**What is STRING??**

1.A string is a sequence of characters. e.g. English language has 26 characters.

So, String data type allows programmer to use character or txt in their program.

2. In Python, a string is a sequence of Unicode characters

Unicode was introduced to include every character in all languages e.g. Hindi/Russian and bring uniformity in encoding.

* Unicode International standard, provides a unique way to define every character in every spoken language of the world by assigning it a unique number. The Unicode standard is maintained by the Unicode Consortium and defines more than 1,40,000 characters from more than 150 modern and historic scripts along with emoji.

e.g.print('संदीप') #hindi fonts works in python

**How it is represented?**

Strings in python are surrounded by either **single quotation** marks, or **double quotation** marks as well as triple quotation marks.

'hello' is the same as "hello".

## Multiline Strings

You can assign a multiline string to a variable by using three quotes:

a = """Lorem ipsum dolor sit amet,  
consectetur adipiscing elit,  
sed do eiusmod tempor incididunt  
ut labore et dolore magna aliqua."""  
print(a)

**Assign String to a Variable**

a = "Hello"  
print(a)

**Access String:**

Python strings are stored one by one index wise

a = "Hello, World!"

**forward indexing**

print(a[1]) #output : e

**backward indexing**

print(a[-1]) #output: !

print(a[30]) #Error-Index out of range

**Slicing**

It returns a range of characters by using the slice syntax.

b = "Hello, World!"  
print(b[2:5]) #output: llo

b = "Hello, World!"  
print(b[:5]) #output: Hello

b = "Hello, World!"  
print(b[2:]) #output: llo, World!

b = "Hello, World!"  
print(b[-5:-2]) #output: orl

**Strings are immutable**. This means that elements of a string cannot be changed once they have been assigned. We can simply reassign different strings to the same name.

my\_string = 'programmer'

my\_string[5] = 'a'

Error: str object does not support item assignment

We cannot delete or remove characters from a string. But deleting the string entirely is possible using the del keyword.

del my\_string[1]

Error- 'str' object doesn't support item deletion

But del my\_string is possible

**Traverse:**

**Loop on String:**

for x in "banana":  
  print(x)

output:

b

a

n

a

n

a

**Length:**

a = "Hello, World!"  
print(len(a))

Output: 13

## Operators :

## Membership Operator

To check if a certain phrase or character or substring is present in a string, we can use the keyword in.

True/False

txt = "work real hard and get success”  
print("get" in txt)

**Concatenate Operator**

a = "Hello"  
b = "World"  
c = a + b  
print(c)

Output: HelloWorld

Add space

a = "Hello"  
b = "World"  
c = a + " " + b  
print(c)

Output: Hello World

**Replication Operator**

# using \*

print(3\*'Hello')

Output: HelloHelloHello

**Comparison Operator:**

print("a"=="a") #True

print("a"!="a") ( A and a are not same because it has different ASCII code correspondingly)

**Concatenate(Formating):**

age ='Fifty'

txt = "My name is John, I am " **+age**

print(txt)

print(type(txt)) #Tuple/String

Note:For**+age** gives string type and for **,age** gives tuple

We can add two strings together

**But we can not add int and string toether**

**Format method**

But we can combine strings and numbers by using the format() method!

age = 50  
txt = "My name is John, and I am {}years old"  
print(txt.format(age))

The format() method takes unlimited number of arguments, and are placed into the respective placeholders:

quantity = 3  
itemno = 567  
price = 49.95  
myorder = "I want {} pieces of item {} for {} dollars."  
print(myorder.format(quantity, itemno, price))

You can use index numbers {0} to be sure the arguments are placed in the correct placeholders:

quantity = 3  
itemno = 567  
price = 49.95  
myorder = "I want to pay {2} dollars for {0} pieces of item {1}."  
print(myorder.format(quantity, itemno, price))

**Escape Characters**

\n : New line

\t : Tab space

\r :return carriage

e.g. print("Hello How are you? \rHi...")

Output: Hi... How are you?

To insert characters that are illegal in a string, use an escape character.

txt = "We are the so-called "Vikings" from the north."(This is not allowed in Python)

By using formating ,it can be possible

txt = "We are the so-called \"Vikings\" from the north."

Print Like this:

M=’Hello I’m Sandip’ #(Not allowed)

m='Hello I\'m Sandip' #(use backslash)

m=”Hello I ‘m Sandip”

print(m)

**Built-in methods**

**Upper()**

To convert string in uppercase

a = "Hello, World!"  
print(a.upper())

Output: HELLO, WORLD!

