**Strings in Python**

* String is a set of characters. (One dimensional character array).
* String represents by using either single quotes or double quotes in python.

>>> s1 = 'Hello'

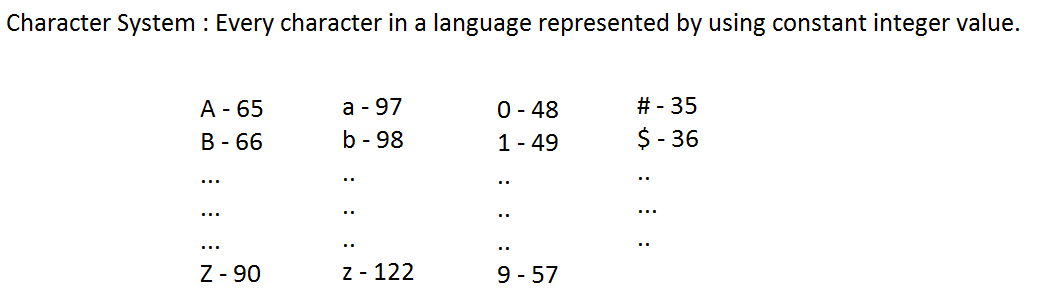
>>> s2 = "Python"

>>> print(s1,s2)

Hello Python

* In other languages, character can be represented using single quotes and strings can be represented using double quotes.
* Python does not support a character type. A character is a String with length ‘1’.

>>> s = 'A' #String declaration



We cannot convert a character(single char string) into integer using int() method in python.

>>> ch = 'A'

>>> int(ch)

ValueError: invalid literal for int() with base 10: 'A'

Ord():

A pre-fined python library function is used to convert string(char) into interger.

>>> ch = 'A'

>>> ord(ch)

65

>>> ch = '#'

>>> ord(ch)

35

>>> ch = '0'

>>> ord(ch)

48

>>> ch = 'AB'

>>> ord(ch)

TypeError: ord() expected a character, but string of length 2 found

Indexing and Slicing in Strings:

>>> s = 'Hello'

>>> print(s[0])

H

>>> print(s[4])

o

>>> len(s)

5

>>> print(s[len(s)-1])

o

Slicing of String:

* It is possible to access part of the string called ‘sub strings’.
* We use the square brackets with the index or indices to obtain your substring.

>>> s = 'Python'

>>> print(s[1:4])

yth

>>> print(s[1:])

ython

>>> print(s[:4])

Pyth

We can display strings through concatenation using ‘+’ operator.

>>> print("Sub String : "+str[0:6]) // concatenation

Sub String : Python

* For example –

>>> s = 'Python'

>>> print(s)

Python

>>> print("String starts with :",s[0])

String starts with : P

>>> print("Length : ", len(s))

Length : 6

>>> print("String ends with : ",s[len(s)-1])

String ends with : n

>>> print(s[0:3])

Pyt

>>> print(s[:3])

Pyt

>>> print(s[2:])

thon

>>> print(s[:3]+s[3:])

Python

**Strings Updation:**

* Modify string by re-assigning a variable.
* Combining Strings also called String updation.
* The new String related to its previous value or to a completely different string altogether.
* For example –

>>> s1 = "Hello"

>>> s2 = "Python"

>>> s1+s2

'HelloPython'

>>> print(s1)

Hello

>>> s1=s1+s2

>>> print(s1)

HelloPython

>>> print(s2)

Python

>>> s1="Hello"

>>> print("Modified : " , s1[:6]+"Python")

Modified : HelloPython

>>> print(s1)

Hello

>>> s1="Hello"

>>> s1=s1[:6]+"Python"

>>> print(s1)

HelloPython

String Special Operators:

