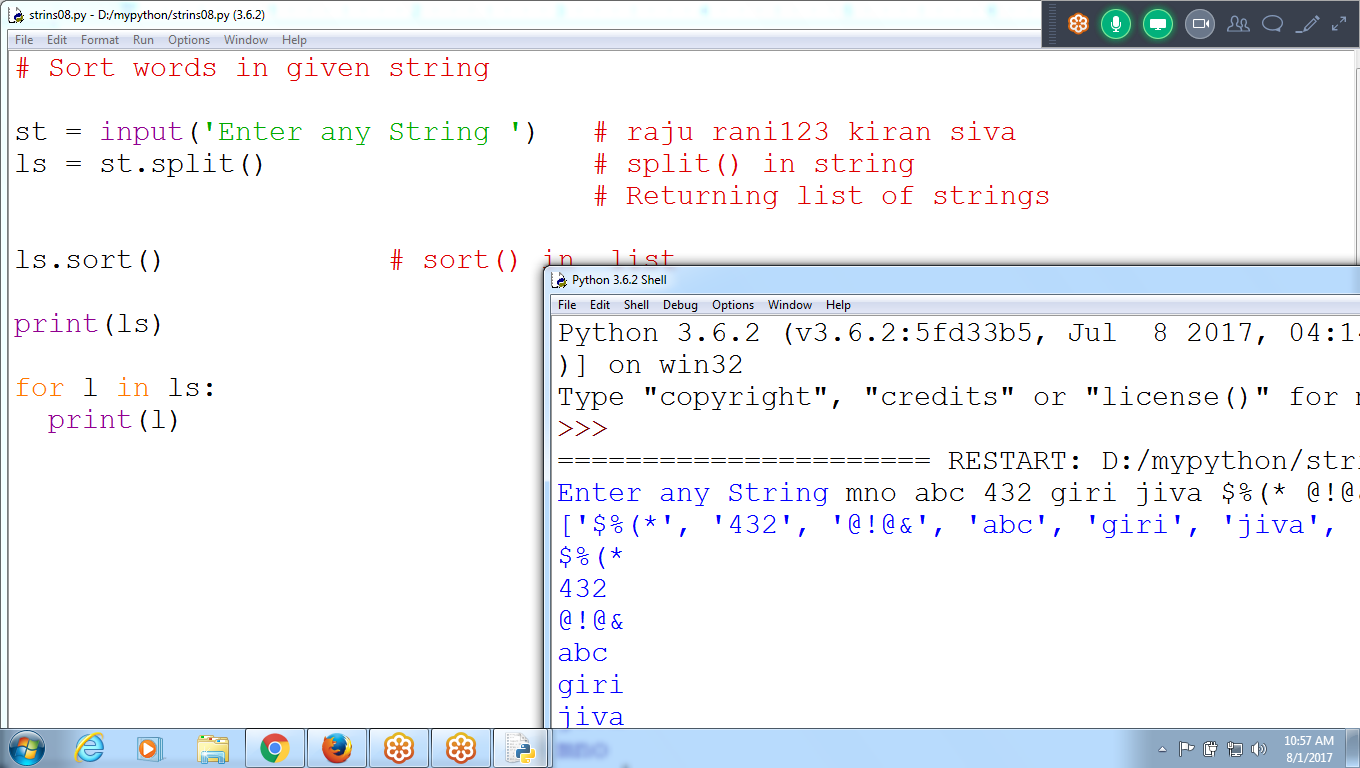
**Program to Sort Words in Alphabetic Order**

