PythonCourse_7_Strings

April 18, 2021

0.0.1 Strings

[9]:
```
## Representation

s1 = 'Jayashree' #single quotes
s2 = "Rekha" #double quotes
s3 = '''Srinivasan''' #triple quotes
print(s1)
print(s2)
print(s3)

#s4 = "Line1   # SyntaxError: EOL while scanning string literal
#Line2
#Line3"

# Triple quotes used for multiline strings
s4 = '''Line1
Line2
Line3'''
print(s4)
s4  #stored with \n characters
```

```
Jayashree
Rekha
Srinivasan
Line1
Line2
Line3
```

[9]:
```
'Line1\nLine2\nLine3'
```

[118]:
```
s = "Jayashree"
print('s',type(s))
print('s[0]',type(s[0])) #Each element in the string is also of type string
```

```
s <class 'str'>
s[0] <class 'str'>
```

[20]:
```
## Indexing

s1 = 'Jayashree'
print(s1[0])
print(s1[len(s1)-1])
#print(s1[15]) #IndexError: string index out of range
print(s1[-1]) #Negative indexing possible

s2 = '''Hello
World'''
print(s2[5]) #prints the \n character
s2[5] #\n character present in multiline strings
```

```
J
e
e
```

[20]:
```
'\n'
```

[23]:
```
## Storing Strings

s = "Jayashree"
print('s', s, id(s))
s = "Rekhu"  #Reference changes
print('s', s, id(s))

a = "Jayashree" #Python optimization- Small strings refernces are same
print('a', a, id(a))
```

```
s Jayashree 83744432
s Rekhu 84547696
a Jayashree 83744432
```

[26]:
```
## Strings are Immutable

s = "Jayashree"
print(s[0])

#Cannot change the contents in the memory-- Immutable unlike Lists
#s[0] = 5 # TypeError: 'str' object does not support item assignment
```

```
J
```

[31]:
```
## String Concatenation

s = 'Jayashree'
print('s', s, id(s))
```

```
s = s + 'Srinivasan' #new string created, therefore refernces different
print('s', s, id(s))

s += "Rekha" #using +=operator
print('s', s, id(s))

s = s * 2 #using * operator-> multiples can be created
print('s', s, id(s))

#s = s + 2 #TypeError: can only concatenate str (not "int") to str
s = s + str(2) #need to convert int to string, only string and string can be␣
 ↪concatenated in Python
print('s', s, id(s))
```

```
s Jayashree 83744432
s JayashreeSrinivasan 84583808
s JayashreeSrinivasanRekha 84591840
s JayashreeSrinivasanRekhaJayashreeSrinivasanRekha 84563984
s JayashreeSrinivasanRekhaJayashreeSrinivasanRekha2 84564992
```

[38]:
```
## Slicing of Strings

s = "ABCDE"
print(s[::]) #(start:stop:step) Default(0:len:1)
print(s[1::2])
print(s[::-1]) #negative indexing possible
print(s[4:1:-1]) #with negative step 4 to 1 possible
print(s[4:1:1])
s[4:1:1] #null since start cant be more than stop with positive step
```

```
ABCDE
BD
EDCBA
EDC
```

[38]:
```
''
```

[43]:
```
## Iterating Through Strings

s = "abc"
# Through Indices
for i in range(len(s)):
    print(s[i])

# Through Elements
for element in s:
    print(element)
```

```
a
b
c
a
b
c
```

[51]:
```
## Check Substring using in/not in

s = 'abcdefghij'
sub_s1 = 'cde' #Has to be continuous part
sub_s2 = 'fe' #Cannot be backwards
sub_s3 = 'ch' #Cannot be discontinuos
sub_s4 = 'j' #Can be just one character

if sub_s1 in s:
    print('s1', 'True')
else:
    print('s1','False')

if sub_s2 in s:
    print('s2', 'True')
else:
    print('s2','False')

if sub_s3  not in s:
    print('s3', 'False')
else:
    print('s3','True')

if sub_s4 not in s:
    print('s4', 'False')
else:
    print('s4','True')
```

```
s1 True
s2 False
s3 False
s4 True
```

[64]:
```
## Comparison Operator on Strings

#Comparison done based on ascii values
print('1',"Jay" == "Jay")
print('2',"jay" > "Jay")
print('3',"jay" <= "Jay")
print('4',"jay" != "Jay")
```

```
print('5',"Jay" <= "Jayashree") #if same chars, then considers length: Longer␣
 ↪length greater value
print('6',"jay" <= "Jayashree") #char by char comparison; here j > J : Length␣
 ↪does not matter
```

```
1 True
2 True
3 False
4 True
5 True
6 False
```

