Python, Day 3.5: String Methods

Andrew Bydlon

January 8, 2019

String Methods: Status

In general, an object has many **method functions** associated with it. Here is a list of a few functions which test a string for a given property. Throughout, it is assumed that the string is named MyString:

1. MyString.isalnum(): Tests whether the string is alphanumeric (composed of letters and numbers).
2. MyString.isalpha(): Tests whether the string is alphabets (composed of letters).
3. MyString.isdigit(): Tests whether the string is numeric (composed of numbers).
4. MyString.islower(): Tests whether the string is all lower case.
5. MyString.isupper(): Tests whether the string is all upper case.
6. MyString.isspace(): Tests whether the string is all whitespace.

These can help to make sure the user is giving the correct type of input. Note that each returns a boolean variable.

Examples

Example (Properties)

```
> > > string1 = "!@#()"
> > > string1.isalnum()   # can also be done with "!@#()".isalnum()
False
> > > string2 = "asdflgjrug1235"
> > > string2.isalnum()
True
> > > string2.isalpha()
False
> > > string2.isdigit()
False
> > > string2.islower()
True
```

String Methods: Searching/Replacing

There are also many ways to search/replace a string.

1. MyString.endswith(test): Tests whether or not the string ends with a given substring test.
2. MyString.startswith(test): Tests whether or not the string starts with a given substring test.

   These commands again return a boolean variable.

3. MyString.find(test): Finds the first occurrence of a substring test.
4. MyString.rfind(test): Finds the last occurrence of a substring test.
5. MyString.count(test): Counts the number of occurrences of a given substring test.

   These commands return a integer variable.

6. MyString.replace(old,new): replaces the substring old with new.

   This command return the modified string.

Examples

Example (Find/Replace)

```
> > > string1 = "A long time ago in a galaxy far, far away..."
> > > string1.endswith("away")
False
> > > string1.startswith("A long time")
True
> > > string1.find("far")
28
> > > string1.rfind("far")
33
> > > string1.count("far")
2
> > > string2 = string1.replace("galaxy", "multiverse")
'A long time ago in a multiverse far, far away...'
```

String Methods: Conversion

Here are a few nice tricks to improve a string MyString.

1. MyString.capitalize(): Capitalizes the first character of the string.
2. MyString.title(): Capitalizes the first word of every word.
3. MyString.lower(): Makes the entire string lower case.
4. MyString.upper(): Makes the entire string upper case
5. MyString.swapcase(): Switches upper and lower case letters
6. MyString.strip(chars): Removes given characters from the beginning and the end of the string. If left blank, remove white space.

Note these commands return the string, but do NOT modify the original string! You would need to redeclare the variable:

MyString = MyString.upper()

Examples

Example (Conversion)

```
> > > string1 = " Hello NYC! "
> > > string1.lower()
'    hello nyc!    '
> > > string1.upper()
'    HELLO NYC!    '
> > > string2 = "@!@!@! why is i So bad at the typing? @!@!@!"
> > > string2.title()
'@!@!@! Why Is I So Bad At The Typing? @!@!@!'
> > > string2.strip("@!")
' why is i So bad at the typing? '
> > > string2.swapcase()
'@!@!!@ WHY IS I sO BAD AT THE TYPING? @!!!!@'
> > > string2.strip("@!").strip().title() # You can combine methods
'Why Is I So Bad At The Typing?'
```

Final formatting tips

You can also use the red .center(), red .ljust(), or red .rjust() to justify your text appropriately.

Example (Justification)

```
> > > string = "William's College"
> > > string.center(25)
"    William's College    "
> > > string.ljust(20)
"William's College    "
> > > string.rjust(20)
"    William's College"
```

Assignment 5

Write a Python program to change a given user input string *MyStr* to a string where the first and last characters have been exchanged. Print this string to the user.

Then take 2 new strings (say *Str1* and *Str2*) from the user. Replace every occurrence (using the unmodified *MyStr*) of *Str1* with *Str2* **except** the first occurrence.

Again print the result to the user.
You do not need to use my variable names. They simply make the explanation of the goal easier.

Don't forget to submit the .py file to GLOW.