

05-Python Basic

String data-types

Posted on September 9, 2021

Last updated on January 31, 2023

5. String Data Type

Strings are used to record the text information such as name. In Python, Strings act as Sequence which means Python tracks every element in the String as a sequence. This is one of the important features of the Python language.

For example, Python understands the string "hello" to be a sequence of letters in a specific order which means the indexing technique to grab particular letters (like first letter or the last letter).

Note: In most of other languages like C, C++,Java, a single character with in single quotes is treated as char data type value. But in Python we are not having char data type. Hence it is treated as String only.

```
ch='a'
type(ch) # <class 'str'>

string = 'Amrit'
type(string) # <class 'str'>
```

5.1 Define String literals

```
# valid string
print("This is ' double quote symbol")
print('This is " single quotes symbol')
print('String with escape \' char')
print('The \"Python Notes\" by \'durga\' is very helpful')
print('''This is a "multi-line"
string''')

# Invalid String
print('This is ' single quote symbol')
print('The "Python Notes" by 'ap' is very helpful')
print("The "Python Notes" by 'ap' is very helpful")
```

5.2 Slicing of Strings

- Slice means a piece
- [] operator is called slice operator, which can be used to retrieve parts of String.

```
# Syntax --> [start: end: step]

a1 = 'Amrit'

-5 -4 -3 -2 -1
A  m  r  i  t
0  1  2  3  4

a1[2:]  # rit
a1[:2]  # Am

a1[::2]  # 'Art'

# a1[::-1]  # 'tirmA'

a1[::]  # Amrit
a1[:]  # Amrit
```

```
print("Hello" + "Worlds")
# HelloWorlds
```

```
print('Hello'*2)
# HelloHello
```

5.3 String in-built function

- `len()`
 - `len()` function to find the number of characters present in the string

```
a1 = 'Amrit'
len(a1) # 5
```

- `count()`
 - Counting substring in the given String

```
s1 = 'Hello Amrit llo LLo'

s1.count('llo') # 2
s1.count('l') # 4
s1.count('AAA') # 0
```

- `replace()`
 - Replace every occurrence of oldstring will be replaced with newstring

```
str1 = "Difficult, Python is Difficult"
str2 = str1.replace("Difficult", "Easy")
str2
# 'Easy, Python is Easy'
```

Removing spaces from the string

- `rstrip()`
 - To remove spaces at right hand side
- `lstrip()`
 - To remove spaces at left hand side
- `strip()`
 - To remove spaces both sides

```
string_a=" pwskills "
```

```
string_a.strip(" ")  
# 'pwskills'
```

```
string_a.lstrip(" ")  
# 'pwskills '
```

```
string_a.rstrip(" ")  
# ' pwskills'
```

Joining of Strings

- join()
 - We can join a group of strings(list or tuple) wrt the given separator

```
str1 = ('sunny', 'bunny', 'chinny')  
' '.join(str1)  
# 'sunny bunny chinny'  
  
" ".join("abcd")  
# 'a b c d'  
  
' Pwskills '.join(reversed("ant"))  
# 't Pwskills n Pwskills a'
```

Splitting of Strings

- split()

```
str1 = '22-02-2018'  
str1.split('-')  
# ['22', '02', '2018']
```

- partition()

```
str1.partition('-')  
# ('22', '-', '02-2018')  
  
str1.rpartition('-')  
# ('22-02', '-', '2018')
```

Checking starting and ending part of the string

- `startswith()`
- `endswith()`

```
s = 'learning Python is very easy'
print(s.startswith('learning')) # True
print(s.endswith('learning'))  # False
print(s.endswith('easy'))     # True
```

Finding Substrings

- `find()` and `index()`
 - Finds sub-string in given string
 - Return index of 1st char of first sub-string in forward direction

```
# find()

s1 = 'Hello Amrit'
s1.find('llo')  # 2

# `No Error` if sub-str not found
s1.find('lloo') # -1
# `Returns -1`
```

```
# index()

s1.index('llo')  # 2

# `gives Error` if sub-str not found
s1.index('loo')
# ValueError: substring not found
```

- `rfind()` and `rindex()`
 - Same as `find()` and `index()`
 - But it works in backward direction

```
s1 = 'Hello Amrit'

s1.rfind('llo')  # 2
s1.rindex('llo') # 2
```

Changing case of a String:

- `upper()`
 - To convert all characters to upper case
- `lower()`
 - To convert all characters to lower case
- `swapcase()`
 - Converts all lower case characters to upper case and all upper case characters to lower case
- `title()`
 - To convert all character to title case
 - i.e first character in every word should be upper case and all remaining characters should be in lower case.
- `capitalize()`
 - Only first character will be converted to upper case and all remaining characters can be converted to lower case

```
s='learning Python is very Easy'
```

```
print(s.upper())
```

```
# LEARNING PYTHON IS VERY EASY
```

```
print(s.lower())
```

```
# learning python is very easy
```

```
print(s.swapcase())
```

```
# LEARNING PYTHON IS VERY eASY
```

```
print(s.title())
```

```
# Learning Python Is Very Easy
```

```
print(s.capitalize())
```

```
# Learning python is very easy
```

To check type of characters present in a string:

- `isalnum()`
 - Returns True if all characters are alphanumeric(a to z , A to Z ,0 to9)
- `isalpha()`
 - Returns True if all characters are only alphabet symbols(a to z,A to Z)
- `isdigit()`
 - Returns True if all characters are digits only(0 to 9)

- `islower()`
 - Returns True if all characters are lower case alphabet symbols
- `isupper()`
 - Returns True if all characters are upper case alphabet symbols
- `istitle()`
 - Returns True if string is in title case
- `isspace()`
 - Returns True if string contains only spaces

Formatting the Strings

```
name, salary, age = 'Amrit', '50K', 22
print("{} 's salary is {} and his age is {}".format(
    name, salary, age))
print("{0} 's salary is {1} and his age is {2}".format(
    name, salary, age))
print("{x} 's salary is {y} and his age is {z}".format(
    z=age, y=salary, x=name))

# Amrit 's salary is 50K and his age is 22
# Amrit 's salary is 50K and his age is 22
# Amrit 's salary is 50K and his age is 22
```

.

.

.

.

.

.

.

.