String

Note Book Owner: Emdadul Hoque

String Declaration

```
In [1]: #declare & create a  string using constructor of a str class
        s1 = str() #create a empty string object
        s2 = str("string declararion")
```

```
In [2]: s2
```

```
Out[2]: 'string declararion'
```

```
In [3]: #create a string with using double cotation
        s1 = " " #create a empty string
        s2 = "Hello"
```

```
In [6]: s2
```

```
Out[6]: 'Hello'
```

InBuid python functions for strings

```
In [7]: string = " I love my country"
```

```
In [9]: len(string) # number of charaters in string
```

```
Out[9]: 18
```

```
In [10]: min(string) #samllest character in a string
```

```
Out[10]: ' '
```

```
In [11]: max(string) #largest character in a string
```

```
Out[11]: 'y'
```

Access value in a string

```
In [12]: name = "Emdadul"
```

```
In [13]: name[0]
```

```
Out[13]: 'E'
```

```
In [18]: name[4]
```

```
Out[18]: 'd'
```

```
In [19]: length = len(name)
```

Accesing Characters via negative index

In [25]:

```
name = "Emdadul"
```

In [26]:

```
name[-1]
```

Out[26]:

```
'l'
```

In [28]:

```
name[-3]
```

Out[28]:

```
'd'
```

Traversing string

In [29]:

```
name = "Emdadul"
```

In [30]:

```
for char in name:
    print(char, end="")
```

```
Emdadul
```

In [32]:

```
#another way
str_len = len(name)

for i in range(str_len):
    print(name[i], end="")
```

```
Emdadul
```

Problem 01: Write a program to traverse every second character of a string

In [37]:

```
sentence = "I love python programming"
for ch in range(0, len(sentence), 2):
    print(sentence[ch], end=" ")
```

```
I l v   y h n p o r m i g
```

In [38]:

```
#sting immutable that means you cann't change string value in same memory location

str1 = "I love python"
```

In [39]:

```
str1[0]
```

Out[39]:

```
'I'
```

In [41]:

```
str1[0] = "E"
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[41], line 1
----> 1 str1[0] = "E"

TypeError: 'str' object does not support item assignment
```

String Slicing Operator [start: end]

In [42]:

```
s = "Emdadul"
```

In [43]:

```
#string_name[start index: End index+1]
s[0:6]
```

Out[43]:

```
'Emdadu'
```

In [44]:

```
s[0:7]
```

```
Out[44]: 'Emdadul'
```

```
In [46]: #string slicing with step size
         s[0:7:2] #string_name[start index: end index+1:number of step]
```

```
Out[46]: 'Eddl'
```

```
In [47]: s[::]
```

```
Out[47]: 'Emdadul'
```

```
In [48]: s[::-1] #print string in reverse
```

```
Out[48]: 'ludadmE'
```

```
In [49]: s[-1:0:-1]
```

```
Out[49]: 'ludadm'
```

The string +, * and in operators

```
In [54]: # "+" operator
         first_name = "Emdadul "
         last_name = "Tareque"
```

```
In [55]: first_name + last_name
```

```
Out[55]: 'Emdadul Tareque'
```

```
In [58]: # "*" operator
         s1 = "tareque "
         s2 = s1 * 3
```

```
In [59]: s2
```

```
Out[59]: 'tareque tareque tareque '
```

```
In [60]: # "in operator"
         s1 = "Emdadul Tareque"
```

```
In [61]: "tareque" in s1
```

```
Out[61]: False
```

```
In [62]: "xyz" in s1
```

```
Out[62]: False
```

Problem 01: Write a program to print all the letters from word1 that also appear in word2. Example: word1 = Dhaka North City Corporation word2 = Dhaka South City Corporation #print the letter that appear in word1 and also word2 output: Dhaka outh City Corporation

```
In [67]: word1 = "Dhaka North City Corporation"
         word2 = "Dhaka South City Corporation"

         # for i in range(len(word1)):
         #     if word2[i] == word1[i]:
         #         print(word2[i], end="")

         for letter in word1:
```

```
    if letter in word2:
        print(letter, end="")
```

```
Dhaka orth City Corporation
```

String Operations

The str class provides different basic methods to perform various operations on a string. It helps to calculate the length of a string, to retrieve the individua characters from the giveb string and to compare and concatenate the two different strings.