**split() returning collection of string, place in LIST type**

**sort() is method in LIST**

****

**# Sort words in given string**

**st = input('Enter any String ') # raju rani123 kiran siva**

**ls = st.split() # split() in string**

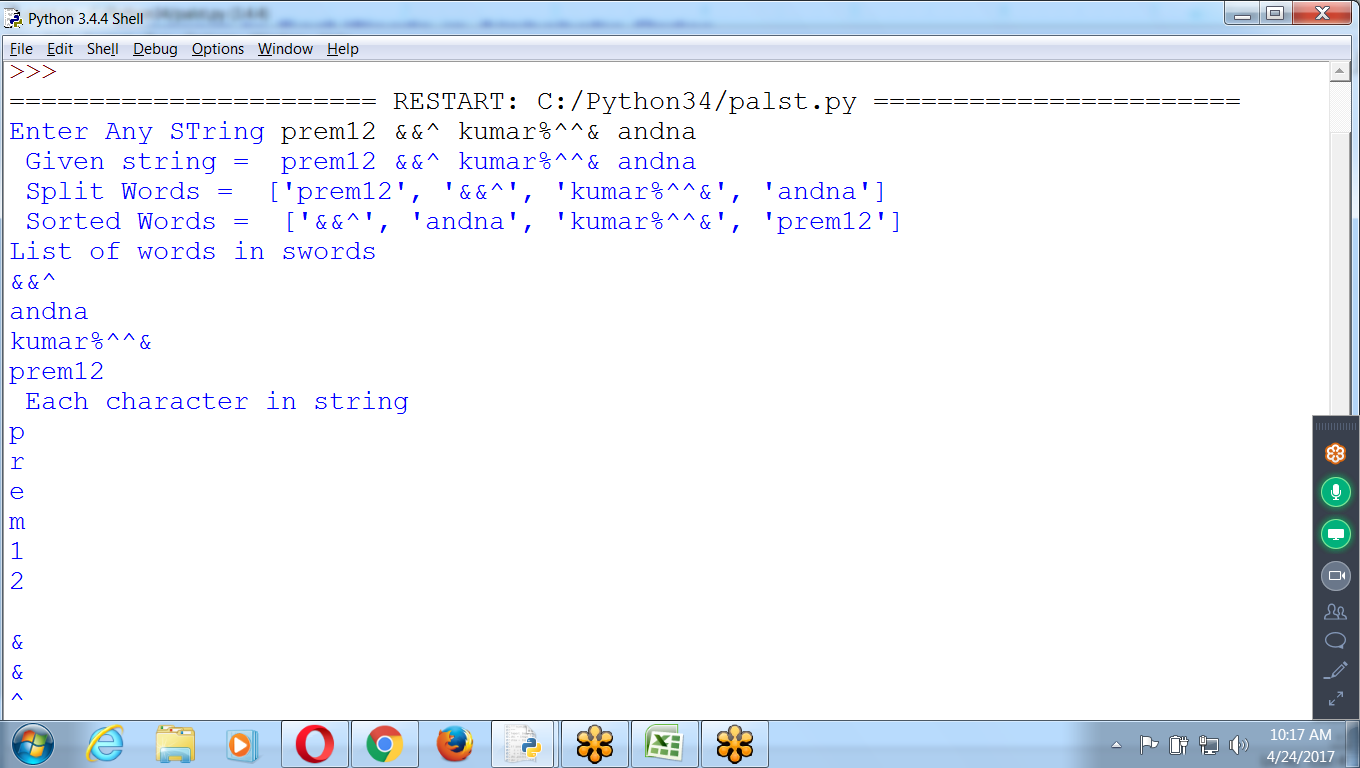
**# Returning list of strings**

**ls.sort() # sort() in list**

**print(ls)**

**for l in ls:**

**print(l)**



**# Input string and print words in SORTING Order**

**st = input('Enter Any STring ') # st is string**

**swords = st.split() # split() is method in string**

**print(' Given string = ', st)**

**print(' Split Words = ',swords) #swords is LIST**

**swords.sort() # sort() is method in List**

**print(' Sorted Words = ', swords)**

**print('List of words in swords ')**

**for w in swords: # w is is string**

**print(w)**

**print(' Each character in string ')**

**for ch in st: # ch is an character**

**print(ch)**

# **# Program to find Common Characters in string**

* : String Concatenation

Append : For LIst

# 

# Common Characters in Two strings

fs = input('Enter First String ') #pyt

ss = input('Enter Second String ') #progpt

cc = ''

cls = []

for f in fs:

if f in ss:

cc = cc + f # adding to string concatenation

cls.append(f) # adding character to List

print(' Given Strings are ')

print(' fs = ',fs)

print(' ss = ',ss)

print(' Common Characters are ', cc)

print(' Common list ', cls)

