Python Strings

Thota Naveen Babu

Note :- follow me on linkedin & Github for questions on python strings

1.0 Introduction

```
    1) In python strings are represented within the single quotation marks (or)
double quotation marks.
    2) 'hello' is the same as "hello".
    3) You can display a string literal with the print() function.
```

1.1 Assign String to a variable

In [1]:

```
#Assigning a string to a variable is done with the variable name followed
# by an equal aign and the string.

s1 = 'Naveen'
print(s1,type(s1))
```

```
Naveen <class 'str'>
```

```
1.2 Multi-line String
```

In [2]:

```
s2 = """
     I am in Hyderabad,
     I want to become a data scientist,
     I am from Andhra Pradesh.
     """
print(s2,type(s2))
```

```
     I am in Hyderabad,
     I want to become a data scientist,
     I am from Andhra Pradesh.
      <class 'str'>
```

1.3 Looping through a string

In [3]:

```
for characters in s1:
    print(characters, end=" ")
```

```
N a v e e n
```

1.4 String Length

In [8]:

```
print(len(s1))
print(len(s2))
```

```
6
101
```

1.5 in and not Keyword

```
   1) To check if a certain character is present in a string, We use the keywor
d called 'in'
   2) to check if a certain character is NOT present in a string, we use the ke
yword called 'not in'
```

In [10]:

```
print('Andhra' in s2)
```

```
True
```

In [13]:

```
print('mumbai' not in s2)
```

```
True
```

1.6 if and if not

In [5]:

```
s2 = """
     I am in Hyderabad,
     I want to become a data scientist,
     I am from Andhra Pradesh.
     """
```

In [6]:

```
def word_checker(in_str,word):
    if word in in_str:
        return "Yes, {} is present ingiven string.".format(word)
    elif word not in in_str:
        return "No, {} is not present in given string.".format(word)

word_checker(s2, 'Andhra')
```

Out[6]:

```
'Yes, Andhra is present ingiven string.'
```

In [8]:

```
word_checker (s2, 'kerala')
```

Out[8]:

```
'No, kerala is not present in given string.'
```

2.0 String Slicing

1) You can return a range of characters by using the slice index.
2) Specify the start index,separated by a colon, to return a part of the string.
3) Indexing start with Zero.

In [9]:

```
s3 = "my name is Naveen"
s3[2:10]
```

Out[9]:

```
' name is'
```

2.1 Slicing from start

1) By leaving out the start index, the range will start at first character.

In [10]:

```
b = "Thota Naveen"
print(b[:7])
```

```
Thota N
```

2.2 Slice to end

1) By leaving out the end index, the range will go to the end.

In [11]:

```
b = "Thota Naveen"
print(b[2:])
```

```
ota Naveen
```

2.3 Step

1) After start and end index we can also specify step, by default stepvalue is one.

In [12]:

```
b = "Thota Naveen"
print(b[2:9:2])
```

```
oaNv
```

2.4 Negative Indexing

1) Uses negative indexes to start the slice from the end of the string.

In [14]:

```
b = "Thota Naveen"
print(b[-1:-8:-1])
```

```
neevaN
```

In [15]:

```
c = "1234567890"
print(c[::-1])
```

```
0987654321
```

In [17]:

```
c = "1234567890"
print(c[-3::-1])
```

```
87654321
```

In [25]:

```
c[-10:-1]
```

Out[25]:

```
'123456789'
```

In [33]:

```
#Note Another way to get whole string using negative index use below mentioned code

c[:-1]
```

Out[33]:

```
'123456789'
```

3.0 Modifying Strings

1) Python has a set of built-in methods that you can use on strings.

3.1 Upper case and lower case

In [1]:

```
# The upper() method returns the string in upper case

n = "thota naveen"
print(n.upper())
```

```
THOTA NAVEEN
```

In [2]:

```
# the lower() method returns the string in lower case

m = "THOTA NAVEEN"
print(n.lower())
```

```
thota naveen
```

3.2 Remove Whitespace

There are three types of strips, based on our need we can select

i) lstrip :- left most space will be removed inside the string.
ii) rstrip :- right most space will be removed inside the string.
iii) strip :- by using this we can remove both left most and right most space inside the string.

In [11]:

```
# lstrip

n1 = "     Naveen"

# before applying lstrip, length is
print(len(n1))
```

```
11
```

In [14]:

```
print(n1.lstrip())
# after applying lstrip, length is
print(len(n1.lstrip()))
```

```
Naveen
6
```

In [9]:

```
# rstrip

n2 = "Naveen     "

# before applying rstrip, length is
print(len(n2))

print(n2.rstrip())

# after applying rstrip, length is
print(len(n2.rstrip()))
```

```
11
Naveen
6
```

In [16]:

```
# strip

n3 = "    Naveen     "
print(len(n3))

print(n3.strip())

# after applying strip, length is
print(len(n3.strip()))
```

```
15
Naveen
6
```

3.3 Replace String

1) The replace() method replaces a string with another string.

In [17]:

```
m1 = "Hello Naveen!"
print(m1.replace("e","#"))
```

```
H#llo Nav##n!
```

3.4 Split String

1) The split() method returns a list by spliting the string into substrings if it finds instances of the specified separator.

In [24]:

```
m2 = "Hello,My,Name,Is,Naveen"
print(m2.split(","))
```

```
['Hello', 'My', 'Name', 'Is', 'Naveen']
```

Note:- join() method can be used to get the string without separator after splitting.

4.0 Concatenate Strings

1) We can use '+' operator to concatenate two strings.

