BCA – 502: Python Programming

Rahul Kumar Singh

In today's Class we have discussed on Built-in String Methods of Python.

Built-in String Methods:-

Python includes the following built-in methods to manipulate strings −

1. String capitalize() Method:-

Python String **capitalize()** method returns a copy of the string with only its first character capitalized.

Syntax:-

```
str.capitalize()
```

Example:-

```
#!/usr/bin/python

str = "this is string example....wow!!!";

print "str.capitalize() : ", str.capitalize()
```

Output:-

```
str.capitalize() :  This is string example....wow!!!
```

2. String center() Method:-

Python string method center() returns centered in a string of length width. Padding is done using the specified fillchar. Default filler is a space.

Syntax

str.center(width[, fillchar])

Parameters

width – This is the total width of the string.

fillchar – This is the filler character.

Return Value

This method returns centered in a string of length width.

Example:-

The following example shows the usage of center() method.

```
#!/usr/bin/python

str = "this is string example....wow!!!"

print "str.center(40, 'a') : ", str.center(40, 'a')
```

Output:-

```
str.center(40, 'a') : aaaathis is string example....wow!!!aaaa
```

3. String count() Method:-

Python string method count() returns the number of occurrences of substring sub in the range [start, end]. Optional arguments start and end are interpreted as in slice notation.

Syntax:-

str.count(sub, start= 0,end=len(string))

Parameters

sub − This is the substring to be searched.

start − Search starts from this index. First character starts from 0 index. By default search starts from 0 index.

end − Search ends from this index. First character starts from 0 index. By default search ends at the last index.

Return Value

Centered in a string of length width.

Example:-

```
#!/usr/bin/python

str = "this is string example....wow!!!";

sub = "i";
```

```
print "str.count(sub) : ", str.count(sub)

print "str.count(sub, 4, 40) : ", str.count(sub, 4, 40)

sub = "wow";

print "str.count(sub) : ", str.count(sub)
```

Output:-

```
str.count(sub) :  3

str.count(sub, 4, 40) :  2

str.count(sub) :  1
```

4. String decode() Method

Python string method decode() decodes the string using the codec registered for encoding. It defaults to the default string encoding.

Syntax

```
Str.decode(encoding='UTF-8',errors='strict')
```

Parameters

encoding – This is the encodings to be used. For a list of all encoding schemes please visit: Standard Encodings.

errors – This may be given to set a different error handling scheme. The default for errors is 'strict', meaning that encoding errors raise a UnicodeError. Other possible values are 'ignore', 'replace', 'xmlcharrefreplace',

'backslashreplace' and any other name registered via codecs.register_error().

Return Value

Decoded string.

Example:-

```
#!/usr/bin/python

Str = "this is string example....wow!!!";

Str = Str.encode('base64','strict');

print "Encoded String: " + Str

print "Decoded String: " + Str.decode('base64','strict')
```

Output:-

```
Encoded String:
dGhpcyBpcyBzdHJpbmcgZXhhbXBsZS4uLi53b3chISE=

Decoded String: this is string example....wow!!!
```

5. String encode() Method:-

Python string method encode() returns an encoded version of the string. Default encoding is the current default string encoding. The errors may be given to set a

different error handling scheme.

Syntax

str.encode(encoding='UTF-8',errors='strict')

Parameters

encoding − This is the encodings to be used. For a list of all encoding schemes please visit: Standard Encodings.

errors − This may be given to set a different error handling scheme. The default for errors is 'strict', meaning that encoding errors raise a UnicodeError. Other possible values are 'ignore', 'replace', 'xmlcharrefreplace', 'backslashreplace' and any other name registered via codecs.register_error().

Return Value

Encoded string.

Example

```
#!/usr/bin/python

str = "this is string example....wow!!!";

print "Encoded String: " + str.encode('base64','strict')
```

Output

```
Encoded String:
dGhpcyBpcyBzdHJpbmcgZXhhbXBsZS4uLi53b3chISE=
```

6. String format() Method:-

To make sure a string will display as expected, we can format the result with the format() method.

The format() method allows you to format selected parts of a string.

Sometimes there are parts of a text that you do not control, maybe they come from a database, or user input?