Operations on String

[79]:
```
## split

#splits the string by the specified delimiter and puts the elements into a list
s = "Christropher Nolan is the best director of all times"
print(s.split(' '))
print(s.split()) #default delimiter is space
print(s.split(' ',1)) #second arg specifies how many times to split
print(s.split(' ',3)) #second arg specifies how many times to split
print(s) #original string unaltered

print(id(s))
print(id(s.split()))
print(id(s.split(' ',1)))
print(id(s.split(' '))) #different for different lists

s1 = "Inception,Mememto,Prestige,Interstellar,Tenet"
print(s1.split(','))
```

```
['Christropher', 'Nolan', 'is', 'the', 'best', 'director', 'of', 'all', 'times']
['Christropher', 'Nolan', 'is', 'the', 'best', 'director', 'of', 'all', 'times']
['Christropher', 'Nolan is the best director of all times']
['Christropher', 'Nolan', 'is', 'the best director of all times']
Christropher Nolan is the best director of all times
84675360
84662208
84251840
84662336
['Inception', 'Mememto', 'Prestige', 'Interstellar', 'Tenet']
```

[90]:
```
## replace

#replaces a subtring in the string
s = "Christropher Nolan is the best director of all times"
print(s.replace('Christropher Nolan','Mani Ratnam')) #(a,b) replaces a with b
print(s) #original string unaltered
```

```
s = s.replace('Christropher Nolan','Mani Ratnam') #assignment needed
print(s)

a = "Hi Hi Hi! How are you?"
print(a.replace('Hi','Hey',1)) #third arg tells how many to replace
print(a.replace('H','Hello',10)) #If more given--> no error
print(a.replace('Jay','Hello')) #If substring not present--> no error
```

```
Mani Ratnam is the best director of all times
Christopher Nolan is the best director of all times
Mani Ratnam is the best director of all times
Hey Hi Hi! How are you?
Helloi Helloi Helloi! Helloow are you?
Hi Hi Hi! How are you?
```

[103]:
```
## find

#Finds the specified substring and returns the index
s = "I have not seen Dunkirk"
print(s.find('have'))
s = "Dunkirk, I have not seen Dunkirk"
print(s.find('Dunkirk',5,33)) #the range of indicies to be searched can be␣
 ↪specified
print(s.find('Dunkirk',5)) #if third arg not specified--> by default takes the␣
 ↪end of the string
print(s.find('Tenet',5)) #return -1 if substring not present
```

```
2
25
25
-1
```

[109]:
```
## upper and lower

# converts to the specified case
s = "Break free like in the Shawahank Redemption"
print(s.upper())
print(s) #original string unaltered
s = s.upper() #assignment needed
print(s)
s = s.lower()
print(s)
```

```
BREAK FREE LIKE IN THE SHAWAHANK REDEMPTION
Break free like in the Shawahank Redemption
BREAK FREE LIKE IN THE SHAWAHANK REDEMPTION
break free like in the shawahank redemption
```

[116]:
```
## startswith

#checks if the string starts with the substring specified
#returns Boolean value
s = "Time and space has no bounds"
print(s.startswith('Tim'))
print(s.startswith('me',2,6)) #the range of indicies to be searched can be␣
 ↪specified
print(s.startswith('me',2,3)) #has to map the entire substring in the range
print(s.startswith('me',2)) #if third arg not specified--> by default takes the␣
 ↪end of the string
```

```
True
True
False
True
```

Programs

[117]:
```
## Replace character in a string

def replace(str,char1,char2):
    newStr = "" #new string needs to be created as strings are immutable
                #we cannot change the element at a particular index in the␣
 ↪original string unlike in lists where we can
    for element in str:
        if element == char1:
            newStr += char2 #append sting with replaced char
        else:
            newStr += element #append with the original string char
    return newStr

str = "abahdkabjeabhayhjabja"
str = replace(str,'a','z')
print(str)
```

```
zbzhdkzbjezbhzyhjzbjz
```

[123]:
```
## Consonants Vowels Digits Special Characters Count"

def CharacterCount(str):
    v,c,d,s = 0,0,0,0 #assignemnt to every variable important
    for element in str:
        if (element >= 'a' and element <= 'z') or (element >= 'A' and element␣
 ↪<= 'Z'):
            element = element.lower() #converts all the chars to lowercase ;␣
 ↪this single char is also string type
            if element == 'a' or element == 'e' or element == 'i' or element ==␣
 ↪'o' or element == 'u':
```

```
                v += 1
            else:
                c += 1
        elif (element >= '0' and element <= '9'):
            d += 1
        else:
            s += 1
    return v,c,d,s

str = "abhsytgfglDFFGlSlfsLFf24DFl@rr3$$6GgktG   REF rgDGg ^Y 3r $$v 99(())"
v,c,d,s = CharacterCount(str)
print(v,c,d,s)
```

```
2 41 7 18
```