**Strings**

Create strings simply by enclosing characters in quotes.

Python treats single quotes the same as double quotes. ‘ ≈ “

Python does not support a character type; these are treated as strings of length one. char = str with length 1

# Creating String

Creating strings is as simple as assigning a value to a variable.

#!/usr/bin/python3

var1 = 'Hello World!'

var2 = "Python Programming"

var3 = 'Hello\nWorld!'

var4 = "Python\nProgramming"

print(var1)

print(var2)

print(var3)

print(var4)

Output:

Hello World!

Python Programming

Hello

World!

Python

Programming

# Updating String

To update an existing string reassign another string to variable.

The new value can be related to its previous value or to a completely different string altogether.

#!/usr/bin/python3

var1 = 'Hello World!'

print("Updated String: ", var1[:6] + 'Python')

Output:

Updated String: Hello Python

# Triple Quotes

Python's triple quotes allow strings to span multiple lines, including verbatim NEWLINEs, TABs, and any other special characters.

The syntax for triple quotes consists of three consecutive single or double quotes.

#!/usr/bin/python3

para\_str = """This is a long string that is made up of several lines and non-printable characters such as TAB ( \t ) and they will show up that way when displayed. NEWLINEs within the string, whether explicitly given like this within the brackets [ \n ], or just a NEWLINE within the variable assignment will also show up."""

print (para\_str)

Output:

This is a long string that is made up of several lines and non-printable characters such as TAB ( ) and they will show up that way when displayed. NEWLINEs within the string, whether explicitly given like this within the brackets [

], or just a NEWLINE within the variable assignment will also show up.

# Raw strings

Raw strings do not treat the backslash as a special character at all.

Every character you put into a raw string stays the way you wrote it.

#!/usr/bin/python3

print('C:\\nowhere')

print(r'C:\\nowhere') # raw string

Output:

C:\nowhere

C:\\nowhere

# Unicode String

In Python 3, all strings are represented in Unicode.

In Python 2 are stored internally as 8-bit ASCII, hence it is required to attach 'u' to make it Unicode.

# String Special Operators

Assume a = 'Hello' and b = 'Python', then:

|  |  |  |
| --- | --- | --- |
| Operator | Description | Example |
| + | Concatenation  Adds values on either side of the operator | a + b  HelloPython |
| \* | Repetition  Creates new strings, concatenating multiple copies of the same string | a\*2  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]  ell |
| in | Membership  Returns true if a character exists in the given string | H in a  1 |
| not in | Membership  Returns true if a character does not exist in the given string | M not in a  1 |
| r/R | Raw String - Suppresses actual meaning of Escape characters  The "r" can be lowercase (r) or uppercase (R) | print r'\n'  \n and  print R'\n'  \n |
| % | Format - Performs String formatting | See String formatting |