To control such values, add placeholders (curly brackets {}) in the text, and run the values through the format() method:

Example:-

```
#!/usr/bin/python

price = 49

txt = "The price is {} dollars"

print(txt.format(price))
```

Output:-

```
The price is 49 dollars
```

You can add parameters inside the curly brackets to specify how to convert the value:

Example:-

```
#!/usr/bin/python

price = 49

txt = "The price is {:.2f} dollars"

print(txt.format(price))
```

Output:-

```
The price is 49.00 dollars
```

If you want to use more values, just add more values to the format() method:

Example:-

```
#!/usr/bin/python

quantity = 3

itemno = 567

price = 49

myorder = "I want {} pieces of item number {} for {:.2f} dollars."

print(myorder.format(quantity, itemno, price))
```

Output:-

```
I want 3 pieces of item number 567 for 49.00 dollars.
```

You can use index numbers (a number inside the curly brackets {0}) to be sure the values are placed in the correct placeholders:

Example:-

```
#!/usr/bin/python

quantity = 3

itemno = 567

price = 49

myorder = "I want {0} pieces of item number {1} for {2:.2f} dollars."

print(myorder.format(quantity, itemno, price))
```

Output:-

I want 3 pieces of item number 567 for 49.00 dollars.

Also, if you want to refer to the same value more than once, use the index number:

Example:-

```
#!/usr/bin/python

age = 29

name = "Rahul"

txt = "His name is {1}. {1} is {0} years old."
```

```
print(txt.format(age, name))
```

Output:-

His name is Rahul. Rahul is 29 years old.

You can also use named indexes by entering a name inside the curly brackets {carname}, but then you must use names when you pass the parameter values txt.format(carname = "Ford"):

Example:-

```
myorder = "I have a {carname} Company Car, It is a {model}."

print(myorder.format(carname = "Hyundai", model = "Creta"))
```

Output:-

I have a Hyundai Company Car, it is a Creta.

7. String max() Method:-

Python string method max() returns the max alphabetical character from the string str.

Syntax-

```
max(str)
```

Parameters:-

str – This is the string from which max alphabetical character needs to be returned.

Return Value:-

This method returns the max alphabetical character from the string str.

Example:-

```
#!/usr/bin/python

str = "this is really a string example....wow!!!";

print "Max character: " + max(str)

str = "this is a string example....wow!!!";

print "Max character: " + max(str)
```

When we run above program, it produces following result–

```
Max character: y

Max character: x
```

8. String min() Method:-

Python string method min() returns the min alphabetical character from the string str.

Syntax:-

min(str)

Parameters

str − This is the string from which min alphabetical character needs to be returned.

Return Value

This method returns the min alphabetical character from the string str.

Example:-

```
#!/usr/bin/python

str = "this-is-real-string-example....wow!!!";

print "Min character: " + min(str)

str = "this-is-a-string-example....wow!!!";

print "Min character: " + min(str)
```

When we run above program, it produces following result−

```
Min character: !

Min character: !
```

9. String swapcase() Method:-

Python string method swapcase() returns a copy of the string in which all the case-based characters have had their case swapped.

Syntax-

str.swapcase();

Example:-

The following example shows the usage of swapcase() method.

```
#!/usr/bin/python

str = "this is string example....wow!!!";

print str.swapcase()

str = "THIS IS STRING EXAMPLE....WOW!!!";

print str.swapcase()
```

When we run above program, it produces following result–

```
THIS IS STRING EXAMPLE....WOW!!!

this is string example....wow!!!
```

10. String lower() Method:-

Python string method lower() returns a copy of the string in which all case-based characters have been lowercased.

Syntax:-

str.lower()

Example:-

#!/usr/bin/python

str = "THIS IS STRING EXAMPLE....WOW!!!";

print str.lower()

When we run above program, it produces following result–

this is string example....wow!!!

11. String upper() Method:-

Python string method upper() returns a copy of the string in which all case-based characters have been uppercased.

Syntax:-

str.upper()

Example:-

```
#!/usr/bin/python

str = "this is string example....wow!!!";

print "str.capitalize() : ", str.upper()
```

When we run above program, it produces following result–

```
str.capitalize() :  THIS IS STRING EXAMPLE....WOW!!!
```

12. String title() Method:-

Python string method title() returns a copy of the string in which first characters of all the words are capitalized.

Syntax:-

```
str.title();
```

Example:-

```
#!/usr/bin/python

str = "this is string example....wow!!!";

print str.title()
```

When we run above program, it produces following result–

```
This Is String Example....Wow!!!
```