In [31]:

```
first_name = "Naveen"
last_name = "Thota"
Name = first_name+last_name
print(Name)
```

```
NaveenThota
```

In [33]:

```
# To add space between them

Name = first_name +" "+last_name

print(Name)
```

```
Naveen Thota
```

5.0 String format

1) As we learned in the python variables chapter, we cannot combine strings a
nd numbers.
2) But we can combine the strings and numbers by using the format() method.
3) The format() method takes the passed arguments, formats them and places th
em in the string where the
placeholders are

In [39]:

```
age = 22
txt = "My name is Naveen, and i am {}"
print (txt.format(age))
```

```
My name is Naveen, and i am 22
```

4) The format() method takes unlimited number of arguments, and are placed int
o the respective placeholders.

In [35]:

```
quantity = 3
itemno = 567
price = 49.95
myorder = "I want {} pieces of item {} for {} dollars."
print(myorder.format(quantity, itemno, price))
```

```
I want 3 pieces of item 567 for 49.95 dollars.
```

5) You can use index numbers {0} to be sure the arguments are placed in the cor
rect placeholders.

In [36]:

```
quantity = 3
itemno = 567
price = 49.95
myorder = "I want to pay {2} dollars for {0} pieces of item {1}."
print(myorder.format(quantity, itemno, price))
```

```
I want to pay 49.95 dollars for 3 pieces of item 567.
```

5.1 f-String

1) To create an f-string, prefix the string with the letter " f ". The string itself can be formatted in such the same way that you would do with str.format().

In [38]:

```
first_name1="Naveen"
last_name1="Thota"
age1=22
print(f"My name is {first_name1} {last_name1} and I am {age1} years old. ")
```

```
My name is Naveen Thota and I am 22 years old.
```

6.0 Escape characters

1) To insert characters that are illegal in a string, use an escape character.
2) An escape character is a backslash \ followed by the character you want to insert.

i) \' or \" Single or double quote.
ii) \\ Backslash
iii) \n new line
iv) \t Tab
v) \b Backspace

In [40]:

```
s = 'It\'s my right'
print(s)
```

```
It's my right
```

In [41]:

```
date = '01\\09\\1005'
print(date)
```

```
01\09\1005
```

In [42]:

```
multiline = "My name is Naveen.\nMy age is 22.\nI live in India."
print(multiline)
```

```
My name is Naveen.
My age is 22.
I live in India.
```

In [43]:

```
tab = "Hello\tWorld\t!"
print(tab)
```

```
Hello	World	!
```

In [46]:

```
backspace = "Hello \bWorld \b!"
print(backspace)
```

```
HelloWorld!
```

7.0 Strings

7.1 Capitalize()

1) Converts the first character of each sentence into upper case.

In [47]:

```
s5 = "my name is Naveen"
s5.capitalize()
```

Out[47]:

```
'My name is naveen'
```

7.2 Center()

```
  1) The center() method will center align the string, using a specified charac
ter (space is default) as the fill character.
  2) Syntax :- string.center(length, character)
```

In [51]:

```
txt3 = "Naveen"

x6 = txt3.center(20, "$")

print(x6)
```

```
$$$$$$$Naveen$$$$$$$
```

In [52]:

```
txt4 = "Naveen"

x7 = txt4.center(20,)

print(x7)
```

```
       Naveen
```

7.3 Count()

```
  1) Return the number of times the specified value appears in the string.
  2) Syntax: string.count(value, start, end)
  3) start: Optional. An Integer. The position to start the search. Default is
0.
  4) end: Optional. An Integer. The position to end the search. Default is the
end of the
     string.
```

In [ ]:

```
# s5='aabbcccdh'
s5.count('c')
```

7.4 Endswith() and Startswith()

```
  1) Returns true if the string ends with the specified value.
  2) Returns true if the string starts with the specified value.
```

In [54]:

```
s7='Are you going to school?'
s7.endswith('?')
```

Out[54]:

```
True
```

In [56]:

```
s7.startswith('A')
```

Out[56]:

```
True
```

7.5 Index()

1) The index() method finds the first occurrence of the specified value.
2) The index() method raises an exception if the value is not found.
3) The index() method is almost the same as the find() method, the only difference is that the find() method returns -1 if the value is not found. (See example below)
4) Syntax :- string.index(value, start, end)

In [57]:

```
s7='Naveenthota'
s7.index('t')
```

Out[57]:

```
6
```

7.6 isalnum()

```
        1) The isalnum() method returns True if all the characters are alphanumeric, meaning alphabet letter (a-z) and
            numbers (0-9).
```

In [58]:

```
s11 = "Naveen01"
s11.isalnum()
```

Out[58]:

```
True
```

7.7 isalpha()

1) The isalpha() method returns True if all the characters are alphabet letters (a-z).

In [59]:

```
s11 = "Naveen01"
s11.isalnum()
```

Out[59]:

```
True
```

7.8 isdigit()

1) The isalpha() method returns True if all the characters are alphabet letters (a-z).

In [60]:

```
s12 = "12345"
s12.isdigit()
```

Out[60]:

```
True
```

7.9 swapcase()

1) The swapcase() method returns a string where all the upper case letters are lower case and vice versa.
2) Syntax :- string.swapcase()

In [63]:

```
n9 = "nAVEEN"
n9.swapcase()
```

Out[63]:

```
'Naveen'
```

7.10 title()

1) Converts the first character of each word in a sentence to upper case.

In [64]:

```
n10 = "andhra pradesh is located in india"
n10.title()
```

Out[64]:

```
'Andhra Pradesh Is Located In India'
```

In [ ]: