# **Working with Strings**

## **String Literals**

### 

Strings, files, OS Operations : Scripting lang

General Purpose

Web scraping, Database, Multithreading, Network, GUI, Generators, Regurlar expres

Editors ( Pycharm, anaconda, juptyer)

Interview questions :

Freshers python JOBS

### **Double Quotes**

using double quotes is that the string can have a single quote character in it

>>> **str = "WEB Technolog "**

Use both single quotes and double quotes in the string, need to use escape characters.

>>> st5 = 'ram\'s House \t\t Ameer\\pet \n Hyderabad \n Telangana'

>>> print(st5)

ram's House Ameer\pet

Hyderabad

Telangana

|  |  |
| --- | --- |
| **Escape character** | **Prints as** |
| \' | Single quote |
| \" | Double quote |
| \t | Tab |
| \n | Newline (line break) |
| \\ | Backslash |

>>> st = 'raju\'straining\\trainer\nsiva\tkumar\"jeevan'

>>> print(st)

raju'straining\trainer

siva kumar"jeevan

>>>

**R Ignores Escape Sequences**

>>> st = r'raju\'straining\\trainer\nsiva\tkumar\"jeevan'

>>> print(st)

raju\'straining\\trainer\nsiva\tkumar\"jeevan

>>>

### **Raw Strings**

A *raw string* completely **ignores all escape characters** and prints any backslash that appears in the string

>>> st = 'ram\'s House \t\t Ameer\\pet \n Hyderabad \n Telangana'

>>> print(st)

ram's House Ameer\pet

Hyderabad

Telangana

>>>

>>>

>>>

>>>

>>>

>>> st = **r**'ram\'s House \t\t Ameer\\pet \n Hyderabad \n Telangana'

>>> print(st)

ram\'s House \t\t Ameer\\pet \n Hyderabad \n Telangana

>>>

>>>

>>>

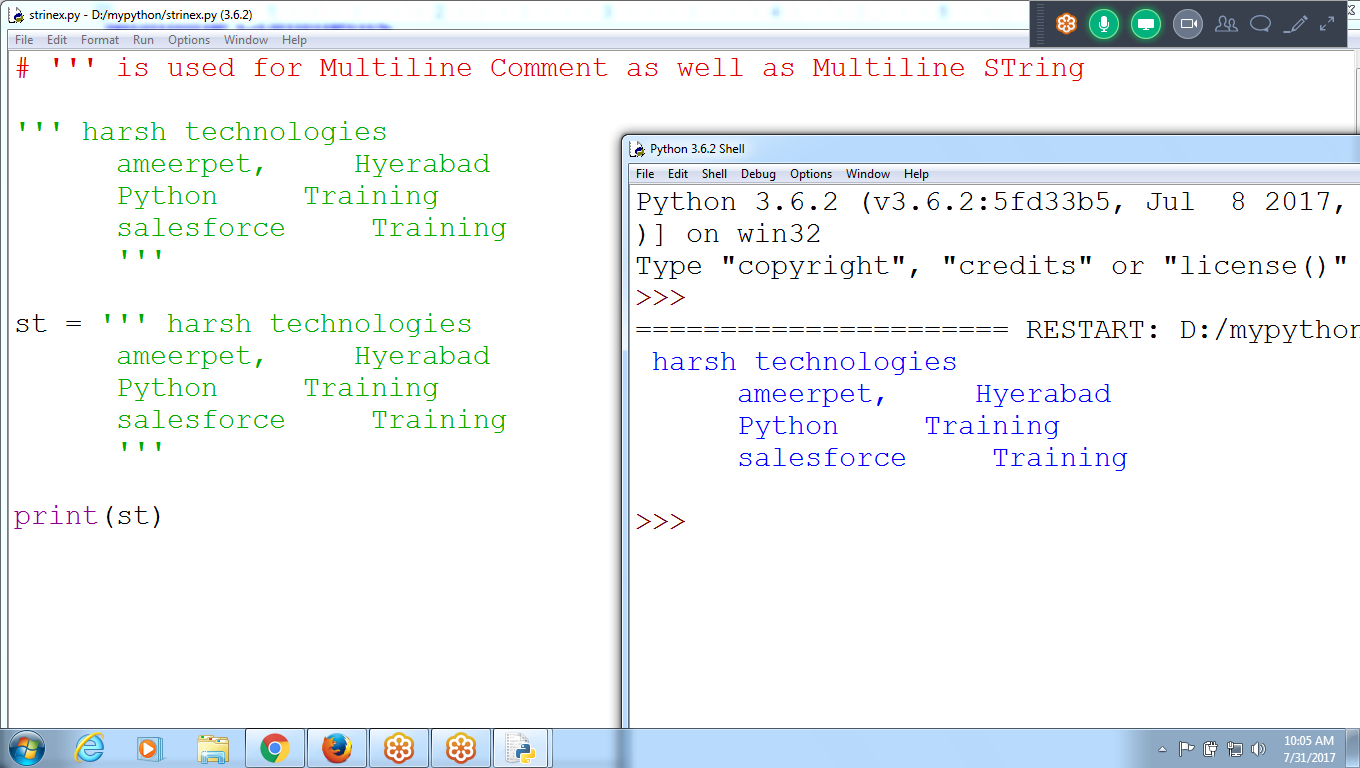
**Python considers the backslash as part of the string and not as the start of an escape character**

### **Multiline Strings with Triple Quotes**

* A multiline string in Python begins and ends with either three single quotes or three double quotes.
* Any quotes, tabs, or newlines in between the “triple quotes” are considered part of the string

### **Multiline Comments**

* hash character (#) marks the beginning of a comment for the rest of the line
* “””” A multiline string is often used for comments that span multiple lines



## **Indexing and Slicing Strings**

Strings use indexes and slices

The space and exclamation point are included in the character count, so 'Hello world!' is 12 characters long, from H at index 0 to ! at index 11.