**Python Program to Check Whether a String is Palindrome or Not**

**Palindrome : 12321**

**: anana, madam, malayam**

****

**# Check whether given string is palindrome or not**

**fs = input('Enter First String ')**

**print(reversed(fs))**

**print('fs = ', fs)**

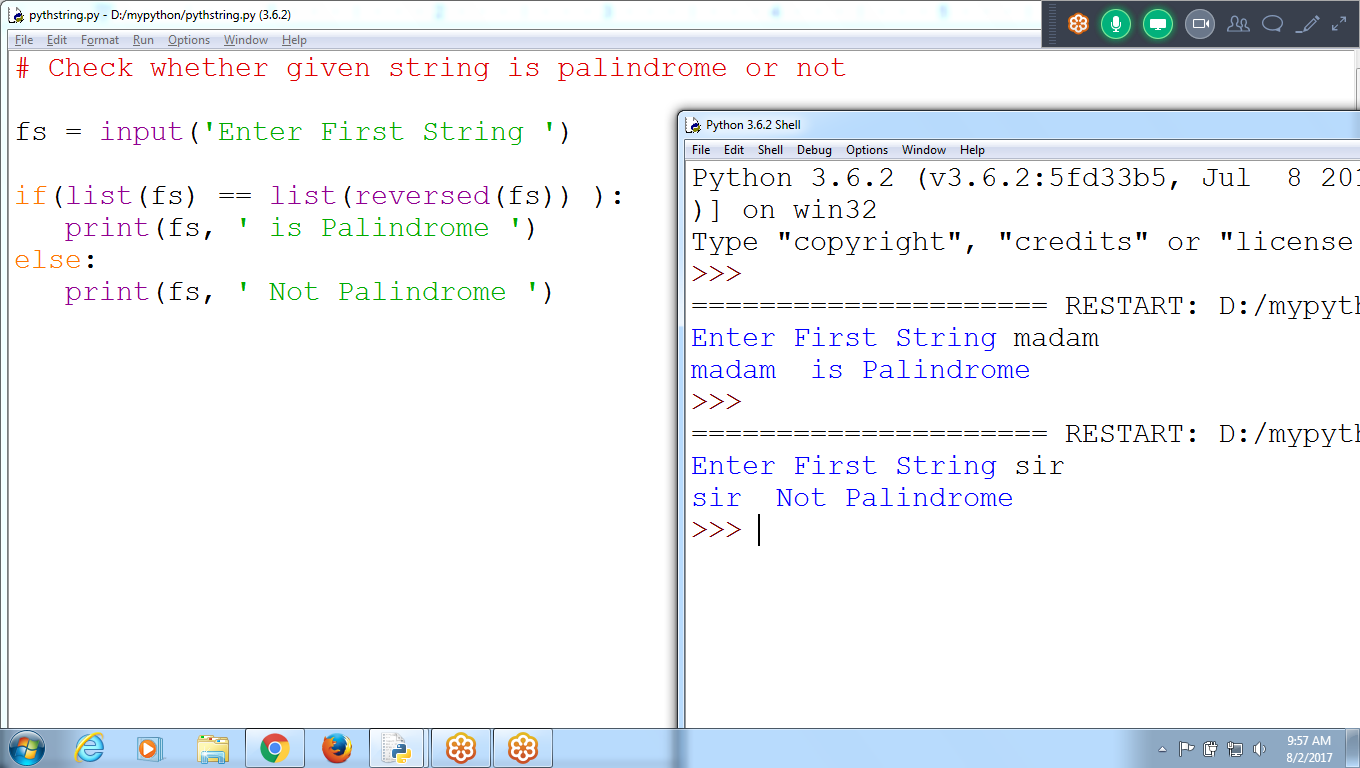
**print('rs = ', list(reversed(fs)))**

**if(list(fs) == list(reversed(fs))):**

**print(fs,' is palindrome ')**

**else:**

**print(fs,' NOt Palindrome ')**

****

**# Check whether given string is palindrome or not**

**fs = input('Enter First String ')**

**if(list(fs) == list(reversed(fs)) ):**

**print(fs, ' is Palindrome ')**

**else:**

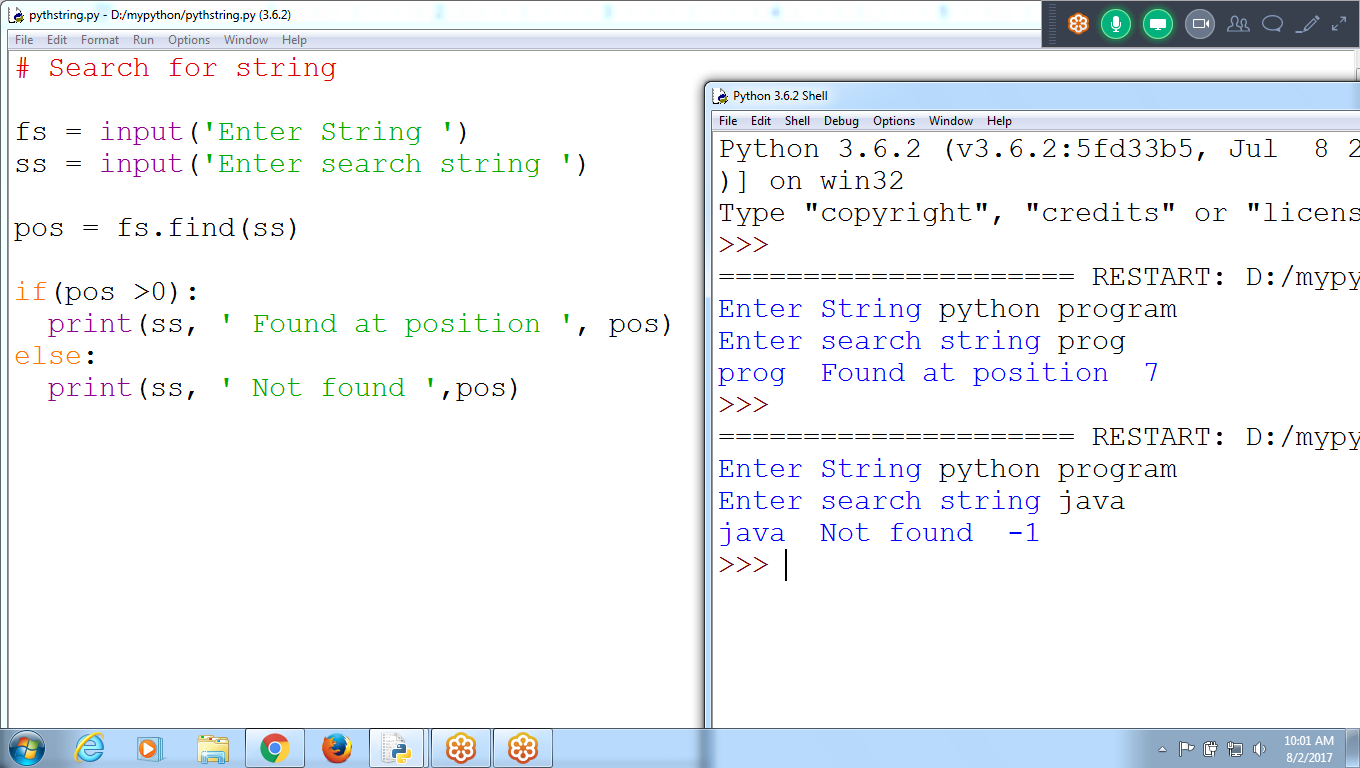
**print(fs, ' Not Palindrome ')**

# 

# 

# **# Search for a given string**

Not Found returns -1



# Search for string

fs = input('Enter String ')

ss = input('Enter search string ')

pos = fs.find(ss)

if(pos >0):

print(ss, ' Found at position ', pos)

else:

print(ss, ' Not found ',pos)

