String :

* It is one of the built-in data structures in python.
* string is a group of one or more characters which are enclosed in between single, double , triple quotes.
* Strings can contain symbols, letters, numbers and special characters.
* Strings are immutable i.e. once they are created they can’t be changed but you can create new strings based on the original strings.
* Strings supports many operations like manipulation, formatting, searching and slicing.

**String creation**

You can create a string by enclosing a group of characters within single quotes or double quotes.

When you want to use the single quote in a string then enclose it with double quotes which is shown in below example.

You can check the datatype of the data structure that you created using type( ) function.

Examples :

INPUT:

single\_quoted\_string ='Hello world'

print(type(single\_quoted\_string))

double\_quoted\_string = " i'll go "

print(type(double\_quoted\_string))

OUTPUT:

<class 'str'>

<class 'str'>

**Accessing elements of strings**

The elements in the strings are accessed using the indexes.

Indexes start with zero and end with a number which is one less than the length of the string.

Python also supports the negative indexing which start from last element with index -1 to - length of the string

Consider the string s=”HELLOWORLD”

|  H |  E |  L |  L  |  O |  W |  O |  R  |  L |  D |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  0 |  1 |  2  |  3 |  4 |  5 |  6 |  7 |  8 |  9 |
|  -10 |  -9 |  -8 |  -7 |  -6 |  -5 |  -4 |  -3 |  -2 |  -1 |

By looking at the table shown above

To access the first element you have to use the index 0 and the last element with number one less than less than the string length i.e. 9

Example :

INPUT:

s="HELLOWORLD"

# Accessing the first element in string

print("First character of string is :", s[0])

# Accessing the last element in string

print("last character of string is :", s[9])

# Accessing the first element in string with negative index

print("First character of string is :", s[-10])

# Accessing the last element in string with negative index

print("last character of string is :", s[-1])

OUTPUT:

First character of string is : H

last character of string is : D

First character of string is : H

last character of string is : D

**Slicing in strings**

Slicing is taking a part of the original string i.e. a substring.

This is done by using the slice operator in python.

Syntax: string\_name[start\_index : End\_index]

start\_index is included and end\_index is excluded in the substring you extract

Example :

INPUT:

st="welcome to programming"

sub\_st=st[0:7] # welcome as substring using slicing

sub\_st1=st[11:22]# programming as substring using slicing

print(sub\_st)

print(sub\_st1)

OUTPUT:

welcome

programming

**Slicing to reverse the string**

You can reverse the string using the slice operator.

Syntax : string\_name[ :: -1]

Example :

INPUT:

st="welcome to programming"

st1="malayalam"

sub\_st=st[::-1] # reversing string st

print(sub\_st)

print(st1[::-1]) #reversing string st1 and printing

OUTPUT:

gnimmargorp ot emoclew

malayalam

**Deletion of string**

Once the string gets created in python you can’t delete or modify the characters in the string, when you try to modify the string you will get the error.

But you can delete the whole string using the del keyword.

String Methods

1. capitalize ( ):converts the first letter of the string to uppercase and the rest of the string to lowercase.

Example

INPUT:

st="welcome to programming"

new\_st=st.capitalize()

print(new\_st)

OUTPUT:

 Welcome to programming

===============================================

1. count ( ): used to count the number of occurrences of a specified substring within a string.

Example

INPUT:

st="the plane flew over the clouds and landed at the airport"

count=st.count("the")

print(count)

OUTPUT:

3

========================================

1. find ( ):returns the lowest index of substring in the given string if the substring is not found return -1.

Example

INPUT:

st="the plane flew over the clouds and landed at the airport"

st1=st.find("flew")

print(st1)

st="the plane flew over the clouds and landed at the airport"

st1=st.find("flews")

print(st1)

OUTPUT:

10

-1

====================================================

1. index ( ):returns the lowest index of substring in the given string if the substring is not found return ValueError.

Example

INPUT:

st="the plane flew over the clouds and landed at the airport"

st1=st.index("flew")

print(st1)

st="the plane flew over the clouds and landed at the airport"

st1=st.index("flews")

print(st1)

OUTPUT:

10

ValueError: substring not found

=======================================================

1. isalnum ( ): returns True if all the characters in the strings are alphabetic letters or numbers otherwise False.

Example

INPUT:

st="welcome to programming"

print(st.isalnum())

st1="Hello"

print(st1.isalnum())

st2="number1"

print(st2.isalnum())

OUTPUT:

False

True

True

===============================================

1. isalpha ( ):returns True if all the characters in the strings are alphabetic letters otherwise False.

Example

INPUT:

st="Hello"

print(st.isalpha())

st1="number1"

print(st1.isalpha())

st1="My\_world"

print(st1.isalpha())

OUTPUT:

True

False

False

==========================================

1. isdigit ( ):returns True if all the characters in the strings are digits otherwise False.

Example

INPUT:

st="number1"

print(st.isdigit())

st1="4321"

print(st1.isdigit())

OUTPUT:

False

True

==================================================

1. islower ( ):returns True if all the characters in the strings are lowercase otherwise False.

Example

INPUT:

st="helloworld"

print(st.islower())

st1="Python"

print(st1.islower())

OUTPUT:

True

False

======================================================

1. isupper ( ):returns True if all the characters in the strings are uppercase otherwise False.

Example

INPUT:

st="Hello world"

print(st.isupper())

st1="PYTHON"

print(st1.isupper())

OUTPUT:

False

True

===========================================

1. join ( ): used to concatenate elements of an iterable like list tuple etc into a single string.

This method takes an iterable as an argument and returns a string.

Example

INPUT:

list = ["Welcome", "to", "Python"]

result = " ".join(list)

print(result)

OUTPUT:

Welcome to Python

================================================

1. title ( ): converts the given string titlecase i.e. the first character of each word is converted to uppercase.

Example

INPUT:

name="welcome to programming"

m=name.title()

print(m)

OUTPUT:

Welcome To Programming

============================

1. lower ( ): converts all the characters in the given string to lowercase

Example

INPUT:

name="Welcome To Programming"

m=name.lower()

print(m)

OUTPUT:

welcome to programming
==============================================

1. upper ( ):converts all the characters in the given string to uppercase

Example

INPUT:

name="welcome to programming"

m=name.upper()

print(m)

OUTPUT:

WELCOME TO PROGRAMMING

============================================

1. replace ( ): replace occurrences of a specified substring within a string with another substring.

This method takes three parameters: old\_substring, new\_substring,count. count is

Example

INPUT:

st="welcome to programming"

new\_st=st.replace(" ","\_")

print(new\_st)

OUTPUT:

welcome\_to\_programming