**String Indexing**

st = 'helloworld' # 0 to 9

print('Length of the string ',len(st))

print('String =',st)

print('Zeroth Position ', st[0])

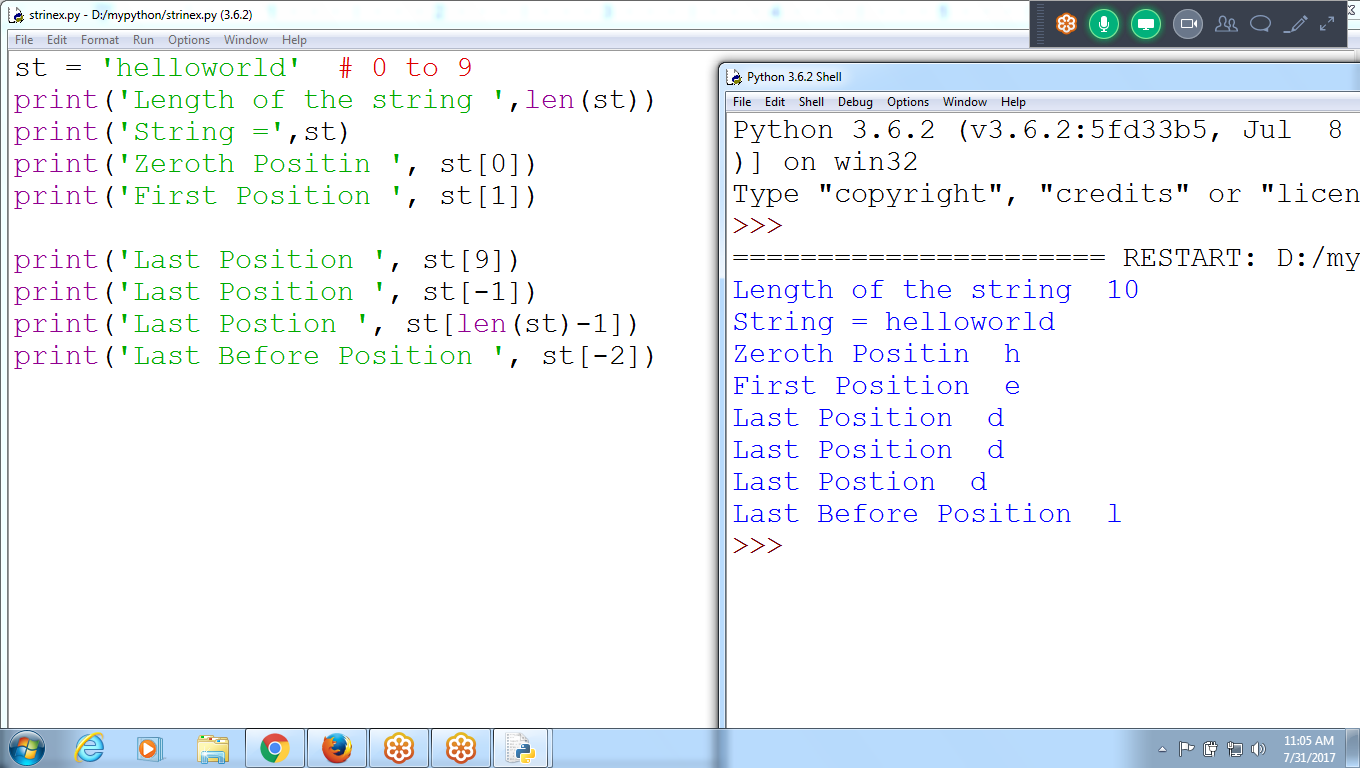
print('First Position ', st[1])

print('Last Position ', st[9])

print('Last Position ', st[-1])

**print('Last Postion ', st[len(st)-1])**

print('Last Before Position ', st[-2])