String comparison

```
In [84]: s1 = "xyz"
         s2 = "XYZ"
```

```
In [86]: s1==s2
```

```
Out[86]: False
```

```
In [88]: s1>s2 #its true because in python ASCII value of "a" is 97 and "A" is 65
```

```
Out[88]: True
```

```
In [79]: s1.upper()
```

```
Out[79]: 'XYZ'
```

```
In [80]: s1==s2
```

```
Out[80]: False
```

```
In [82]: s1= s1.upper()
```

```
In [83]: s1==s2
```

```
Out[83]: True
```

String .format() method

```
In [91]: s = "my name is {} and i am from {}".format("Emdadul", "Bangladesh")
```

```
In [92]: s
```

```
Out[92]: 'my name is Emdadul and i am from Bangladesh'
```

Explanation The format() method is called on the string literal with arguments 4,5 and ‘nine’. The empty {} are replaced with the arguments in order. The first {} curly bracket is replaced with the first argument and so on. By default, the index of the first argument in format always start from zero. One can also give a position of arguments inside the curly brackets. The following example illustrates the use of index as argument inside the curly bracket.

```
In [93]: s = "my name is {0} and i am from {1}".format("Emdadul", "Bangladesh")
```

```
In [94]: s
```

```
Out[94]: 'my name is Emdadul and i am from Bangladesh'
```

```
In [95]: s = "my name is {1} and i am from {0}".format("Emdadul", "Bangladesh")
```

```
In [96]: s
```

```
Out[96]: 'my name is Bangladesh and i am from Emdadul'
```

```
In [97]: s = "I am {0} years old. I love to work on {pc} laptop".format(28, pc="HP")
```

```
In [98]: s
```

```
Out[98]: 'I am 28 years old. I love to work on HP laptop'
```

The split() method

The split() method returns a list of all the words in a string. It is used to break up a string into smaller strings.

```
In [99]: str = "My name is tareque"
         str.split()
```

```
Out[99]: ['My', 'name', 'is', 'tareque']
```

Problem 02:Consider a input string that has a list of names of various multinational companies, such as TCS, INFOSYS, MICROSOFT, YAHOO and GOOGLE. Use split method and display the name of each company in a different line.

```
In [101…: it_company = "Meta Google Microsoft Amazon OpenAI"

          it_company_list = it_company.split()

          print(it_company_list)
          for item in it_company_list:
              print(item)
```

```
['Meta', 'Google', 'Microsoft', 'Amazon', 'OpenAI']
Meta
Google
Microsoft
Amazon
OpenAI
```

Testing String

A string may contain digits, alphabets or a combination of both of these. Thus, various methods are available to test if the entered string is a digit or alphabet or is alphanumeric

isalnum() --> Returns True if characters in the string are alphanumeric and there is at least one character.

```
In [135…: s = "python"
          s.isalnum()
```

```
Out[135…: True
```

```
In [110…: s = "Python Programming"
          s.isalnum()
```

```
Out[110… False
```

```
In [113… s = "Age23"
         s.isalnum()
```

```
Out[113… False
```

```
In [116… s = "Age:23" # ইংরেজি শব্দ অথবা সংখ্যা বাদে যে কোন কিছু থাকলে এই ফাংশন ফলস রিটার্ন দিবে।
         s.isalnum()
```

```
Out[116… False
```

isalpha() --> Returns True if the characters in the string are alphabetic and there is at least one character.

```
In [118… s = "Emdadul"
         s.isalpha()
```

```
Out[118… True
```

```
In [119… s = "age45"
         s.isalpha()
```

```
Out[119… False
```

isdigit() --> Returns True if the characters in the string contain only digits.

```
In [121… s = "1234"
         s.isdigit()
```

```
Out[121… True
```

```
In [122… s = "abc"
         s.isdigit()
```

```
Out[122… False
```

islower() ---> Returns True if all the characters in the string are in lowercase.

```
In [132… s = "happy"
         s.islower()
```

```
Out[132… True
```

```
In [133… s = "Happy"
         s.islower()
```

```
Out[133… False
```

isupper() --> Returns True if all the characters in the string are in uppercase.

```
In [128… s = "happy"
         s.isupper()
```

```
Out[128… False
```

