

## Module-2 Lecture-10

### **Python – 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"
```

GALGOTIAS  
UNIVERSITY

## 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.

### Example :

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

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

```
print "var1[0]: ", var1[0]
```

```
print "var2[1:5]: ", var2[1:5]
```

When the above code is executed, it produces the following result –

```
var1[0]: H
```

```
var2[1:5]: 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.

### Example :

```
var1 = 'Hello World!' print "
```

```
Updated String :- ", var1[:6] + 'Python'
```

When the above code is executed, it produces the following result –

```
Updated String :- Hello Python
```

GALGOTIAS  
UNIVERSITY

## Escape Characters

Following table is a list of escape or non-printable characters that can be represented with backslash notation.

An escape character gets interpreted; in a single quoted as well as double quoted strings.

Backslash notation	Hexadecimal character	Description
<code>\a</code>	0x07	Bell or alert <code>print("\a")</code> <code>&gt;&gt;&gt; print('Hello \aWorld!')</code>
<code>\b</code>	0x08	Backspace <code>&gt;&gt;&gt; print('Hello \bWorld!')</code> Hello World!
<code>\cx</code>		Control-x
<code>\C-x</code>		Control-x
<code>\e</code>	0x1b	Escape

# School of Basic and Applied Sciences

Course Code : BSCM 304

Course Name: Programming Using Python

Backslash notation	Hexadecimal character	Description
<code>\f</code>	<code>0x0c</code>	Formfeed
<code>\M-\C-x</code>		Meta-Control-x
<code>\n</code>	<code>0x0a</code>	Newline <pre>&gt;&gt;&gt; print('Hello \nWorld!') Hello World! &gt;&gt;&gt; print('Hello \x0aWorld!') Hello World!</pre>
<code>\nnn</code>		Octal notation, where n is in the range 0-7
<code>\r</code>	<code>0x0d</code>	Carriage return <pre>&gt;&gt;&gt; print('Hello \rWorld!') Hello World!</pre>

# School of Basic and Applied Sciences

Course Code : BSCM 304

Course Name: Programming Using Python

Backslash notation	Hexadecimal character	Description
<code>\s</code>	<code>0x20</code>	Space Ex: <code>print('Hello \s World!')</code> Hello \s World!
<code>\t</code>	<code>0x09</code>	Tab >>> <code>print('Hello \t World!')</code> Hello      World!
<code>\v</code>	<code>0x0b</code>	Vertical tab <code>print('Hello \v World!')</code> Hello World!
<code>\x</code>		Character x
<code>\xnn</code>		Hexadecimal notation, where n is in the range 0-9, a-f, or A-F

## String Special Operators:

Assume string variable a holds 'Hello' and variable b holds 'Python', then –

Operator	Description	Example
+	Concatenation - Adds values on either side of the operator	a + b will give HelloPython
*	Repetition - Creates new strings, concatenating multiple copies of the same string	a*2 will give -HelloHello
[]	Slice - Gives the character from the given index	a[1] will give e
[ : ]	Range Slice - Gives the characters from the given range	a[1:4] will give ell

# School of Basic and Applied Sciences

Course Code : BSCM 304

Course Name: Programming Using Python

Operator	Description	Example
in	Membership - Returns true if a character exists in the given string	H in a will give 1 a='Hello' if 'H' in a: print('Love')
not in	Membership - Returns true if a character does not exist in the given string	M not in a will give 1
r/R	Raw String - Suppresses actual meaning of Escape characters. The syntax for raw strings is exactly the same as for normal strings with the exception of the raw string operator, the letter "r," which precedes the quotation marks. The "r" can be lowercase (r) or uppercase (R) and must be placed immediately preceding the first quote mark.	print r'\n' prints: \n and print R'\n' prints: \n print('\r Love') Love
%	Format - Performs String formatting	See at next section



## References:

1. Introduction to Computation and Programming using Python, by John Guttag, PHI Publisher
2. Python Programming using problem solving Approach by Reema Thareja, Oxford University, Higher Education Oxford University Press; First edition (10 June 2017), ISBN-10: 0199480173
3. <https://www.tutorialspoint.com/python/index.htm>
4. <https://www.geeksforgeeks.org/python-programming-language>

GALGOTIAS  
UNIVERSITY

**\*\*\*END OF THE LECTURE\*\*\***

**\*\*\*THANK YOU\*\*\***

GALGOTIAS  
UNIVERSITY