Unit 4 - String

By : Bhavika Vaghela
PICA- BCA & IMCA
Parul University

What is String?

- A string is a **sequence** of characters. You can access the characters one at a time with the bracket operator.
- strings can be created by enclosing the character or the sequence of characters in the quotes. Python allows us to use single quotes, double quotes, or triple quotes to create the string.
- A character is simply a symbol. For example, the English language has 26 characters.
- In Python, a string is a sequence of Unicode characters. Unicode was introduced to include every character in all languages and bring uniformity in encoding.

How to declare or create string?

- String can be any thing which is declare using single quote, double quote or triple quote.

Message=‘hello’

Message=“hello”

Message=“””hello”””

Message=‘12’

- To check data type of it type() in built function is used.

How to quote some word into string using double quote?

Message = ‘this is very “beautiful” ’

Message = ‘”this” “is” “beautiful” ‘

How to access element or character of string?

- To access element or single character from string need to pass its index value inside [ ] bracket.
- Index value of string start by 0.
- We can also access it by passing negative index value.

e.g. print(Message[2]) or print(Message[-2])

Example of string Declaration:

```
>>> mes='welcome'
>>> print(mes)
welcome
>>> mes="welcome to parul university"
>>> print(mes)
welcome to parul university
>>> mes="""  welcome to parul university"""
>>> print(mes)
 welcome to parul university
>>> mes='''  welcome to parul university'''
>>> print(mes)
 welcome to parul university
>>> mes = """  welcome
                              to
                  parul                    university"""
>>> print(mes)
 welcome
                              to
                  parul                    university
```

Example of string declaration with quoted string:

```
>>> mes = 'welcome
SyntaxError: EOL while scanning string literal
>>> mes = "welcome
SyntaxError: EOL while scanning string literal
>>> mes = '''welcome
           to        parul
           university'''
>>> print(mes)
welcome
           to        parul
           university
>>> mes = 'welcome to "parul university"'
>>> print(mes)
welcome to "parul university"
>>> mes = '"welcome" "to" "parul" "university"'
>>> print(mes)
"welcome" "to" "parul" "university"
>>> mes = "welcome to 'parul university'"
>>> print(mes)
welcome to 'parul university'
>>> mes = "'welcome' 'to' 'parul' 'university'"
>>> print(mes)
'welcome' 'to' 'parul' 'university'
```

How to change element or string by passing index value?

- As string is immutable so we can change whole string but not single character by passing index value like list.
- Trying to access a character out of index range will raise an **IndexError**.
- The index must be an integer. We can't use float or other types, this will result into **TypeError**.

How to access range of element or character from string?

- To access range of element or character need to use slice ":" operator inside [ ] bracket.

e.g. print(Message[2:7]) **we need to pass start index and end index value.**

- **If we pass print(Message[::-1]) than it will print string in reverse order.**

Example of slicing operation on string or access string by passing index value

```
>>> mes[3.5]
Traceback (most recent call last):
  File "<pyshell#37>", line 1, in <module>
    mes[3.5]
TypeError: string indices must be integers
>>> mes
'good morning'
>>> mes[2:6]
'od m'
>>> mes[:4]
'good'
>>> mes[::2]
'go onn'
>>> mes[::4]
'g n'
>>> print("string in reverse order : ",mes[::-1])
string in reverse order :  gninrom doog
>>> mes[::-2]
'girmdo'
>>> print(mes)
good morning
>>> mes[5]='M'
Traceback (most recent call last):
  File "<pyshell#46>", line 1, in <module>
    mes[5]='M'
```

Example to change string

```
>>> mes = "good morning"
>>> mes[3]
'd'
>>> mes[8]
'n'
>>> mes[9]
'i'
>>> mes[-1]
'g'
>>> mes[-3]
'i'
>>> mes[11]
'g'
>>> mes[12]
Traceback (most recent call last):
  File "<pyshell#34>", line 1, in <module>
    mes[12]
IndexError: string index out of range
>>> mes[-12]
'g'
>>> mes[-13]
Traceback (most recent call last):
  File "<pyshell#36>", line 1, in <module>
    mes[-13]
```

How to change or delete string?

- As we discuss earlier change string is not possible. Same as deleting some character by passing index value is also not possible.
- Because string is immutable.
- So to delete whole string is possible using inbuilt function del.

e.g. del Message or del(Message)

Find length of string

- To find length of string one can use len() inbuilt function of string class.
- We can also iterate string using loop and find its length by declaring counter variable inside loop like.

```
>>> mes="welcome to parul university"
>>> print("lenght of string is : ",len(mes))
lenght of string is :  27
>>> len(mes)
27
```

Without len() function

```
Count=0
for I in Message:
    Count+=1
print(Count)
```

Indexing and Splitting (Slicing) of string

Syntax : string_var[start index : end index]

Positive index value

0	1	2	3	4	5	6	7	8	9
U	N	I	V	E	R	S	I	T	Y
-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

Negative index value

- It will print character up to **end index -1.**

print(mystr[2:6] output is : IVER

print(mystr[4:] output is : ERSITY

print(mystr[:] output is : UNIVERSITY

print(mystr[::-1] output is : YTISREVINU

LIKE WISE YOU CAN DO ANY COMBINATION (please check example on slide no 7)