**String Slicing (range)**

# Example on STring Slicing

st = 'helloworld' # 0 to 9

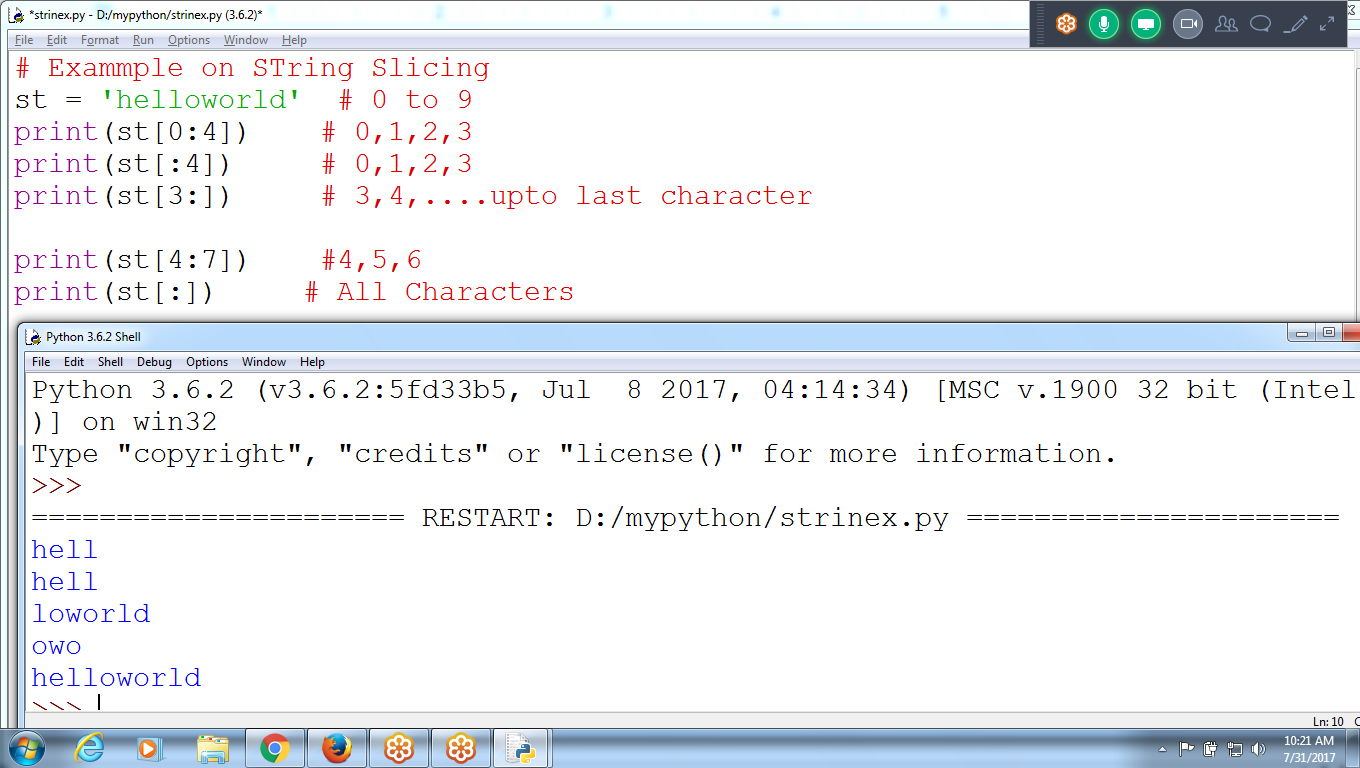
print(st[0:4]) # 0,1,2,3

print(st[:4]) **# 0,1,2,3**

print(st[3:]) **# 3,4,....upto last character**

print(st[4:7]) #4,5,6

print(st[:]) # All Characters



## **The in and not in Operators with Strings**

* The **in and not in** operators can be used with strings
* An expression with two strings joined using in or not in will evaluate to a Boolean **True or False**

