

Strings

String is a primitive datatype. Strings are represented by a sequence of letters, numbers and special

characters enclosed between single(' ') or double("") quotes. For example:

- o 'Hello World!'
- o "She said 'no' to me"

When double quotes are used to enclose strings, single quotes can be used freely inside:

"Plants love to 'eat' sunlight."

This also applies vice-versa:

'He said, "Today is MY day!", and ran away.'

When using both single and double quotes inside, enclose with either of them and use escape sequences for the other:

'He said, "Plants love to \'eat\' sunlight.", and sat.'

or

"He said, \"Plants love to 'eat' sunlight.\", and sat."

Escape Sequences

Escape sequences are special characters that play a completely different role than they actually play. Escape sequences start with a backslash(\) followed by one or more characters. Python treats all characters in an escape sequence as one character. Some of the commonly used escape sequences are:

- o \n - newline
- o \t - tab space
- o \b - backspace
- o \r - carriage return
- o \f - form feed
- o \\ - backslash
- o \' - single quote
- o \" - double quotes

• Strings are indexable, which means that constituent characters can be accessed using index numbers, ranging from 0 to n-1, where n is the length of string.

• Strings are iterable, which means they can be used in a for loop to access each character.

• Strings are immutable, which means that individual characters cannot be changed by using indexes too.

E.g:

a = 'Sup?'

a[3] = '!' # illegal!!

Operations on Strings

Concatenation

Two strings can be joined using the + operator to give a joined string. For example:

'Hello' + 'World' returns 'HelloWorld'

Duplication

When a string is multiplied with a positive integer, it returns the string repeated by the number multiplied. For example:

'Sup' * 5 returns 'SupSupSupSupSup'

Membership

The in and not in operators can be used to check whether a character or a sequence of characters is present in a string or not. For example:

'lie' in 'believe' returns True

String Functions

Assuming a string a = 'hello',

1) a.upper()

- o Returns a in UPPERCASE, non-letters ignored
- o a.upper() -> 'HELLO'

2) a.lower()

- o Returns a in lowercase, non-letters ignored
- o 'WORLD!'.lower() -> 'world!'

3) a.isalpha()

- o Returns True if complete string consists of letters, else returns False
- o a.isalpha() -> True

4) a.isdigit()

- o Returns True if complete string consists of numbers, else returns False
- o '123'.isdigit() -> True

5) a.strip()

- o Remove leading and trailing spaces and new-lines, but not middle spaces, and returns string
- o ' a aa '.strip() -> 'a aa'

6) a.replace(from, to)

- o Replaces each occurrence of from in a with to and returns it
- o 'old is gold'.replace('old', 'van') -> 'van is gvan'

7) a.split(st)

- o Splits a into a list using st as a break point
- o 'old#is#gold'.split('#') -> ['old', 'is', 'gold']

8) a.index(x)

- o Returns index of first occurrence of x in a
- o a.index('l') -> 2

9) a.title()

- o Converts the first character of each word to upper case and returns it
- o 'old is gold'.title() -> 'Old Is Gold'