Traverse string using loop

Using while loop

```
fruit = 'banana'
index = 0
while index < len(fruit) :
   letter = fruit[index]
    print index, letter
    index = index + 1
```

```
0 b
1 a
2 n
3 a
4 n
5 a
```

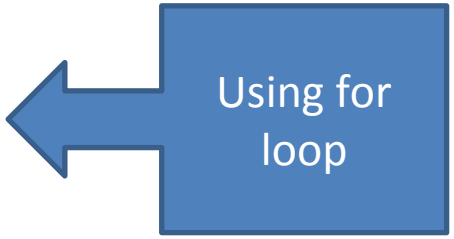
Using for loop

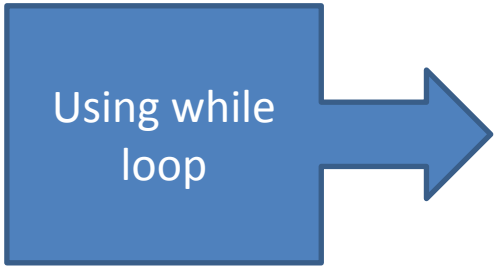
```
fruit=“banana”
for I in fruit:
    print I
```

Count occurrence of character or counting of character

- You can also use count() inbuilt method like mystr.count("a").
- Count character occurrence using for loop or while loop also.

```
Mystr="wel come"
Count=0
for I in Mystr:
    if I=='e':
        Count+=1
print(Count)
```






```
Mystr="wel come"
Count=0
index=0
while index < len(Mystr):
        if (Mystr[index] == 'e'):
            Count+=1
        index+=1
print(Count)
```

Concatenation operation on string

To perform concatenate operation on string use “+” operator.

```
>>> str1 = "Hello”
>>> str2 = 'there'
>>> bob = str1 + str2
>>> print bob
Hellothere
>>> str3 = '123'
>>> str3 = str3 + 1
Traceback (most recent call last):  File "<stdin>", line 1, in
<module>TypeError: cannot concatenate 'str' and 'int' objects
>>> x = int(str3) + 1     here we convert string into integer
>>> print x
124
```

Reading and Converting

- By using input() method we can read data or value through key board.
- But by default type of input() method is string so we need to convert it.
- In python 2 we have raw_input() and input() method both but from python 3 onwards only input() is there.

```
number=input("enter any number : ")
enter any number : 100
>>> print(number + 1)
Traceback (most recent call last):
  File "<pyshell#12>", line 1, in <module>
    print(number + 1)
TypeError: can only concatenate str (not "int") to str
>>> number=int(number)
>>> print(number +1)
101
You can directly convert like this also
Number=int(input(“enter number : “))
```

String Membership Test (using in and not in operator)

- We can test if a sub string exists within a string or not, using the keyword.
- It will return Boolean value True or False.

```
>>> 'at' in 'battle'
True
>>> 'at' not in 'battle'
False
>>> 'I' in 'index'
False
>>> 'I' not in 'index'
True
```

String repetition

- String repetition done using * operator like,
- print(mystr*3) it will repeat mystr three times on prompt

Searching

- Searching is possible using inbuilt method find() of string.
- You can also search character by iterating your string using for loop or while loop.

```
>>> mystr='bhavika'
>>> print(mystr.find('v'))
3
```

```
index = 0
word='banana'
letter='a'
while index < len(word):
        if word[index] == letter:
            print( index)
        index = index + 1
```

Using while loop

Using for loop

```
word="banana"
Letter='a'
Count=0
for I in word:
        if I == Letter:
            print(Count)
        Count+=1
```

Built in method of string

Method	Description
capitalize()	It capitalizes the first character of the String.
count(string,begin,end)	It counts the number of occurrences of a substring in a String between begin and end index.
isdigit()	It returns true if all the characters are digits
islower()	It returns true if the characters of a string are in lower case, otherwise false.
isupper()	It returns false if characters of a string are in Upper case, otherwise False.
len(string)	It returns the length of a string.
upper()	It converts all the characters of a string to Upper Case.
lower()	It converts all the characters of a string to Lower case.
find()	It returns index value if character found.

Perform below listed programs

- Write a Python function that takes a list of words and returns the length of the longest one.
- Write a Python script that takes input from the user and displays that input back in upper and lower cases.
- Write a Python function to reverses a string if it's length is a multiple of 2.
- Write a Python program to check whether a string starts with specified characters.
- Write a Python program to add a prefix text to all of the lines in a string.
- Write a Python program to add a postfix text to all of the lines in a string.
- Write a Python program to count numbers of vowels in the string.
- Write a Python program to convert string in upper case and lowercase without using inbuilt function.
- Write a menu driven program to perform all operation using user defined function. (you can use inbuilt function in UDF).
- Write a Python program to count numbers of white space in given sentence.
- Write a Python program to swap the last and first character of given string and print updated string.

Summery

- Any number or alphabet or group of character enclosed within single quote or double quote known as string.
- We can also say its sequence of characters.
- Can not perform any arithmetical operation on string.
- By using + one can perform string concatenation.
- Its immutable so cannot change its data by passing index value.
- Its accessible from both the side right to left as well as left to right.
- Can allow to perform slice operation and repetition (replication) operation using * operator.
- To convert any data or element into string we have str() inbuilt function.
- Its support various inbuilt methods like len(), isUpper(), isLower, lower(), upper() etc method to perform string operations.