**st = 'helloworld'**

**st2 = 'hello'**

**st3 = 'Hello' # Upper Case "H"**

**st4 = 'llo'**

**print(' Using IN Operator ')**

**print(st2 in st)**

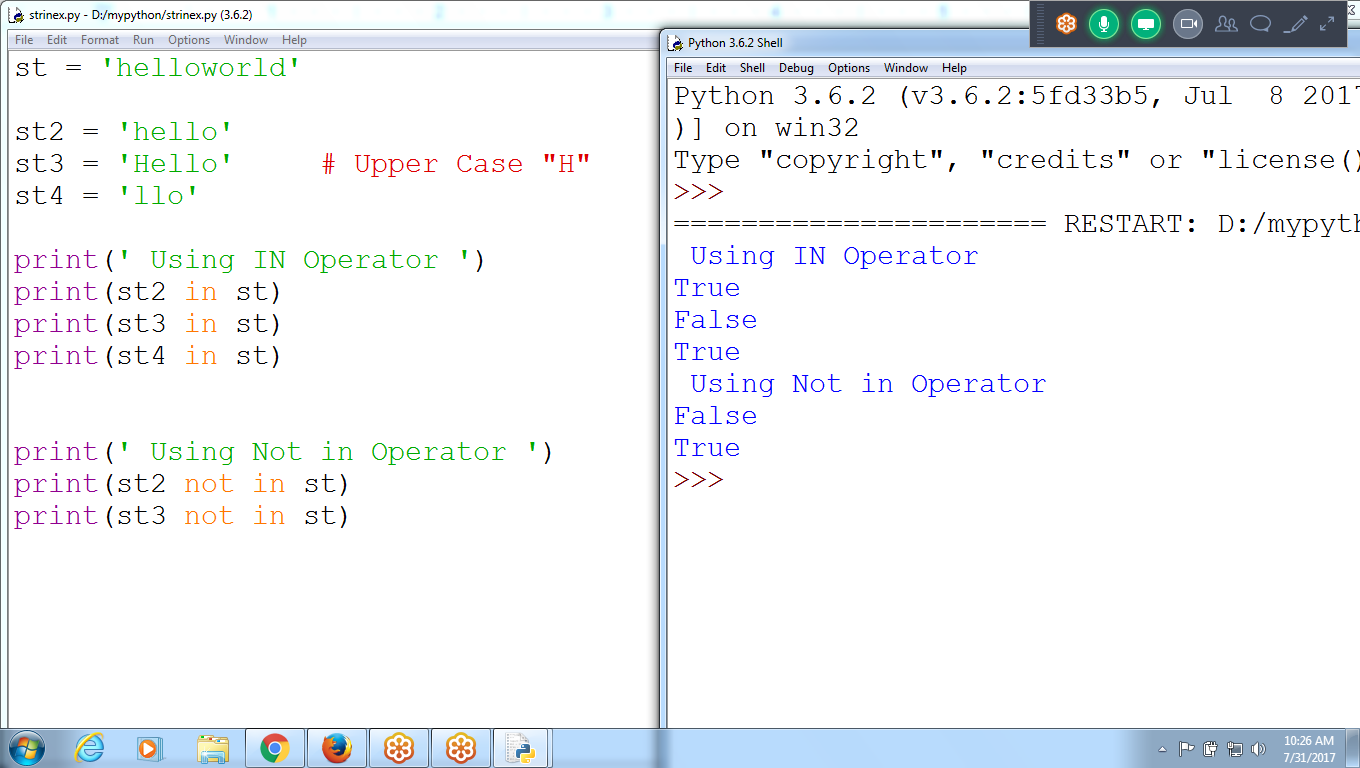
**print(st3 in st)**

**print(st4 in st)**

**print(' Using Not in Operator ')**

**print(st2 not in st)**

**print(st3 not in st)**

****

**Try program Without Using in and not in operator**

**fs = 'llo'**

**ss = 'helloworld'**

**print(fs in ss)**

**'''print(' WIthout using in and not in operator ')**

**i=0**

**while( i<len(fs)):**

**j=0**

**while(j<len(ss)):**

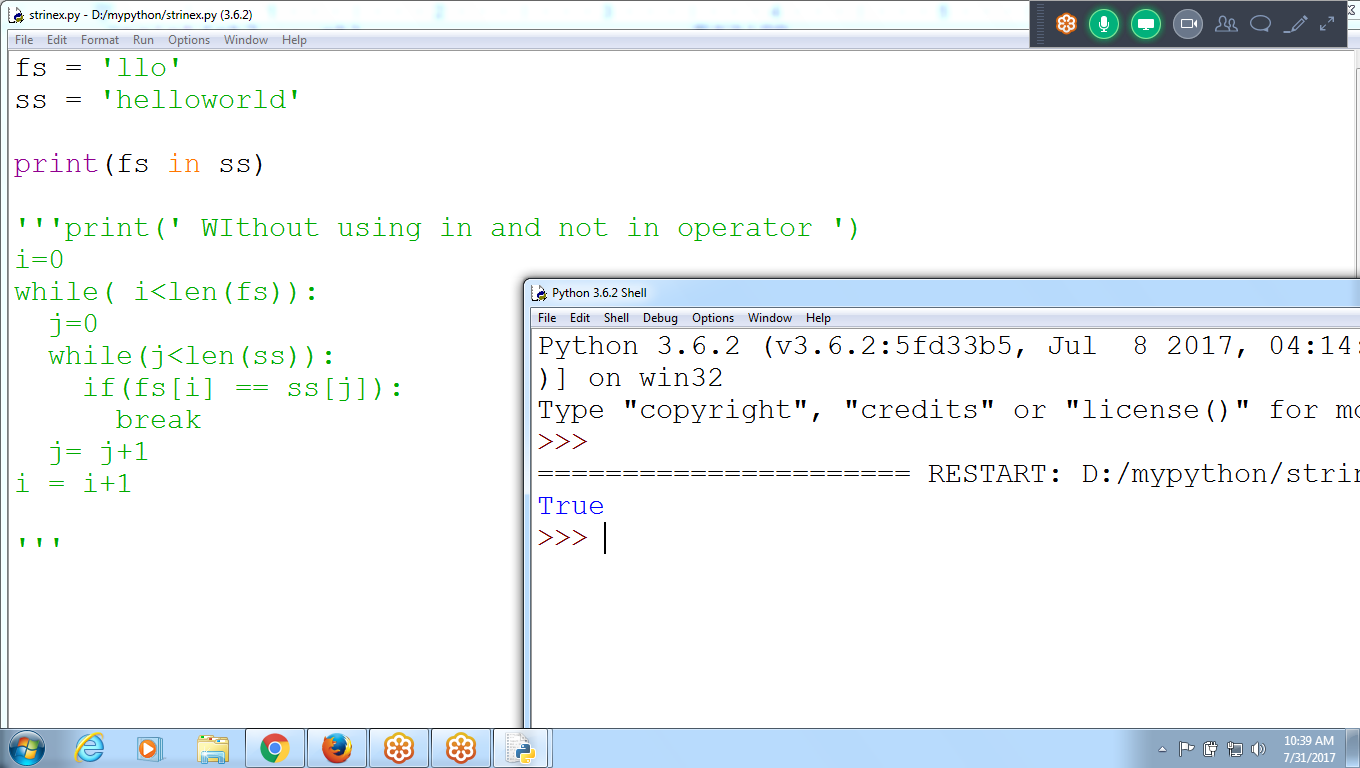
**if(fs[i] == ss[j]):**

**break**

**j= j+1**

**i = i+1**

**'''**

****

**>>> st**

**'Hello World123#'**

**>>> 'hello' in st**

**False**

**>>> 'Hello' in st**

**True**

**>>>**

**>>>**

**>>>**

**>>>**

**>>> 'Hello' not in st**

**False**

**>>>**

**# Using in and not in operators**

**# Input two string check whether existing or not**

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

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

if(fs **in** ss):

print(fs, ' Found in ss ')

else:

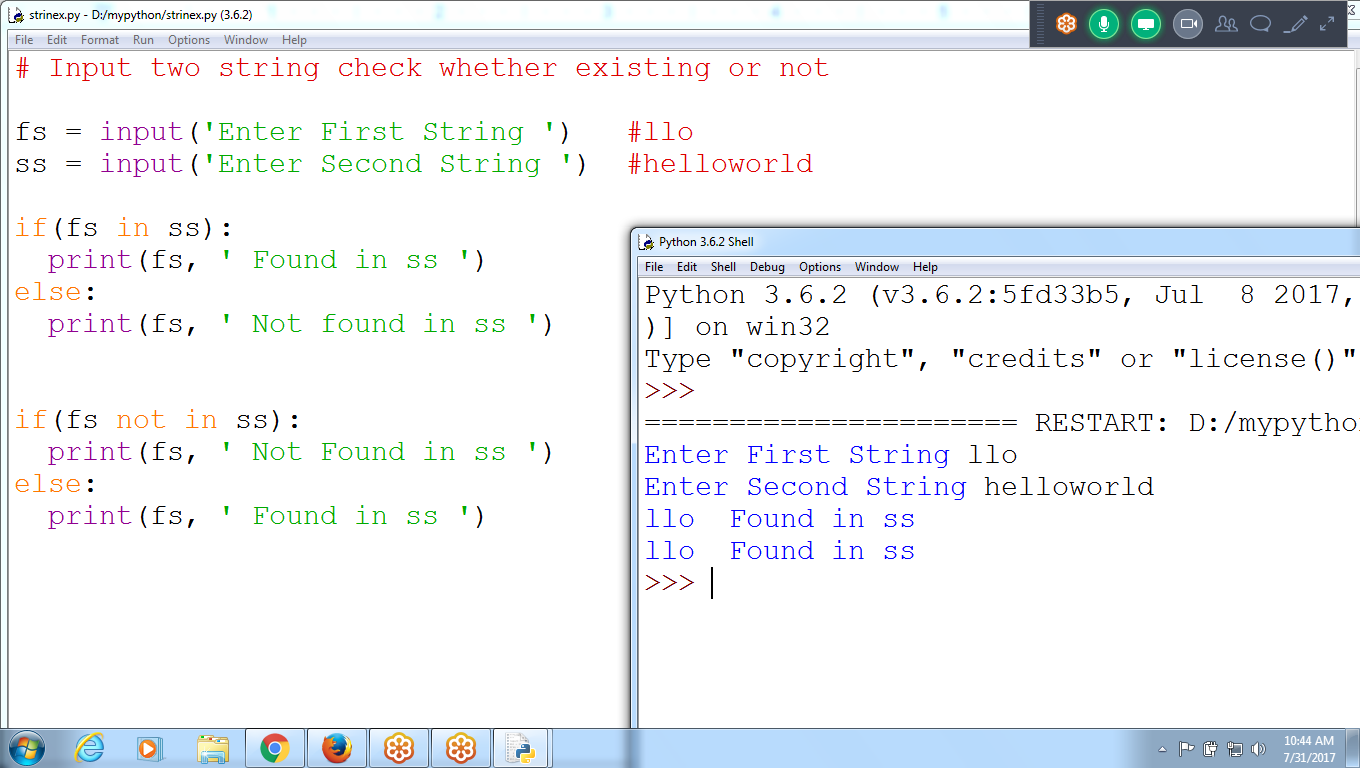
print(fs, ' Not found in ss ')

if(fs **not in** ss):

print(fs, ' Not Found in ss ')

else:

print(fs, ' Found in ss ')

****

# **String Methods**

* String methods analyze strings or create transformed string values
* if the string has at least one letter and all the letters are uppercase or lowercase

Several string methods that have names beginning with the word *is*. which returns a Boolean value

* isalpha() returns True if the string consists only of letters.
* isalnum() returns True if the string consists only of letters or numbers.
* isdecimal() returns True if the string consists only of numeric characters.
* isspace() returns True if the string consists only of spaces, tabs, and newlines.
* istitle() returns True if the string consists only of words that begin with an uppercase letter followed by only lowercase letters.

st = 'Helloworld'

st2 = 'Helloworld123'

print(' Checking for only alphabets ')

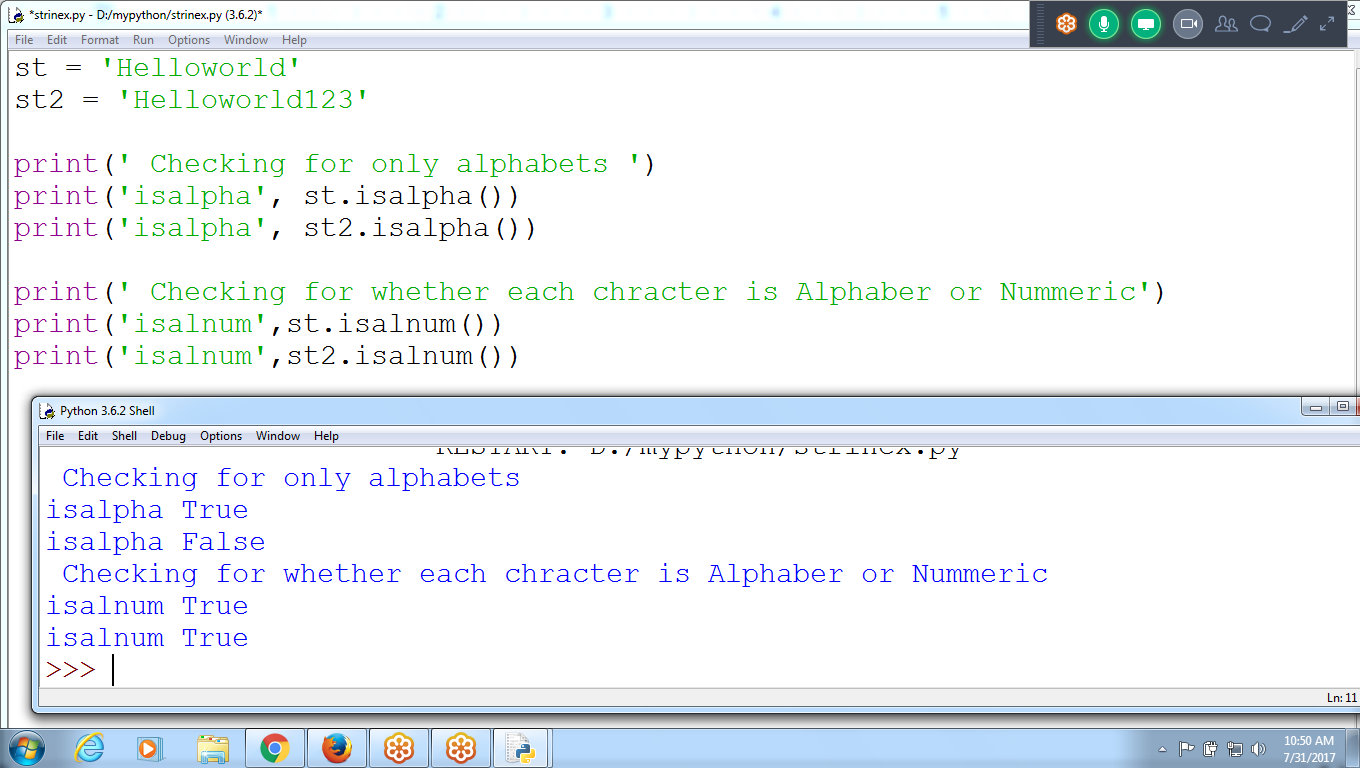
print('isalpha', st.isalpha())

print('isalpha', st2.isalpha())

print(' Checking for whether each chracter is Alphaber or Nummeric')

print('isalnum',st.isalnum())

print('isalnum',st2.isalnum())



#Checking for Decimal for DIGITS (base 10)

st = 'Helloworld'

st2 = 'Helloworld123'

