Strings

1. **String concatenation:**

s1 = "I am going to school"

s2 = "Today is Monday"

s3 = "Because"

print (s1 + s3 + s2)

s4 = "Cheer for Friday"

#s4 = #0123456789012345

1. **Slicing**

print (s4[6:9] + s4[13:16])

print (s4[6:9])

print(s4[-7:-1])

1. **Creating a String**

Single and double quotes are special characters. There are used to defined strings. There are actually 3 ways to define a string using either single, double or triple quotes:

str = ‘ The area of a circle is pi \* r \*r’

str = ” The area of a circle is pi\*r\*r”

str = ‘” The area of a circle is pi\*r\*r ”’

In fact the latest is generally written using triple double quotes:

str = “ ” ” The area of a circle is pi\*r\*r “ ” ”

Strings in double quotes work exactly the same as in single quotes but allow inserting single quote character inside them.

The interest of the triple quotes (‘’’ or “””) is that you can specify multi-line strings. Moreover, single quotes and double quotes can be used freely within the triple quotes:

1. **Strings are immutable.**

You can access to any character using slicing:

str= “Good Morning”

print (str[0:5])

However, you cannot change any character:

str[0] = ‘g’ # This is wrong

1. **Formatting**

One of Python's coolest features is the string format operator %. This operator is unique to strings and makes up for the pack of having functions from C's printf() family. Following is a simple example −

**print ("My name is %s and weight is %d kg!" % ('Zara', 21))**

**o/p**

**My name is Zara and weight is 21 kg!**

1. **Escape Characters**

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

str = “I am going for a morning walk”

print(“I am going for\n a morning walk”)

print(“I am going for\r a morning walk”)

print(“I am going for\t a morning walk”)

1. **Operators**

**t = 'This is a test'**

**t2 = t+t**

**t3 = t\*3**

**print(t)**

**print(t2)**

**print(t3)**

1. **Methods**
2. **Methods to query information**

**"44".isdigit() # is the string made of digits only ?**

**True**

**"44".isalpha() # is the string made of alphabetic characters only ?**

**False**

**"44".isalnum() # is the string made of alphabetic characters or digits only ?**

**True**

**"Aa".isupper() # is it made of upper cases only ?**

**False**

**"aa".islower() # or lower cases only ?**

**True**

**"Aa".istitle() # does the string start with a capital letter ?**

**True**

**text = "There are spaces but not only"**

**text.isspace() # is the string made of spaces only ?**

**False**

1. **Methods to return the modified versions of the string**

**mystr = "this is a dummy string"**

**mystr.title() # return a titlecase version of the string**

**'This Is A Dummy String'**

**mystr.capitalize() # return a string with first letter capitalised only.**

**'This is a dummy string'**

**mystr.upper() # return a capitalised version of the string**

**'THIS IS A DUMMY STRING'**

**mystr.lower() # return a copy of the string converted to lower case**

**'this is a dummy string'**

**mystr.swapcase() # replace lower case by upper case and vice versa**

**'THIS IS A DUMMY STRING'**

**str= “Jogging and walking”**

**len(str)**

**o/p = 19**