**Capitalize()**

converts first letter of string in capital

txt = "hello, and welcome to my world."  
  
x = txt.capitalize()  
  
print (x)

Output: Hello, and welcome to my world!

**Lower()**

Convert string in lowercase

a = "Hello, World!"  
print(a.lower())

output: hello, world!

**Casefold()**

*is used to convert string into lowercase*

txt = "Hello, And Welcome To My World!"  
  
x = txt.casefold()  
print(x)

Output: hello, and welcome to my world!

Swapcase()

Will convert upper case to lowercase and lowercase to uppercase

my\_var12=**"This IS BrainWORKS"**print(**"Swap case:"**,my\_var12.swapcase())

*Output:*

Swap case: tHIS is bRAINworks

**Length**

Gives length of string

H13="JHGFHHHHGBHHHHDSSSSSSSSSSSSSS VBJMV NNNNNNNNNNNNN"

print(len(H13))

Output: 49

**Strip**

The strip() method removes any whitespace from the beginning or the end:

a = " Hello, World! "  
print(a.strip())

Output:

“Hello, World!”

**lstrip()**

removes whitespace from left of string

H18=" Hii"

print(H18.lstrip())

Output:

“Hii”

**rstrip**

removes whitespace from right of string

H19="Alll "

print(H19.rstrip())

Output:”Alll”

**strip unwanter characters**

H55=("abcaaaaaabaabcjjjjabckl")

print(H55.strip('abc'))

Output: jjjjabckl

**Enumerate()**

Enumerate gives each character of string with corresponding index location

H27="HELLO"

print(list(enumerate(H27)))

Output: [(0, 'H'), (1, 'E'), (2, 'L'), (3, 'L'), (4, 'O')]

**Replace()**:

Replace given substring with another substring

a = "Hello, World!"  
print(a.replace("H", "J"))

Output:Jello,World!

H22="I am SUCCESSFUL"

print(H22.replace("SUCCESSFUL","ENTREPRENEUR"))

Output: I am ENTREPRENEUR

**Split:**

The split() method splits the string into substrings if it finds instances of the separator:

a = "Hello, World!"  
print(a.split(","))

Output: ['Hello', ' World!']

#by default separate substring by whitespace

#or if we provide comma or other delimiter then accordingly will split

H25="Today is going to be great day"

print(H25.split())

Output: ['Today', 'is', 'going', 'to', 'be', 'great', 'day']

H25="Today , is going to be great day"

print(H25.split(","))

Output: ['Today', ' is going to be great day']

**Join**

text = ['Python', 'is', 'a', 'fun', 'programming', 'language']

# join elements of text with space

print(' '.join(text))

Output: Python is a fun programming language

**Count()**  
*# Count returns number of times character repeats in string*my\_var2 = **"Hello How are you Harry"**

print(my\_var2.count(**"H"**)) #returns how many times ‘H’appeared

Output:3

print(my\_var2.count(**"H"**, 0, 5)) #returns How many times ‘H’ appeared depending upon range provided

Output:1  
 *#* ***endswith()***

*if string ends with specific value, true or false*end\_with = **"This is Python session and we are learning python"**result = end\_with.endswith(**"python"**)  
print(result)  
Output: True

result1 = end\_with.endswith(**"python"**, 0, 20) *# start from first till 20th index ,Check if it ends with python then true or false*print(result1)

Output:False  
  
***startswith()***

*if string Starts with specific substring then true or else false*my\_var13=**"This is good"**print(my\_var13.startswith(**"This"**))

Output:True

***find()*** *# use to search specific string value and returns position(index)*

*#returns -1 if does not find*

H4="Believe in yourself"

print(H4.find("in"))

Output:8

print(H4.find("inn")) #returns -1 if not found

**index()**

H5="I am going to be very rich"

print(H5.index("very")) #returns index location of ‘very’

Output:17

print(H5.index("verys")) #returns ValueError if substring not found

***title()*** *Returns string in title case/initcap/pascal case*

#It means first letter is capital of each words in string

my\_var6 = **"welcome to amazon"**print(my\_var6.title())

Output: Welcome To Amazon

*# Encode() returns encoded version of string ,encoding to utf-8*endcode\_example = **"Testing Shastra"**print(endcode\_example.encode())

Output: b'Testing Shastra'

**Max()**  
ma = **"Hi there Everyone"**print(max(ma))

Output:y

**Min()**  
mi = **"helloafriendz"**print(**"Min:"**, min(mi))  
Output:a

#Note:Depending upon ASCII values it will rank characters

e.g. First whitespace 🡪Numbers🡪Upper case characters🡪Lower case characters

**Center()**

The center() method will center align the string, using a specified character (space is default) as the fill character.