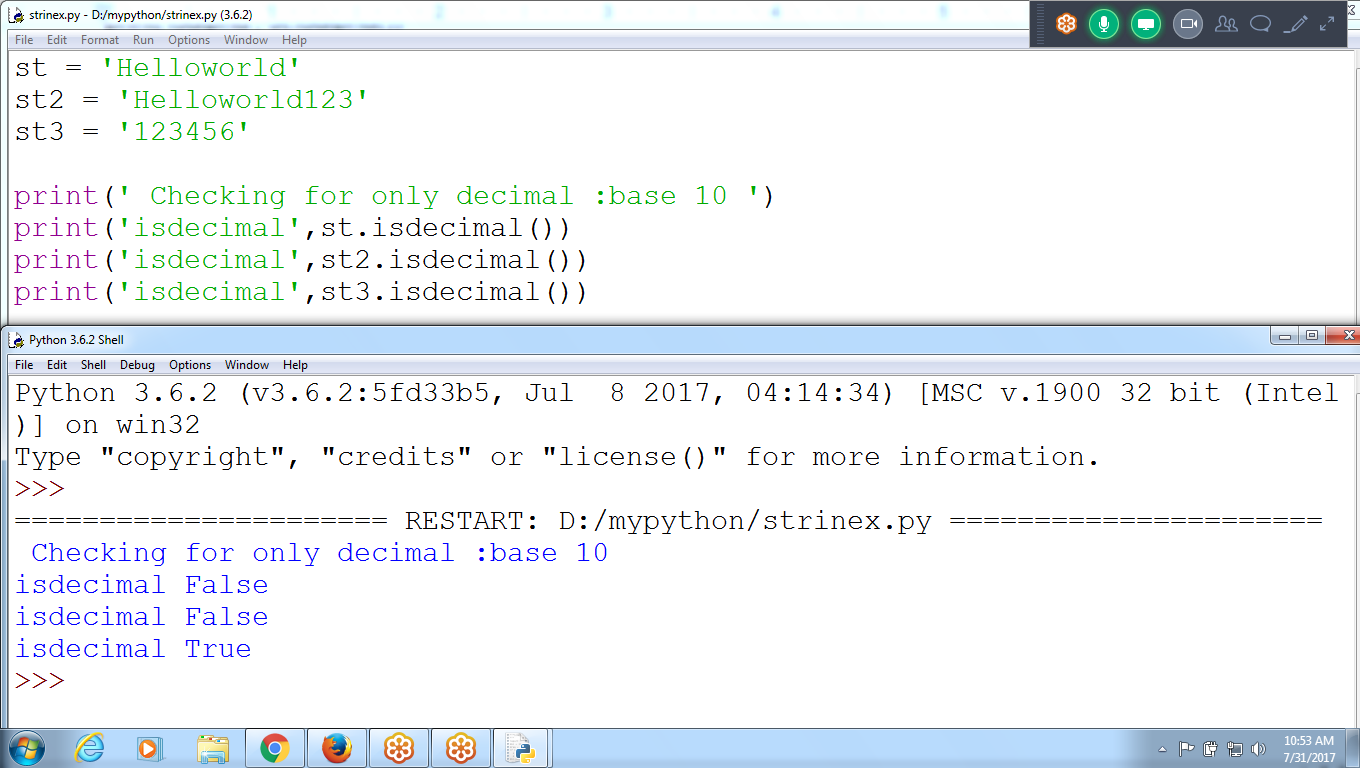
st3 = '123456'

print(' Checking for only decimal :base 10 ')

print('isdecimal',st.isdecimal())

print('isdecimal',st2.isdecimal())

print('isdecimal',st3.isdecimal())



**#Checking for space**

st = 'Hello\tworld'

st2 = 'Helloworld\n123'

st3 = '123 456'

st4 = ''

st5 = ' '

print(' Checking for space')

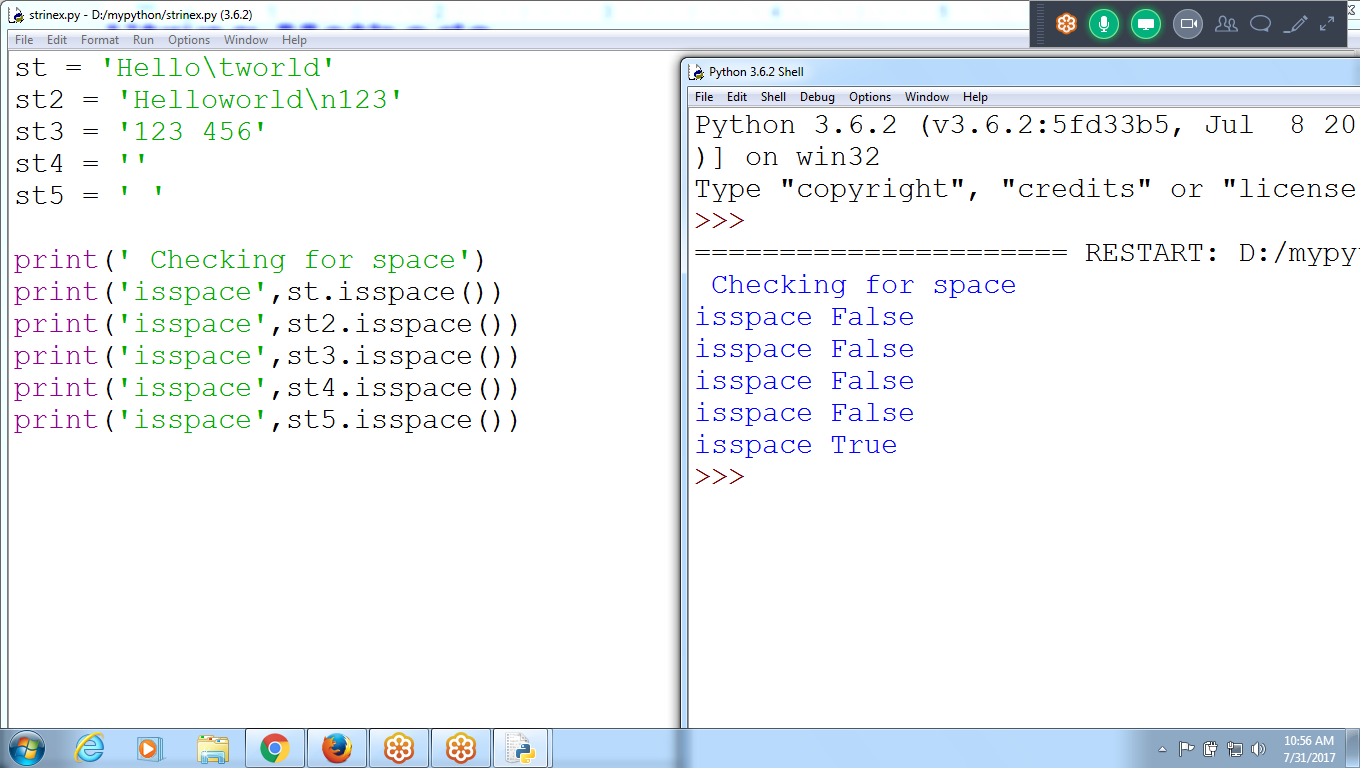
print('isspace',st.isspace())

print('isspace',st2.isspace())

print('isspace',st3.isspace())

print('isspace',st4.isspace())

print('isspace',st5.isspace())