In [129…

```
s = "Happy"
s.isupper()
```

Out[129…

```
False
```

In [130…

```
s = "HAPPY"
s.isupper()
```

Out[130…

```
True
```

isspace() ---> Returns true if the string contains only white space characters.

In [136…

```
s = "My name is tareque"
s.isspace()
```

Out[136…

```
False
```

In [137…

```
s = " "
s.isspace()
```

Out[137…

```
True
```

Searching Substring in a string

endswith(str str1) --> Returns true if the string ends with the substring Str1.

In [140…

```
s = "python programming"
s.endswith("programming")
```

Out[140…

```
True
```

In [141…

```
s = "python programming"
s.endswith("python")
```

Out[141…

```
False
```

startswith(str) -->Returns true if the string starts with the substring Str1

In [142…

```
s = "python programming"
s.startswith("python")
```

Out[142…

```
True
```

find(str) --> Returns the lowest index where the string Str1 starts in this string or returns -1 if the string Str1 is not found in this string.

In [148…

```
str = "Emdadul Hoque"
str.find("Hoque")
```

Out[148…

```
8
```

In [151…

```
str = "Emdadul Hoque"
str.find("Hoque")
```

Out[151…

```
8
```

rfind(str) --> Returns the highest index where the string Str1 starts in this string

or returns -1 if the string Str1 is not found in this string

In [154…

```
str = "Emdadul Hoque"
str.rfind("Hoque")
```

Out[154…

```
8
```

Methods to Convert a String into Another String

In [161…

```
s = "emdadul hoque"
s.capitalize()
```

Out[161…

```
'Emdadul hoque'
```

In [162…

```
s.lower()
```

Out[162…

```
'emdadul hoque'
```

In [163…

```
s.upper()
```

Out[163…

```
'EMDADUL HOQUE'
```

In [164…

```
s.title()
```

Out[164…

```
'Emdadul Hoque'
```

In [166…

```
s = "Emdadul Hoque"
s.swapcase()
```

Out[166…

```
'eMDADUL hOQUE'
```

In [167…

```
s = "I have brought two chocolates, two cookies and two cakes"
str = s.replace("two", "Three")
```

In [168…

```
str
```

Out[168…

```
'I have brought Three chocolates, Three cookies and Three cakes'
```

In [169…

```
s = "I have brought two chocolates, two cookies and two cakes"
str = s.replace("two", "Three", 2)
```

In [170…

```
str
```

Out[170…

```
'I have brought Three chocolates, Three cookies and two cakes'
```

Stripping Unwanted Characters from a String

In [171…

```
s = " my name is khan"
s.lstrip() # সেনটেন্সের শুরুতে স্পেস থাকলে রিমুভ করে দেয়
```

Out[171…

```
'my name is khan'
```

In [172…

```
s = "\t \t \t my name is khan"
s.lstrip()
```

Out[172…

```
'my name is khan'
```

In [184…

```
s = "my name is khan!!! \t \t \t"
s = s.rstrip() # Return a copy of the string with trailing whitespace removed.}
```

In [185…

```
s
```

Out[185…

```
'my name is khan!!!'
```

In [187…

```
s = "\t \t \t \t my name is khan \t \t \t"
s.strip() #Return a copy of the string with leading and trailing whitespace removed.
```

Out[187…

```
'my name is khan'
```

Formating String

In [199…

```
s = "Apple Macos"
s.center(20)# Return a centered string of length width.
```

Out[199…

```
'    Apple Macos     '
```

In [200…

```
s.ljust(15)
```

Out[200…

```
'Apple Macos    '
```

In [201…

```
s.rjust(15)
```

Out[201…

```
'    Apple Macos'
```

In [ ]: