STRINGS IN PYTHON

- Represents a group of characters.
- **Str datatype**.
- Conversion of character to a number is called **encoding** and the reverse process is **decoding.**
- In python, a string is a sequence of **Unicode characters.**

How to create a string in Python?

Examples:

```
s1='Welcome to Python Learning'
s2="Welcome to Python Learning"

str=''' WElcome to Python class
   timing of the class is 11 to 12.'''
str=""" WElcome to Python class
timing of the class is 11 to 12."""

s1='WElcome to "Python" Class'
print(s1)

s1="WElcome to 'Python' Class"
print(s1)

s1="WElcome to \tPython\n Class"
print(s1)
```

- To nullify the effect of escape characters, we can create the string as a **‘raw’ string** by adding **‘r’** before the string.
- To create a string with **Unicode characters,** we should add **’u’** at the beginning of the string.

Example:

```
s1= r"Welcome to \t Python \n Learning"
print(s1)
```

Length of a String

```
str='Welcome to Python Class'
n=len(str)
print(n)
```

How to access characters in a string?

- We can access individual characters using **indexing** and a range of characters using **slicing.**

1. **Indexing in Strings**

    ○ Str[i] used to refer the i^th element of a string.
    ○ When we use index as **negative number**, it refers to elements in the reverse order.
    ○ Trying to access a character out of index range will raise an **Index Error.**
    ○ We cannot use other types; this will result into **Type Error.**
    ○ **Python does not support a character type.**

Example:

#accessing string characters in Python

```
str='Good Morning'
print('str = ', str)
print('str[0]= ',str[0])
print('str[-1]= ',str[-1])
```

```
#index must be in range
print(str[15])
#index must be an integer
print(str[1.5])
```

2. **Slicing the Strings**

    ○ Slice represents a part of a string.

Syntax: string_name[start: stop: stepsize]

Examples:

```
str='Good Morning'
print(str[0:9:1])
print(str[0:9:2])
print(str[::]) #access string from 0th to last character
print(str[::2]) #access entire string in steps 2
print(str[2::]) #access string from str[2] to ending
print(str[:4:]) #access string from str[0] to str[3] in steps of 1
print(str[-4:-1]) #access string from str[-4] to str[-2] from left to right
print(str[-8::]) #access from str[-8] till the end of the string
print(str[-1:-4:-1]) #retrieve from str[-1] to str[-3] from right to left
print(str[-1::-1]) #retrieve from str[-1] till first element from right to left
print(str[5:-2])
```

Various String Operators

Operator	**Name**	**Description**
[ ]	Slice	It gives the letter from the given index
[:]	Range Slice	It gives the characters from the given range
In	Membership	Returns true if a letter exist in the given string
Not in	Membership	Returns true if a letter exist is not in the given string
r/R	Raw String	Suppresses actual meaning if escape characters
%	Used for string format	%r – It insert the canonical string representation of the object %s - It insert the presentation string representation of the object %d – It will format a number for display
+	Concatenate	It concatenates the strings and gives the result
*	Repeat	It prints the character twice

How to change or delete a string?

- Strings are immutable.
- We can reassign different strings to the same name.
- We cannot delete or remove characters from a string.
- But deleting the string entirely is possible using the **del keyword.**

Examples:

```
str=”Hello”
print(str)
str=”Python”
print(str)

str=”Hello”
print(str)
del str
print(str)

str=”Hello”
del str[0]
Type Error: ‘str’ object doesn’t support item deletion.
```