

Python Programming

- Presentation By Uplatz
- Contact us: <https://training.uplatz.com>
- Email: info@uplatz.com
- Phone: +44 7836 212635

STRINGS

Learning outcomes:

Accessing Values in Strings

Updating Strings

String Special Operators

Built-in String Methods

STRINGS

Strings are amongst the most popular types in Python. We can create them simply by enclosing characters in quotes. Python treats single quotes the same as double quotes. Creating strings is as simple as assigning a value to a variable. For example:

```
var1 = 'Hello World!'
```

```
var2 = "Python Programming"
```

Accessing Values in Strings

Python does not support a character type; these are treated as strings of length one, thus also considered a substring.

To access substrings, use the square brackets for slicing along with the index or indices to obtain your substring.

For example:

```
s = "Python Programming"  
print(s[0]) # output is 'P'  
print(s[1]) # output is 'y'  
print(s[1:5]) # output is 'ytho'
```

Updating Strings

You can "update" an existing string by (re)assigning a variable to another string. The new value can be related to its previous value or to a completely different string altogether.

For example:

```
x = 'Learning from an expert'  
print(x[:6])  
print(x[:8]+'python')
```

Updating Strings

In Python, **Updating or deleting of characters from a String is not allowed.** This will cause an **error** because **item assignment or item deletion from a String is not supported.** Although deletion of entire String is possible with the use of a built-in **del** keyword. **This is because Strings are immutable, hence elements of a String cannot be changed once it has been assigned. Only new strings can be reassigned to the same name.**

String Special Operators

- + “Concatenation - Adds values on either side of the operator “
- * “Repetition - Creates new strings, concatenating multiple copies of the same string”
- [] “Slice - Gives the character from string index”
- [:] “Range Slice - Gives the characters from the given range”
- in “Membership - Returns true if a character exists in the given string “
- not in “Membership - Returns true if a character does not exist in the given string “
- r/R “Raw String”

String Special Operators

Example:

```
a=""Hello""
```

```
b='World'
```

```
print(a+b)
```

```
print(a*2)
```

```
print(b[2])
```

```
c='W'
```

```
print(c in b)
```

```
print(c not in b)
```

```
print(r"Your father's name is \n")
```

Built-in String Methods

Python includes the following built-in methods to manipulate strings :

capitalize() :Capitalizes first letter of string.

center(width, fillchar) :Returns a space-padded string with the original string centered to a total of width columns.

count() : The method **count()** returns the number of occurrences of substring sub in the range [start, end].

find():It determines if string *str* occurs in string, or in a substring of string if starting index *beg* and ending index *end* are given.

Built-in String Methods

index(): It determines if string *str* occurs in string or in a substring of string if starting index *beg* and ending index *end* are given. This method is same as `find()`, but raises an exception if sub is not found.

isalnum(): It checks whether the string consists of alphanumeric characters.

isalpha() : The method **isalpha()** checks whether the string consists of alphabetic characters only.

isdigit() : The method **isdigit()** checks whether the string consists of digits only.

Built-in String Methods

islower(): The method **islower()** checks whether all the case-based characters (letters) of the string are lowercase.

isnumeric() :The method **isnumeric()** checks whether the string consists of only numeric characters. This method is present only on unicode objects.

isspace() :The method **isspace()** checks whether the string consists of whitespace.

isupper() : The method **isupper()** checks whether all the case-based characters (letters) of the string are uppercase.

Built-in String Methods

len() : It returns the length of the string.

lower() : The method **lower()** returns a copy of the string in which all case-based characters have been lowercased.

max() : The method **max()** returns the max alphabetical character from the string *str*.

min() : The method **min()** returns the min alphabetical character from the string *str*.

replace() : The method **replace()** returns a copy of the string in which the occurrences of *old* have been replaced with *new*, optionally restricting the number of replacements to *max*.



Thank you