**To Print ASCII Code, Character, Binary, Decimal, hexa**

**Python 2 Version : uses ASCII code ( 8 bits)**

**Python 3 Version : uses UNICODE (16 bits)**

**\*8 Bits ( one Byte)**

**\*16 Bits ( 2 Bytes)**

**Binary Format ( Contains only 0 ‘s and 1’s)**

**1**

**1**

**---**

**10**

**----**

**0**

**1**

**---**

**1**

**---**

**1 0**

**1**

**-------**

**1 1**

**--------**

**1 1**

**1**

**-------------**

**1 0 0**

**8 Bits**

**0 0 0 0 0 0 0 0 =0**

**0 0 0 0 0 0 0 1 = 1**

**0 0 0 0 0 0 1 0 = 2**

**0 0 0 0 0 0 1 1 = 3**

**0 0 0 0 0 1 0 0 =4**

**0 0 0 0 0 1 0 1 = 5**

**0 0 0 0 0 1 1 0 = 6**

**0 0 0 0 0 1 1 1 = 7**

**0 0 0 0 0 1 1 1**

**2(2) 2(1) 2 (0)**

**4 + 2 + 1 = 7**

**8 Bits**

**How many Combinations**

**3 bits = 8 Combinations**

**0 0 0**

**0 0 1**

**0 1 0**

**0 1 1**

**1 0 0**

**1 0 1**

**1 1 0**

**1 1 1**

**Formula for 3 Bits**

**: each cell can store 2 values (0 or 1)**

**: 2 power (3 cells)**

**: 2 power(3) :: 8**

**Formula for 8 Bits**

**: each cell can store 2 values**

**: 2 power(8) :: 256**

**Formula for 16 Bits**

**: each cell can store 2 values**

**: 2 power(16) :: 65,536**

**C lang : 8 bits : ascii : 256 values**

**a-z : 26**

**A-Z : 26**

**0-9 : 10**

**~!@#$%^&\*()\_+**

**>>> ord('a')**

**97**

**>>>**

**>>>**

**>>> ord('0')**

**48**

**>>> ord('1')**

**49**

**>>> ord('9')**

**57**

**>>>**

**>>>**

**>>> ord('A')**

**65**

**>>>**

**On Typing A : 65 Ascii code**

**2)65**

**32 - 1**

**------**

**16 - 0**

**8 - 0**

**4 -0**

**2 - 0**

**1 - 0**

**Bottom to TOP ( 0100 0001)**

**8 bits**

**0 1 0 0 0 0 0 1**

**2(6) 2(5) 2(2) 2(1) 2(0)**

**0+ 64+ 0+ 0+ 0+ 0+ 0+ 1 = 65**

**Result : 65**

**8 bits : can store : 256 Combinations**

**256 Symbols (alphabets, integers, special charac)**

**16 bits : UNICODE : 2 power (16): 65536 COmbinations**

**>>> ord('A')**

**65**

**>>> ord('a')**

**97**

**>>> bin(65)**

**'0b1000001'**

**>>>**

**>>> chr(48)**

**'0'**

**>>> chr(58)**

**':'**

**>>> chr(57)**

**'9'**

**>>>**

**>>> ord('0')**

**48**

**>>> chr(65)**

**'A'**

**>>> chr(30)**

**'\x1e' # Printer Characters**

**>>> chr(40)**

**'('**

**>>> chr(39)**

**"'" # Single Quote**

**>>>**

**>>> a = 25**

**>>> bin(a)**

**'0b**11001**'**

**2)25**

**-----**

**2) 12 - 1**

**-----**

**6 - 0**

**\_\_\_**

**2) 3 - 0**

**-------**

**1 - 1**

**Bottom to Top**

**1 1 0 0 1**

**2 power4 2 power3 2 power2 2 power1 2 power 0**

**(1101) base 2 = ( 25) base 10**

**(16\*1) + (8 \*1)+ (4\*0) + (2 \*0) + (1\*1)**

**16 + 8 + 0 + 0 + 1**

**(25)base 10**

**BInary Contains :: 0’s and 1’s**

**>>> oct(a) # 0 to 7**

**'0o31'**

**>>>**

**8) 25**

**24 - 3**

**--------**

**1**

**Bottom to TOP**

**3 1**

**8 power1 8 power 0**

**(8\*3) + (1\*1)**

**24 + 1 = 25**

**Oct contains 0 to 7 values**

**>>> a = 25**

**>>> a**

**25**

**>>> bin(a)**

**'0b11001'**

**>>> oct(a)**

**'0o31'**

**>>> hex(a)**

**'0x19'**

**>>>**

**>>> oct(dec)**

**'0o2310'**

**>>> hex(dec) # 0 to 15**

**'0x4c8'**

**>>>**

5 positions : binary : 2 power 5 :: each cell can store o or 1

5 position in HEXa : each cell can store 0 to 15 : 15 Power 5

>>> 2 \*\* 8 (python 2 Version:: ASCII

256

>>> 2 \*\* 16 (python 3 : UNICODE)

65536

>>>

ASCII : 8 bits ( in each bit can store o or 1)

Combinations : 0 0 0 0 0 0 0 0 : 0

0 0 0 0 0 0 0 1 : 1

0 0 0 0 0 0 1 1

Each cell can store (0 or 1) : 2 values

ASCII Follows : 8 cells

Combinations: 2 power (8) = 256

ASCII can store : 256 symbols

0 to 255 Combinations

Python, Java, php, : UNICODE Format : 16 Bits

UNICODE Follows : 16 cells

Combinations: 2 power (16) = 65,536

UNICODE can store : 65536 symbols

0 to 65535 Combinations

# Display list of UNICODES

for i in range(65536):

if(i%20 ==0):

input('Press Any Key Continue ')

else:

print(i, ' Symbol = ', chr(i), ' Binary = ', bin(i))