**#Checking for cases**

st = 'helloworld'

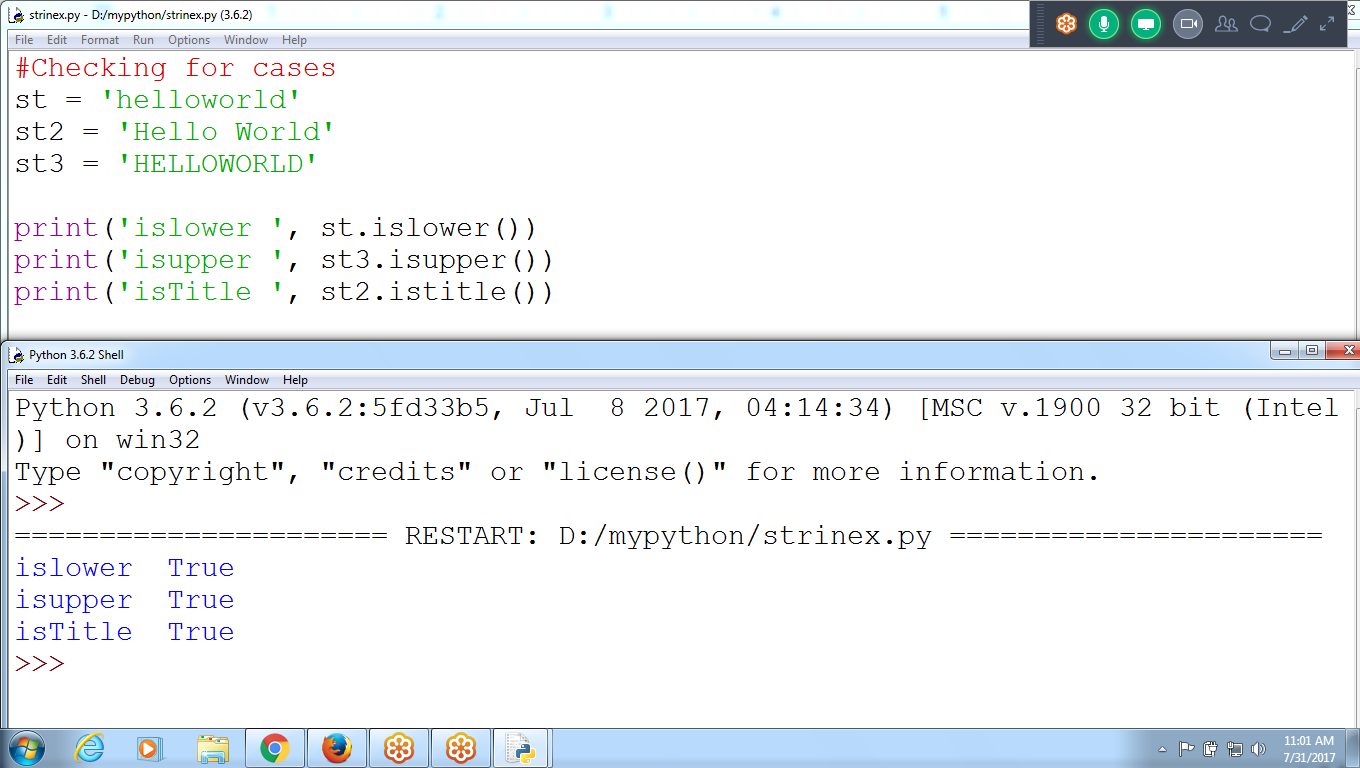
st2 = 'Hello World'

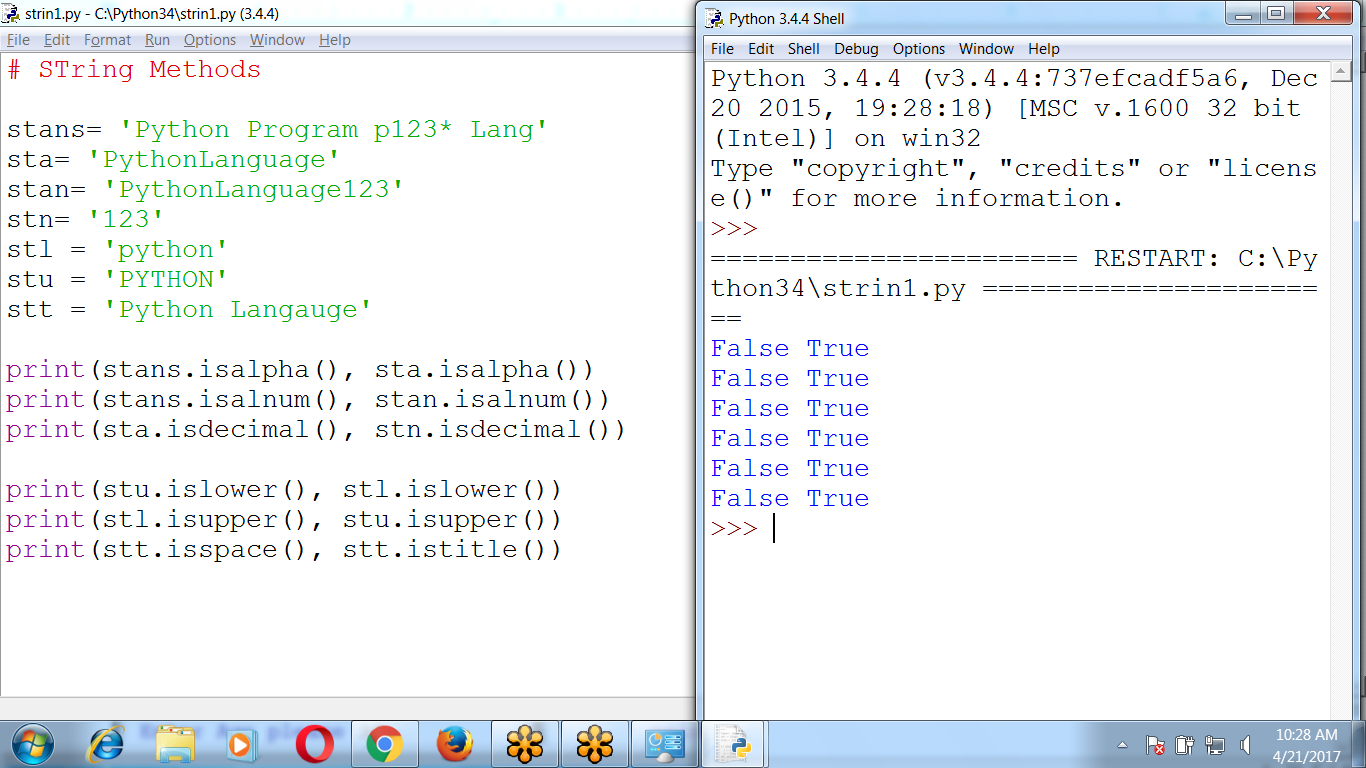
st3 = 'HELLOWORLD'

