Strings: A type of data structure in different programming languages.

Strings in python are stored as individual characters with two way index for each location.

Simplifying, it is a group of various characters with an index location for each character.

String indexing:

Forward indexing: go from left to right, start from 0 and add 1 for every character.

Backward indexing: go from right to left, start from -1 and add 1 for every character.

For eg) python

FI: 0,1,2,3,4,5

BI:-6……-1

A= “my name is anay kumar”

print(a[1st index:2nd index+1])

print(a[0:15])

>>my name is anay

How to check length of a string?

>> len(variable name)

>>len(string)

String Operators:

1. String Concatenation Operator (+) : the + operator creates a new string by joining two other individual strings.

For eg) “Akshat ” + “Gupta”

>>Akshat Gupta

1. String Replication Operator (\*) : the \* operator is used with string and an integer.

For eg) “he” \* 5

>>hehehehehe

1. Membership Operators:
2. in: returns True if a character exists in the given string

b)Not in: returns True is a character does not exist in the given string.

For eg) “a” in “anay”

>> True

“b” in “anay”

>>False

“b” not in “anay”

>>True

1. Comparison Operators:

Python’s standard comparison operators are all relational operators (>,<,>=,<=,==,!=) and they apply to string as well. The comparison operators return True or False by comparing character-by-character comparison rules for ASCII/Unicode.

A->65

a->97

for eg)

“a”==”a” >> True

“A” == “a”>> False

“A” > “a” >> False

How to check ASCII value?

ord(‘A’),ord(‘a’)

chr(66), chr(97)

>>B

String Slicing: The term ‘String Slicing’ refers to a part of the string, where strings are sliced using a range of indices.

Name= any string

Name[m:n]

>>all the characters falling between indices m and n-1.

M,m+1,m+2……n-1

For eg)

a= “Hello World”

for any index n, s[:n]+s[n:] will return the original string.

s=“amazing”

s[:-4]+[-4:]

>> “amaz” + “ing”

>> amazing

s= “my name is anay”

print(s[0:10:2])

>> m aei

s[::-1] is an easy way to print the original string in reverse.

PYTHON FUNCTIONS FOR STRINGS:

Python offers many built-in functions and methods for string Manipulation. The string manipulation methods that we will discuss can be used using the following syntax:

<stringVar>.<functionName>()

Str= “Akshat”

Str.length()

>>6

1. <StrVar>.length(): It returns the length of the string stored in a variable.
2. <StrVar>.capitalize(): Returns a copy of the string <StrVar> with its first letter capitalized.

For eg)

A= “nilaksh”

a.capitalize()

>>Nilaksh

1. <str>.isalnum(): Returns True if the characters in the string <str> are alphanumeric (alphabets or numbers), returns False otherwise.
2. <str>.isalpha(): Returns True if all the characters in the string <str> are alphabets.
3. <str>.isdigit(): Returns True if all the characters in the string <str> are digits/numbers,otherwise it returns False.

s1= ‘abc123’

s2= ‘hello’

s3= ‘12345’

s4= ‘ ‘

s3.isdigit()

1. <str>.find(sub,FI,LI): Returns the lowest index in the string <str> where the substring sub is found within the slice range (FI to LI-1).Returns -1 if sub is not found.

for eg)

string1= ‘my name is Akshat Gupta’

subs= ‘is’

string1.find(subs, 12, 18)

>> -1

string1.find(subs, 0, 18)

>>8

string1= ‘it goes as – ri**n**ga ringa **r**oses’

subs= ‘ringa’

string1.find(subs,15,22)

>> -1

String1.find(subs,15,25)

>>19