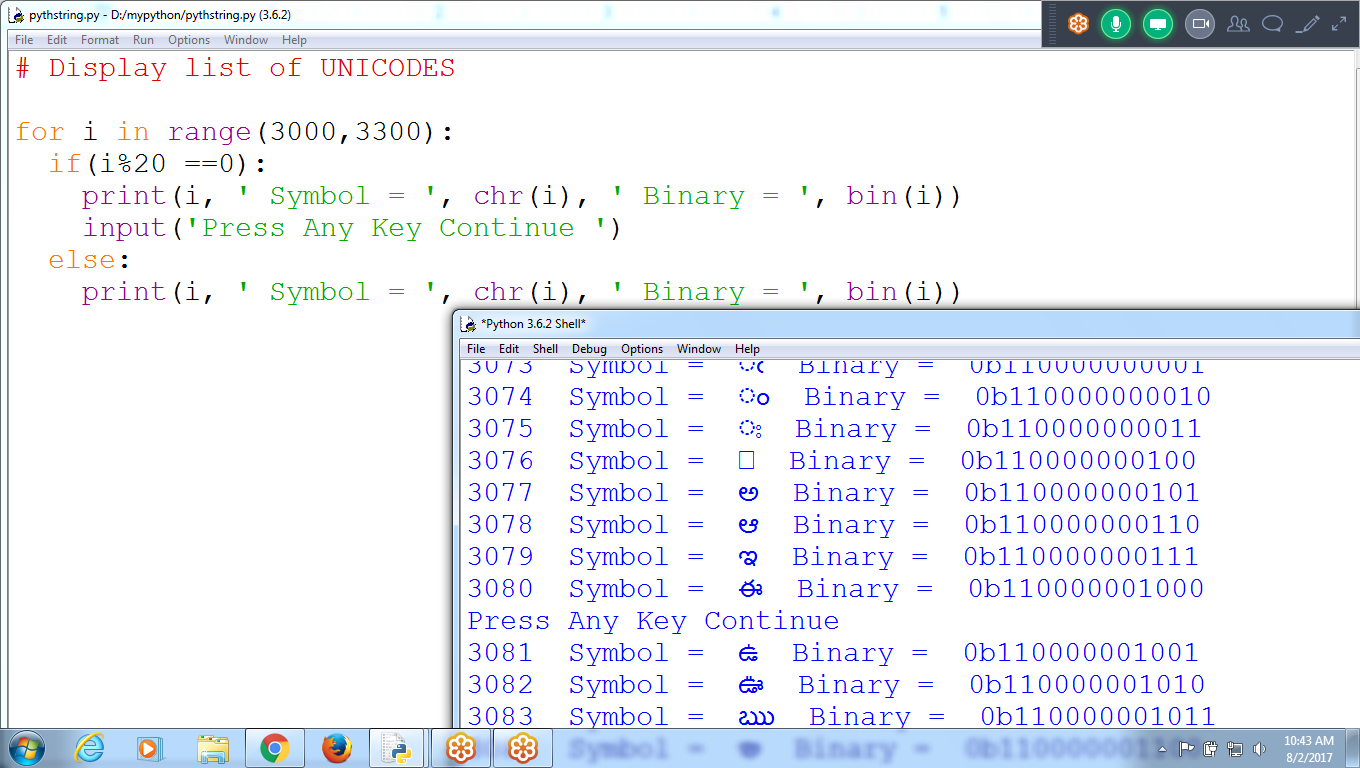


Display UNICODE Characters per page 20 characters

**>>> chr(3077)**

**'అ'**

Display Telugu Characters



# Display list of UNICODES

for i in range(3000,3300):

if(i%20 ==0):