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| + | Concatenate two strings | a + b : HelloPython |
| \* | Repeat same string 2 times called duplication | a\*2 : HelloHello |
| [] | Slice - Gives the character from the given index | a[1] : e |
| [ : ] | Slicing the String / acquire sub string | a[1:4] : ell |
| In | Returns true if a character is present in the given string | H in a : True |
| not in | Returns true if a character is not present in the given string | M not in a : True |
| % | Format specifier of String like in C language | Will see later |

>>> s1 = "Hello"

>>> s2 = "Python"

>>> print(s1+s2)

HelloPython

>>> print(s1\*2)

HelloHello

>>> print(s2[0])

P

>>> str = "Hello Python"

>>> print(str[-1])

n

>>> print(str[7:-2])

yth

>>> print(s1[0:5],s2[4:6])

Hello on

>>> 'H' in s1

True

>>> 'P' not in s2

False

>>> print('\n')

>>> print('\\n')

\n

>>> print(r'\n')

\n

**String Formatting Operator(%s) :**

* %s is called String Format Specifier in Python.
* It is belongs to C's printf() family.
* Following is a simple example –

>>> str = "Python"

>>> print("%s is a language"%(str))

Python is a language

>>> name, age = "Annie", 23

>>> print("I am %s, and my Age is : %d"%(name,age))

I am Annie, and my Age is : 23

>>> print("I am '%s', and my Age is : '%d'"%(name,age))

I am 'Annie', and my Age is : '23'

**Old style formatting:**

* We can even format strings like the old printf() style used in C programming language.
* We use the % operator to accomplish this.

>>> x = 12.3456789

>>> print("x value : %d" %x)

x value : 12

>>> print("x value : %f" %x)

x value : 12.345679

>>> print("x value : %.2f" %x)

x value : 12.35

**upper() and lower() methods:**

* str.upper() function converts all the lower case characters into upper case.
* str.lower() function converts upper case character into lower case.
* Strings are Immutable hence the returned string will be a new string.
* Special symbols or Digits of String will not be changed.

>>> str = "hello"

>>> print(str.upper())

HELLO

>>> str = "PYTHON"

>>> print(str.lower())

python

>>> str = "P+tH\*n"

>>> print(str.upper())

P+TH\*N

Boolean Methods

* Most of the string handling functions returns Boolean value.
* We can proceed with user logic based on Boolean value.
* There are a number of string methods that will return Boolean values:

|  |  |
| --- | --- |
| Method | True if |
| str.isalnum() | String consists of only alphanumeric characters (no symbols) |
| str.isalpha() | String consists of only alphabetic characters (no symbols) |
| str.islower() | String’s alphabetic characters are all lower case |
| str.isnumeric() | String consists of only numeric characters |
| str.isspace() | String consists of only whitespace characters |
| str.istitle() | String is in title case |
| str.isupper() | String’s alphabetic characters are all upper case |

>>> str = "Hello"

>>> print(str.endswith('a'))

False

>>> str = "Hello Python"

>>> print(str.find('P'))

6

>>> print(str.find('p'))

-1

>>> print(str.find('s'))

-1

>>> s1 = "Hello$"

>>> s2 = "Python"

>>> print(s1.isalnum())

False

>>> print(s2.isalnum())

True

>>> s1 = "Hello5"

>>> s2 = "Python"

>>> print(s1.isalpha())

False

>>> print(s2.isalpha())

True

>>> s1 = "1234"

>>> s2 = "Python"

>>> print(s1.isnumeric())

True

>>> print(s2.isnumeric())

False

>>> str = "Hello"

>>> print(str.islower())

False

>>> str="hello"

>>> print(str.islower())

True

>>> print(str.replace('h', 'm'))

mello

**Length of String:**

* The len() returns the number of characters in a string.
* This method is useful for when you need to enforce minimum or maximum password lengths.

>>>web\_Site = "Python tutorial @ java2python.com"

>>>print(len(web\_site))

**join() and split() :**

* The str.join() method will concatenate two strings.
* Instead of concatenating 2 strings, it will insert one string into another.
* To split strings, we will use the str.split() method.

