**String Handling**

**Day-01**

**28-03-2025**

**===============**

-> String in python is one of the pre-defined or built-in datatype.

-> for the string, there is a pre-defined class 'str'.

String Definition:

-------------------

In python, string is group of characters which must be enclosed with single quotes ('') or double quotes ("") or triple quotes (''' ''').

Note:

-----

In python, there is no character data representation. The character in python also the string type.

s1 = 'a'

s2 = 'abcd'

print(type(s1))

print(type(s2))

s3 = "b"

s4 = "bcde"

print(type(s3))

print(type(s4))

s5 = '''s'''

s6 = '''stop'''

print(type(s5))

print(type(s6))

s7 = """Python"""

print(type(s7))

Multi-line string:

------------------

Multi-line string can always define with triple quotes only.

lines = '''

Good Evening.

Welcome To Ashok IT.

As part of today's, we are

learning about strings.

'''

l1 = (''

'afbcde'

'akhdf')

print(type(lines))

print(lines)

print(l1)

-> While defining the string, we can write single quote text within the double quotes or double quote text within the single quotes. But we cannot allow to write single quote text in the single quote or double quote text in the double quote.

l1 = "Python is 'easy' Language."

l2 = 'Python is "easy" language'

# l3 = 'Python is 'easy' language'

# l3 = "Python is "easy" language"

print(l1)

print(l2)

How to access the characters of the string:

-------------------------------------------

-> to access the characters of the string, we can use "indexing".

-> Index is also called as "radix" or "sub-script".

-> In python, the indexing can be defined in two types:

1) Positive indexing

2) Negative indexing

1) Positive Indexing:

--------------------

-> When we need to access the characters of the string from left to right, then we can use "positive indexing".

And this access is called as "Forward access".

-> Positive indexing can always start with: "0"

and ends with: "number-of-character-of-string - 1"

Ex: string is with 10 characters:

index range ==> 0 to 9

2) Negative Indexing:

----------------------

When we need to access the characters of the string from right to left, we can use "negative indexing".

This access is called as "reverse access".

-> Negative indexing can start at: -1

and ends with: -total-no-of-characters

Ex: string with 10 characters:

negative index range ==>-1 to -10

Syntax:

string-data-name[index-value]

string = "Python"

print("The character at index-0 is = ",string[0])

print("The character at index-1 is = ",string[1])

print("The character at index-2 is = ",string[2])

print("The character at index-3 is = ",string[3])

print("The character at index-4 is = ",string[4])

print("The character at index-5 is = ",string[5])

# print(string[6])

print("The character at index -1 is = ",string[-1])

print("The character at index -2 is = ",string[-2])

print("The character at index -3 is = ",string[-3])

print("The character at index -4 is = ",string[-4])

print("The character at index -5 is = ",string[-5])

print("The character at index -6 is = ",string[-6])

len():

----

-> is a pre-defined method can use to return the length of the string.

Syntax:

len(string-data-name)

Traversing on the string:

-------------------------

1) using while loop

-------------------

**"""**

**Write a python program to access all the characters of the string along with positive index and**

**negative index values using while loop.**

**Sample output:**

**The character at positive index 0 and negative index -10 is = C**

**"""**

s = "Python is easy"

index = 0 # initialization

while index < len(s):

print("The character at positive index",index,"and negative index",index - len(s),"is = ",s[index])

index += 1

2) using for loop

-----------------

Syntax:

for iterable in string-data:

loop body

s = "Python"

index = 0

for i in s:

print("The character at positive index",index,"and at negative index",index-len(s),"is = ",i)

index += 1

Assignment:

-----------

Write a python program to define the string and print all the characters of the string which are at even places.