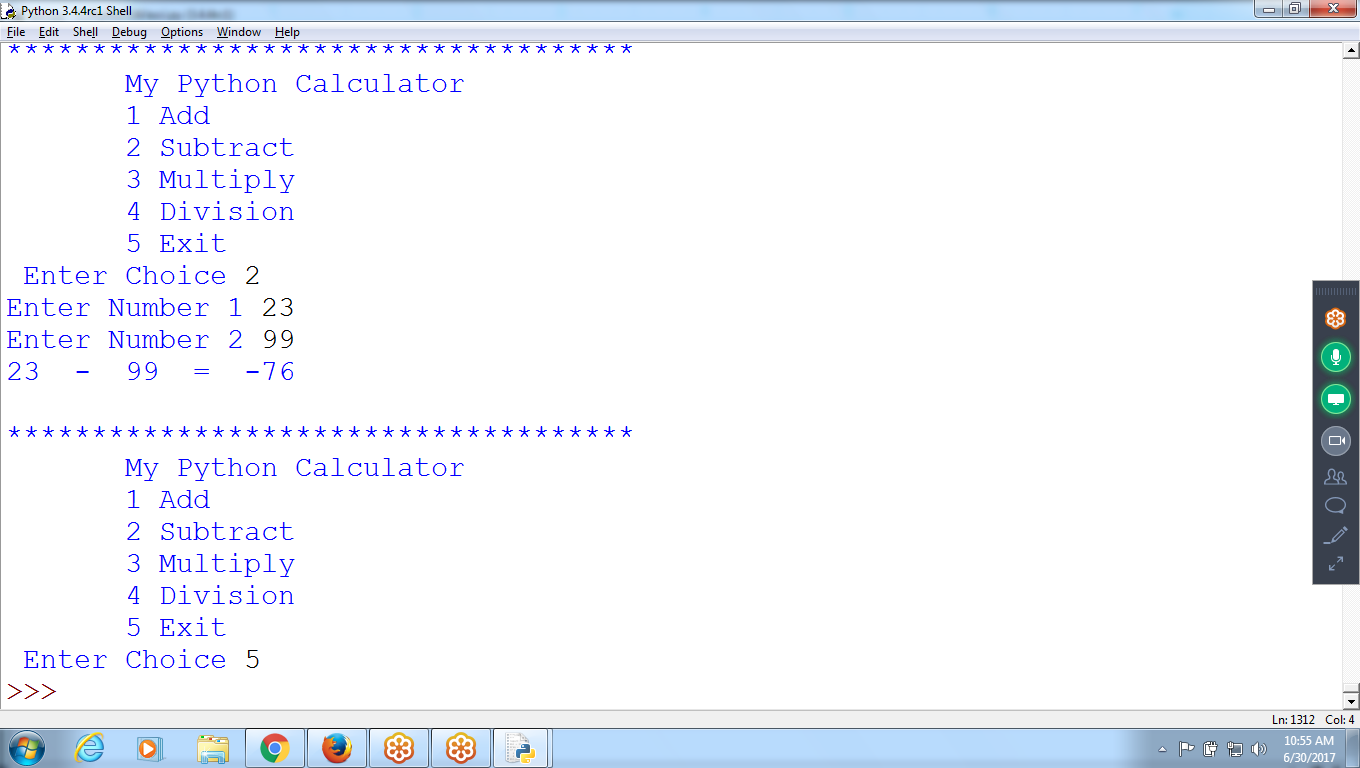
print(i, ' Symbol = ', chr(i), ' Binary = ', bin(i))

input('Press Any Key Continue ')

else:

print(i, ' Symbol = ', chr(i), ' Binary = ', bin(i))

Program to Calculate numbers



# Calculator

def add(a,b):

return a + b

def sub(a,b):

return a - b

def mul(a,b):

return a \* b

def div(a,b):

return a / b

**while True:**

print('''

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

My Python Calculator

1 Add

2 Subtract

3 Multiply

4 Division

5 Exit ''')

ch = input(' Enter Choice ')

**if(ch == '5'):**

**break**

else:

**num1 = int(input('Enter Number 1 '))**

**num2 = int(input('Enter Number 2 '))**

**if(ch == '1'):**

**print(num1, ' + ', num2, ' = ', add(num1,num2))**

**elif( ch == '2'):**

**print(num1, ' - ', num2, ' = ', sub(num1,num2))**

**elif( ch == '3'):**

**print(num1, ' \* ', num2, ' = ', mul(num1,num2))**

**elif( ch == '4'):**

**print(num1, ' / ', num2, ' = ', div(num1,num2))**

**else:**

**print(' INvalid Input ')**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Daily : 3 Hours**

**Python**

**Ksihore : django**

**GOPI: Weekend Django and python :**

**Advance:**

**M 1: Python**

**M2 : django**

**M3 : Advance Python : 30**

**M4: Django Projects : 30**