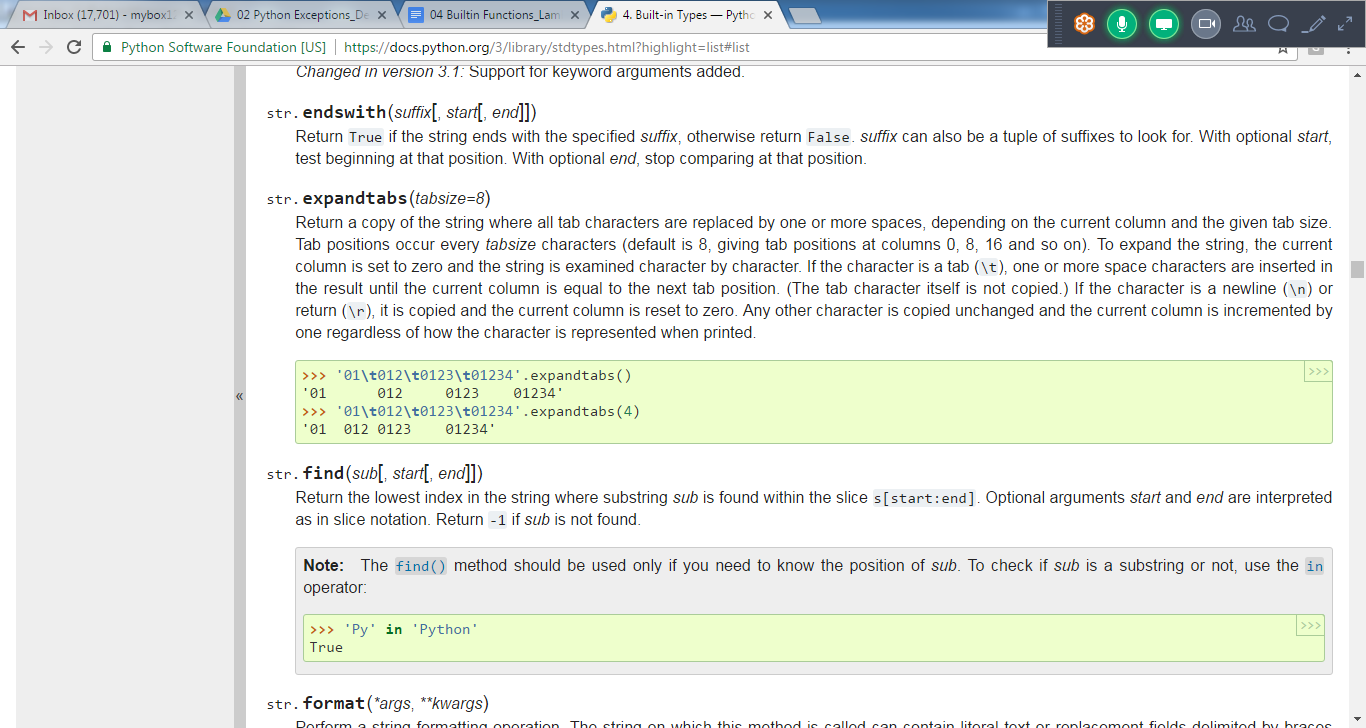


Detail Help from Python.org

docs.python.org

Search :: List, click “list”





## **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

## **Ls2 = ls1**

Means refering to same memory Block

**If ls1 changes if effects in ls2, Since List is MUTABLE Objects**

>>> ls1 = ['a', 'b', 'c']

>>> ls1

['a', 'b', 'c']

**>>> ls2 = ls1**

>>> ls2

['a', 'b', 'c']

>>> ls1[2] = 'r'

>>> ls1

['a', 'b', 'r']

>>> ls2

['a', 'b', 'r']