print('islower ', st.islower())

print('isupper ', st3.isupper())

print('isTitle ', st2.istitle()) # First Letter must be Upper case in Each word





# STring Methods

stans= 'Python Program p123\* Lang'

sta= 'PythonLanguage'

stan= 'PythonLanguage123'

stn= '123'

stl = 'python'

stu = 'PYTHON'

stt = 'Python Langauge'

print(stans.isalpha(), sta.isalpha())

print(stans.isalnum(), stan.isalnum())

print(sta.isdecimal(), stn.isdecimal())

print(stu.islower(), stl.islower())

print(stl.isupper(), stu.isupper())

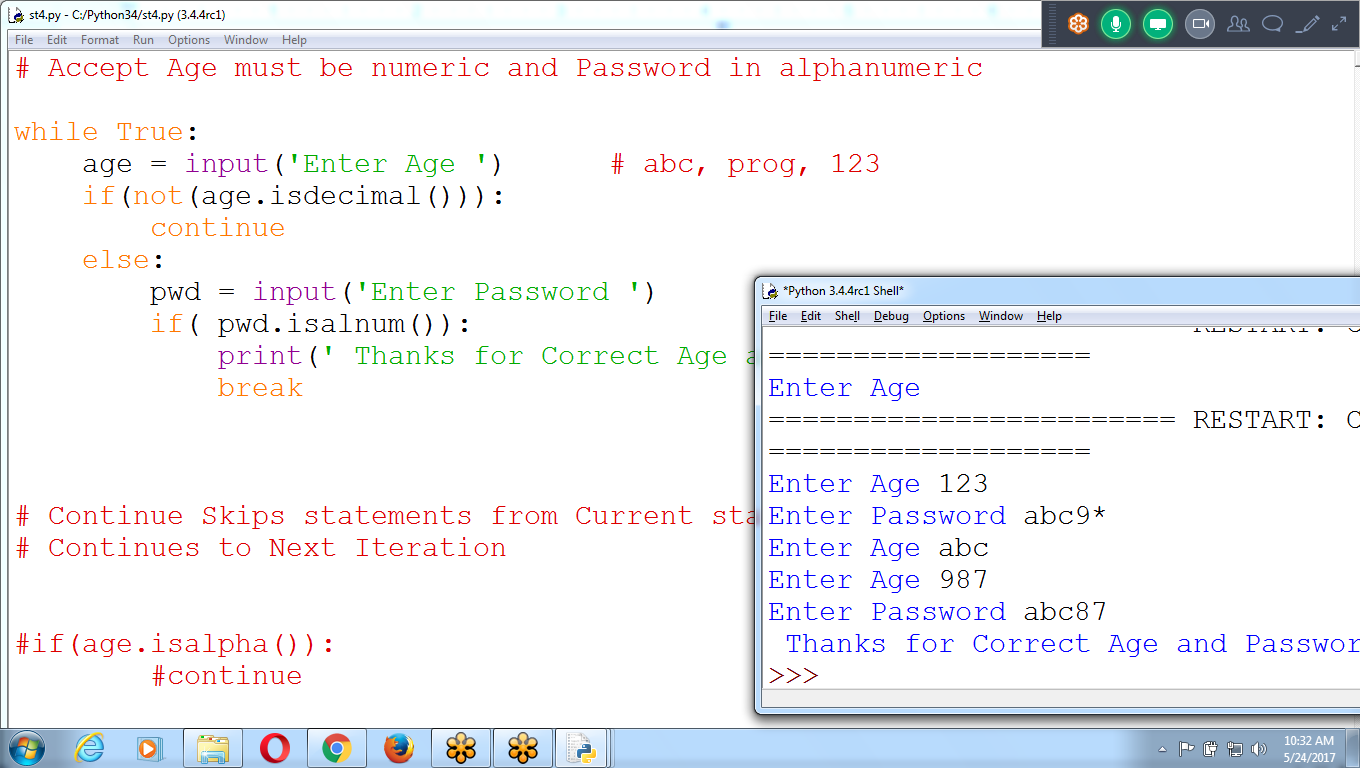
print(stt.isspace(), stt.istitle())

**program repeatedly asks users for their age and a password until they provide valid input**

**Age must be Numerics,**

**password must be Alphanumerics (alphabets and numerics)**

**Using String Functions, Continue and Break**

****

**# Continue Skips statements from Current statement to End of the loop and Continues to Next Iteration**

**# Break : Moves Control out of the current LOOP**

**#if(age.isalpha()):**

**#continue**

Python

Django : ONe Project

BDM :: 12K + 20% incentive :: 25 to 30k