Handling Strings

- Strings in Python are identified as a contiguous set of characters represented in the quotation marks.
- Python allows either pair of single or double quotes.
- Subsets of strings can be taken using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the string and working their way from -1 to the end.
- The plus (+) sign is the string concatenation operator and the asterisk (*) is the repetition operator. For example-

Accessing Values in Strings

Python does not support a character type; these are treated as strings of length one, thus also considered a substring.
To access substrings, use the square brackets for slicing along with the index or indices to obtain your substring. For example −

```
var1 = 'Hello World!'
var2 = "Python Programming"
print "var1[0]: ", var1[0]
print "var2[1:5]: ", var2[1:5]
```

String Special Operators

Assume string variable **a** holds 'Hello' and variable **b** holds 'Python', then −

Operator	Description
+	Concatenation - Adds values on either side of the operator
*	Repetition - Creates new strings, concatenating multiple copies of the same string
[]	Slice - Gives the character from the given index
[:]	Range Slice - Gives the characters from the given range
In	Membership - Returns true if a character exists in the given string

not in	Membership - Returns true if a character does not exist in the given string
r/R	Raw String - Suppresses actual meaning of Escape characters. The syntax for raw strings is exactly the same as for normal strings with the exception of the raw string operator, the letter "r," which precedes the quotation marks. The "r" can be lowercase (r) or uppercase (R) and must be placed immediately preceding the first quote mark.
%	Format - Performs String formatting

. Run the following in Python shell.

a) Print the first three characters of a string.

```
>>> st=”Python”
st[0:4]
```

b) Prints the last three characters of a string.

```
>>> st=”Python”
>>> st[-3:]
'hon'
```

c) Write a string, replace its first letter.

```
>>> st="Python"
>>> 'J'+st[1:]
'Jython'
```

d) Write a string include two more letters after the first two characters.

```
>>> st="Python"
>>> st[0:2]+st[-2:]
'Pyon'
```

e) Save a sentence in a variable. Print every third position character from first.

```
>>>st=”he ate camel food”
>>>st[::3]
‘ha m o’
```

f) Save a sentence in a variable. Print every second position character from last.

```
>>>st=”he ate camel food”
>>>st[::-2]
‘do ea t h’
```

g) Save a sentence in a variable. Print it in reverse without using the function.

```
>>> s='he ate camel food'
>>> s[::-1]
'doof lemac eta eh'
```

h) Store a list of strings. Print it by joining all of them using “-<>-”.

```
>>> list=['hello','welcome','bye']
>>> "-<>-".join(list)
'hello-<>-welcome-<>-bye'
```

i) Store a character in a variable in such a way that it will be printed five times. (Do it in a single statement)

```
>>> s="="*5
>>> print(s)
=====
```

j) Mention width and print a string with center aligned.

```
>>> s='hello'
>>> s.center(10,' ')
'  hello   '
```

k) Count the number of occurrences of a string in another string.

```
>>> s='hello'
>>> s.count('l')
2
```

l) Print leftmost position of matched substring. What will be the output if the substring is not matched Check the result.

```
>>> s="Welcome to the world of wonder"
>>> x1="ld"
>>> s.find(x1,0,len(s))
18
>>> x="ly"
>>> s.find(x,0,len(s))
-1
```

m) Mention a string and substring which is present in the string. Now partition it.

```
>>> s="Welcome to the world of Python Jython"
>>> t="thon"
>>> s.partition(t)
('Welcome to the world of Py', 'thon', ' Jython')
>>> s='madam'
>>> t='ma'
>>> s.partition(t)
('', 'ma', 'dam')
```

n) Replace a substring by another substring in a string.

```
>>> s="Welcome to the world of Python Jython"
>>> t='on'
>>> u='xxyy'
>>> s.replace(t,u)
'Welcome to the world of Pythxxyy Jythxxyy'
>>> t1='x'
>>> s.replace(t1,u)
'Welcome to the world of Python Jython'
```

o) Split a sentence w.r.t. a substring mentioning both one and two parameters.

```
>>> s="Welcome to the world of Python Jython"
>>> t="thon"
>>> s.split(t)
['Welcome to the world of Pyth', ' Jyth', '']
>>> s.split(t,1)
['Welcome to the world of Pyth', ' Jython']
```

p) Do the above for thrice from start.

```
>>> s="Welcome to the world of Python Jython"
>>> x='o'
>>> u=' '
>>> s.split(x,3)
['Welc', 'me t', ' the w', 'rld of Python Jython']
```

q) Do the above for thrice from end.

```
>>> s="Welcome to the world of Python Jython"
>>> x='o'
>>> u=' '
>>> s.rsplit(x,3)
['Welcome to the world ', 'f Pyth', 'n Jyth', 'n']
```

r) Write a sentence having a name, date of birth, date of death separated by * without space. Split it and store it in a list. Split the dates and save it in two separate lists. Print as follows:
XYZ lived about 82 years
(Name and age are variables) (Dates are written as YYYY-MM-DD)

```
>>> record='Leo Tolstroy*1828-8-28*1910-11-20'
>>> fields=record.split("*")
>>> fields
['Leo Tolstroy', '1828-8-28', '1910-11-20']
>>> born=fields[1].split("-")
>>> born
['1828', '8', '28']
>>> died=fields[2].split("-")
>>> died
['1910', '11', '20']
>>> print("lived about",int(died[0])-int(born[0]),"years")
lived about 82 years
```

s) Using format specification print it as right aligned and left aligned

```
>>> s="When I step along the world I go"
>>> s.ljust(30,'x')
'When I step along the world I go'
>>> s.ljust(50,'x')
'When I step along the world I goxxxxxxxxxxxxxxxxxx'
>>> s.rjust(50,'x')
'xxxxxxxxxxxxxxxxxxWhen I step along the world I go'
```

t) Using “in” operator check whether a substring is present in the string.

```
>>> 'a' in 'banana'
True
>>> t='a'
>>> s='banana'
>>> t in s
True
```

v) Capitalize the first letter.

```
>>> s='hello'
>>> s.capitalize()
'Hello'
```

w) Check whether a tuple of substring are present in a string and substring slice. Provide at least five different instances.

```
>>> s="Welcome to the world of wonder"
>>> x=("me","ld","er")
>>> s.endswith(x,0,30)
True
>>> s.endswith(x,0,len(s))
True
>>> s.endswith(x,0,15)
False
>>> x1="ld"
>>> s.endswith(x1,0,len(s))
False
```

x) Split lines which are in triple quote. Also, split another lines which includes escape sequences.

```
>>> s='''welcome dear
how are you?
Where do you live?
Do you love Kolkata?'''
>>> s.splitlines()
['welcome dear', 'how are you?', 'Where do you live?', 'Do you love Kolkata?']
```

y) Check the starting substring of a string for different instances.

```
>>> s="Welcome to the world of Python Jython"
>>> t=('a',"W",'t')
>>> s.startswith(t)
True
```

1. Write a program which will print each character of a string.

```
sampleStr = "Hello!!"
print("**** Iterate over string using for loop****")
for elem in sampleStr:
    print(elem)
```

2. Write a program to count a particular letter in a string.

```
sampleStr = "Hello!!"
count=0
print("**** Iterate over string using for loop****")
for elem in sampleStr:
        if(elem==’l’)
    count=count+1
print(count)
```