>>>

Integer values

**If a value is changed, b is not affecting**

>>> a=10

>>> a

10

>>> b=a

>>> b

10

>>> a

10

>>> a=20

>>> b

10

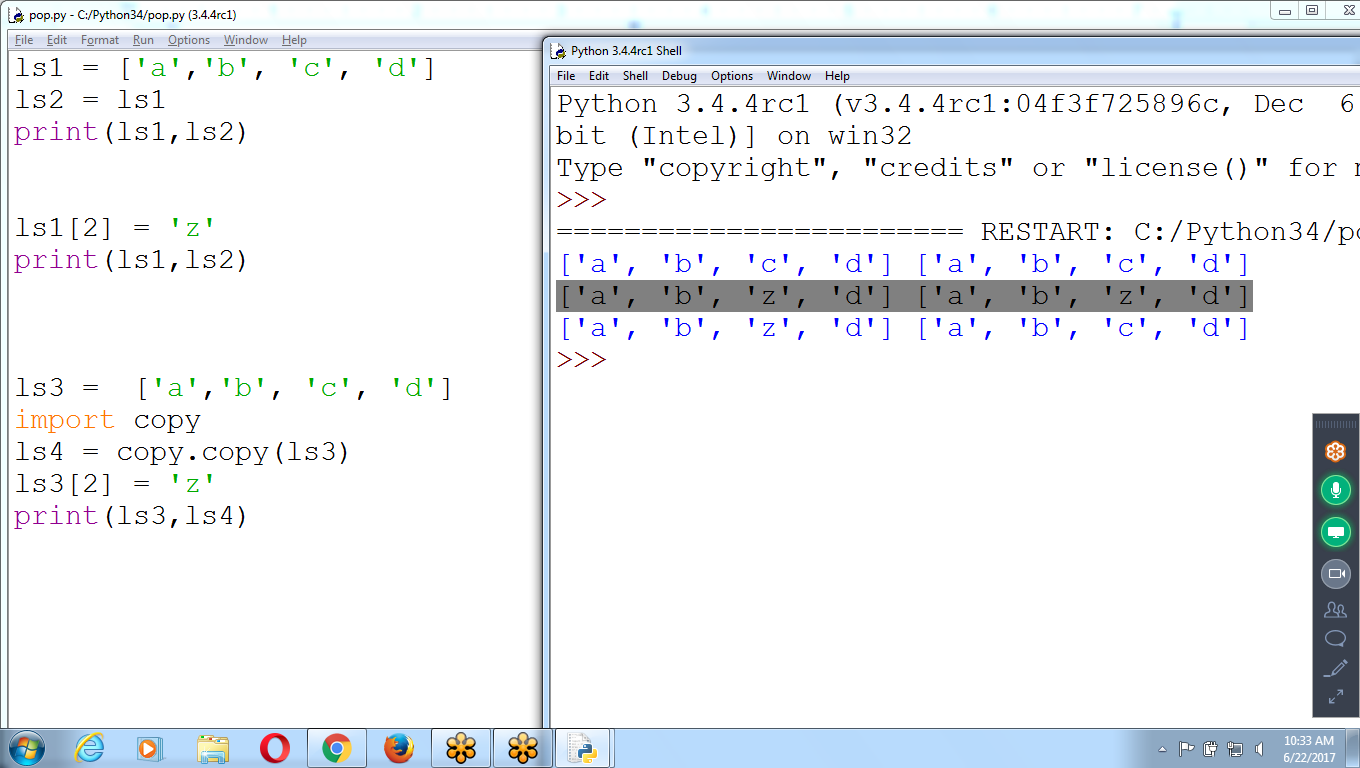
>>>

## 

## 

## **The copy Module copy()**

if the function modifies the list or dictionary that is passed, you may not want these changes in the original list or dictionary value. For this, Python provides a module named copy that provides the copy() function.



ls1 = ['a','b', 'c', 'd']

ls2 = ls1

print(ls1,ls2)

ls1[2] = 'z'

print(ls1,ls2)

ls3 = ['a','b', 'c', 'd']

import copy

ls4 = copy.copy(ls3)

ls3[2] = 'z'

print(ls3,ls4)

# 

## 

## **Comparing == and is**

## 

L1 = [1, ('a', 3)] # same value, unique objects

L2 = [1, ('a', 3)]

print(L1 == L2, L1 is L2) # equivalent?, same object?

S1 = 'python'

S2 = 'python'

print (S1 == S2, S1 is S2)

## **The Multiple Assignment**

* Assign multiple variables with the values in a list in one line of code.
* The number of variables and the length of the list must be exactly equal

>>> fruits = ['orange','apple', 'grapes']

>>> **f,a,g = fruits**

>>> f

'orange'

>>> a

'apple'

>>> g

'grapes'

>>>

# 