>>> p = "Python"

>>> s = "Script"

>>> " ".join(p)

'P y t h o n'

>>> ",".join(s)

'S,c,r,i,p,t'

>>> p = "Python"

>>> "\*".join(p)

'P\*y\*t\*h\*o\*n'

>>> ",".join("\*".join(p))

'P,\*,y,\*,t,\*,h,\*,o,\*,n'

>>> p = "Python"

>>> print(p)

Python

>>> " ".join(reversed(p))

'n o h t y P'

>>> url = "java2python is python tutorial portal"

>>> print(url.split())

['java2python', 'is', 'python', 'tutorial', 'portal']

>>> print(url.split(' ',2))

['java2python', 'is', 'python tutorial portal']

>>> print(url.split("python"))

['java2', ' is ', ' tutorial portal']

**String Through:**

* Using [for loop](https://www.programiz.com/python-programming/for-loop) we can iterate through a string.
* Here is an example to count the number of 'l' in a string.

def findChar(sequence , letter):

count=0

for x in sequence:

if(x==letter):

count=count+1

print("count : ",count)

return

findChar("Hello Python", 'o')

**Escape Sequence:**

* If we want to print a text like -He said, "What's there?".
* We can neither use single quote or double quotes.
* This will result into SyntaxError as the text itself contains both single and double quotes.

>>> print("He said, "What's there?"")

...

SyntaxError: invalid syntax

>>> print('He said, "What's there?"')

...

SyntaxError: invalid syntax

**Note:**

One way to get around this problem is to use triple quotes.

Alternatively, we can use escape sequences.

An escape sequence starts with a backslash and is interpreted differently.

# using triple quotes

print('''He said, "What's there?"''')

# escaping single quotes

print('He said, "What\'s there?"')

# escaping double quotes

print("He said, \"What's there?\"")

Immutability :

>>> s1 = "Hello"

>>> s2 = "World"

>>> print(s1+s2)

HelloWorld

>>> print(s1,s2)

Hello World

>>> s1+s2

'HelloWorld'

>>> print(s1)

Hello

>>> print(s2)

World

>>> s1 = "Hello"

>>> s2 = "Hello"

>>> s3 = "World"

>>> id(s1)

44843392

>>> id(s2)

44843392

>>> id(s3)

44977664

>>> s1=s1+s3

>>> print(s1)

HelloWorld

>>> print(s2)

Hello

>>> id(s1)

44995312

isdecimal() :

The isdecimal() doesn't take any parameters.

Return Value from isdecimal()

The isdecimal() returns:

True if all characters in the string are decimal characters.

False if at least one character is not decimal character.

>>> s = "100"

>>> int(s)

100

>>> s = 'Python'

>>> int(s)

ValueError: invalid literal for int() with base 10: 'Python'

>>> s = "100"

>>> s.isdecimal()

True

>>> s = "Python"

>>> s.isdecimal()

False

Finding length:

def length(s):

count=0

for x in s:

count+=1

return count

s = input("Enter one string : ")

len = length(s)

print("Length of '",s,"' is : ",len)

Upper to Lower String :

def strip(s):

i=0

n=len(s)

while i<n:

c = ord(s[i])

if c>=65 and c<=90:

s = s[:i]+chr(c + 32)+s[i+1:]

i+=1

print("Final string : ",s)

return

s=input("Enter one string : ")

strip(s)

Program to check and convert a string into integer:

def StringToInteger(s):

if(s.isdecimal()):

n = int(s)

print("int value : ",n)

else:

print("Invalid string to convert");

return

s = input("Enter a string : ")

StringToInteger(s)

Iteration:

def isAlpha(s):

res = False

for x in s:

if((ord(x)>=48 and ord(x)<=57)):

res = True

break

if res:

print('Not alpha')

else:

print('alpha')

return

isAlpha('He9lo')