txt = "banana"

x = txt.center(20)

print(x)

**Output: “ banana “**

S5="Sandip"

print(S5.center(10,'\*'))

Output: \*\*Sandip\*\*

**Ljust /Rjust()**

#Left justifies

#Right justified

Padding of characters

H14="yes"

print(H14.ljust(5,'\*'))

output: yes\*\*

print(H14.rjust(15,'\*'))

Output:

\*\*\*\*\*\*\*\*\*\*\*\*yes

**Zfill**()

Right padded with zeroes(by default zeroes will be taken)

H15="www"

print(H15.zfill(8))

Output: 00000www

**isalnum()**

*use to check string has numeric+letters,returns true else false*

*Note -If you give space also ,it will return False*my\_var4 = **"Testing11"**print(my\_var4.isalnum())

Output: True  
  
**isalpha()** *# Returns true if specified string contains all alphabets else false*my\_var5 = **"TestinhS70"**print( my\_var5.isalpha())

Output:False #because it has number in it

isdecimal()  
isdigit()  
my\_var7 = **"33"**print(**"Isdecimal:"**, my\_var7.isdecimal())  
print(**"is digit:"**, my\_var7.isdigit())

Isnumeric()  
  
**isspace()**my\_var8 = **" "**print(my\_var8.isspace())  
Output:True

**Istitle()**

print(**"title:"**, my\_var6.istitle()) #true/false

**isupper()**my\_var10 = **"UPPER "**print(my\_var10.isupper())

Output:True

**islower()**

H10="hello"

print(H10.islower()) #Every characters should be in lower case

Programs:

**#Palindrome String**

def isPalin(s):

if s==s[::-1]:

print("String is palindrome..")

else:

print("Not palindrome..")

s=input("Enter string..")

isPalin(s)

Output:

Enter string..level

String is palindrome..

**OR**

s =input("Enter String:")

y=("palindrome" if s==s[::-1] else "not")

print(y)

**Program: Print index of string**

s="Hello everyone"

i=0

while(i<len(s)):

print("index:",i,"value:",s[i])

i+=1

Output:

index: 0 value: H

index: 1 value: e

index: 2 value: l

index: 3 value: l

index: 4 value: o

index: 5 value:

index: 6 value: e

index: 7 value: v

index: 8 value: e

index: 9 value: r

index: 10 value: y

index: 11 value: o

index: 12 value: n

index: 13 value: e

**#reversed**

s="Hello How are you doing?"

for i in reversed(s):

print(i)

Or

t=reversed(s)

print(list(t))

**Program**:

#To check if substring is present in string

s=input("Enter String..")

x=input("Enter substring..")

if x in s:

print("Substring is present")

else:

print("Not present")

Output:

Enter String..Brainworks

Enter substring..Brain

Substring is present

**#substring present**

str=input("Enter string..")

str1=input("Enter substring..")

y=("Present" if str1 in str else "Not present")

print(y)

Output:

Enter string..Pineapple

Enter substringapple

Present

**Program**:

#characters at even and odd position

s=input("Enter String:")

print("Even position",s[0::2])

print("Odd position",s[1::2])

Output:

Enter String:Good Evening

Even position Go vnn

Odd position odEeig

**Remove** **substring**:

s="Hello India"

print(s.replace("e"," "))

Output:

H llo India

#**Program to check Vowels present and count it**

vowels=['a','e','i','o','u','A','E','I','O','U']

str=input("Enter string..")

count\_v=0

count\_c=0

for i in str:

if i in vowels:

count\_v=count\_v+1

else:

count\_c=count\_c+1

print(count\_v)

print(count\_c)

Output:

Enter string..Maharashtra

4

7

**#Remove duplicates from string**

str1=input("Enter string 1:")

str2=input("Enter string 2:")

d=""

for i in str2:

if i in str1:

d=d+i

else:

pass

print(d)

Output:

Enter string 1:I am learning python

Enter string 2:python

python

**#Remove Duplicate characters,print unique**

s = 'Hello How are Hun'

m=set(s)

print(m)

Output:

{' ', 'w', 'r', 'u', 'e', 'a', 'l', 'o', 'H', 'n'}

**#Traverse using for loop + Reverese traverse**

s="Hello"

for i in s:

print(i)

#print(s[::-1])

for i in reversed(s):

print(i)

**Program: How many specific letters are present in string**

# Iterating through a string

count = 0

for letter in 'Hello World':

if(letter == 'l'):

count += 1

print(count,'letters found')

Output:

2 letters found