# String

## *STRING DATA TYPE*

The most commonly used object in any project and in any programming language is String

### How to define multi-line String Literals?

We can define multi-line String literals by using triple single or double quotes.

```
multi_line_str = """Any sequence of characters
within either single quotes
or double quotes is considered
as a String."""
print(multi_line_str)
```

### How to Access Characters of a String?

We can access characters of a string by using the following ways.

1. By using index
2. By using slice operator

## 1) Accessing Characters By using Index

- Python supports both +ve and -ve Index.

```
name = "Ashish Bindra"

print(name[0]) # A
print(name[5]) # h
print(name[-1]) # a
print(name[20])
IndexError: string index out of range
```

## 2) Accessing Characters by using Slice Operator

- ***Syntax:*** name bEginindex:endindex:step]
- ***Begin Index:*** From where we have to consider slice (substring)
- ***End Index:*** We have to terminate the slice (substring) at endindex-1
- ***Step:*** Incremented Value.

```
str1 = "Learning Python is very very easy!!!"

print(str1[1:7:1]) # earnin
print(str1[1:7])   # earnin
print(str1[1:7:2]) # eri
print(str1[:7])    # Learnin
print(str1[7:])    # g Python is very very easy!!!
print(str1[::])    # Learning Python is very very easy!!!
print(str1[:])     # Learning Python is very very easy!!!
print(str1[::-1])  # !!!ysae yrev yrev si nohtyP gninraeL
```

### Mathematical Operators for String

We can apply the following mathematical operators for Strings.

- operator for concatenation

2) * operator for repetition

```
print("Hello" + "World") # HelloWorld
print("Hi"*2)            # HiHi
```

## Note

1. To use + operator for Strings, compulsory both arguments should be str type.
   2) To use * operator for Strings, compulsory one argument should be str and other argument should be int.

## Q) Write a Program to access each Character of String in Forward and Backward Direction by using while Loop?

```
s = "Learning Python is very easy !!!"
n = len(s)
i = 0

print("Forward direction")

while i<n:
    print(s[i],end=' ')
    i +=1

print("\n\nBackward direction")
i = -1

while i >= -n:
    print(s[i],end=' ')
    i = i-1


# o/p
# Forward direction
# L e a r n i n g   P y t h o n   i s   v e r y   e a s y   ! ! !

# Backward direction
# ! ! !   y s a e   y r e v   s i   n o h t y P   g n i n r a e L
```

Alternative ways:

```
s = "Learning Python is very easy !!!"

print("Forward direction")
for i in s[::]:
    print(i,end=' ')

print("\nBackward direction")
for i in s[::-1]:
    print(i,end=' ')
```

## Checking Membership

We can check whether the character or string is the member of another string or not by using in and not in operators.

```
str1 = 'durga'

print('d' in str1) # True
print('z' in str1) # False


str1 = input("Enter main string: ")
subs = input("Enter sub string: ")

if subs in str1:
    print(f"\"{subs}\" is found in main string")
else:
    print(f"\"{subs}\" is not found in main string")
# Enter main string: Ashish Bindra
# Enter sub string: sh
# "sh" is found in main string
```

## Comparison of Strings

- We can use comparison operators (<, <=, >, >=) and equality operators (==, !=) for strings.
- Comparison will be performed based on alphabetical order.

```
s1=input("Enter first string: ")
s2=input("Enter Second string: ")

if s1==s2:
    print("Both strings are equal")
elif s1<s2:
    print("First String is less than Second String")
else:
    print("First String is greater than Second String")
```

o/p
Enter first string: durga
Enter Second string: durga Both strings are equal

## Removing Spaces from the String

We can use the following 3 methods

1. rstrip(): To remove spaces at right hand side
   2) lstrip(): To remove spaces at left hand side
   3) strip(): To remove spaces both sides

```
city=input("Enter your city Name: ")
scity=city.strip()

if scity=='Hyderabad':
    print("Hello Hyderbadi..Adab")
elif scity=='Chennai':
    print("Hello Madrasi...Vanakkam")
elif scity=="Bangalore":
    print("Hello Kannadiga...Shubhodaya")
else:
    print("your entered city is invalid")

# Enter your city Name:Bangalore
# Hello Kannadiga...Shubhodaya
```

## Finding Substrings

We can use the following 4 methods
For forward direction:

1. find()
   2) index()

For backward direction:

1. rfind()
   2) rindex()

## find()

***s.find(substring)***
Returns index of first occurrence of the given substring. If it is not available then we will get -1.

```
s="Learning Python is very easy"

print(s.find("Python")) #9
print(s.find("Java")) # -1
print(s.find("r"))#3
print(s.rfind("r"))#21
```

## s.find(substring,bEgin,end)

It will always search from bEgin index to end-1 index.

```
s="durgaravipavanshiva"

print(s.find('a'))#4
print(s.find('a',7,15))#10
print(s.find('z',7,15))#-1
```

## index()

index() method is exactly same as find() method except that if the specified substring is not available then we will get ValueError.

```
s=input("Enter main string:")
subs=input("Enter sub string:")

try:
    n=s.index(subs)
except ValueError:
    print("substring not found")
else:
    print("substring found")
```

o/p
Enter main string:learning python is very easy Enter sub string:python substring found

## Counting substring in the given String

We can find the number of occurrences of substring present in the given string by using count() method.

1. s.count(substring): It will search through out the string.
2. s.count(substring, bEgin, end): It will search from bEgin index to end-1 index.

```
s="abcabcabcabcaddab"

print(s.count('a'))     # 6
print(s.count('ab'))    # 5
print(s.count('abc'))   # 4
print(s.count('a',3,7)) # 2
```

## Replacing a String with another String

s.replace(oldstring, newstring) inside s, every occurrence of old String will be replaced with new String.

```
s = "Learning Python is very difficult"

s1 = s.replace("difficult","easy")
print(s1) # Learning Python is very easy

s = "abababababbab"

replace_str = s.replace("a","b")
print(replace_str)
```

## Q) String Objects are Immutable then how we can change the Content by using replace() Method

- Once we creates string object, we cannot change the content.This non changeable behaviour is nothing but immutability.
- If we are trying to change the content by using any method, then with those changes a new object will be created and changes won't be happend in existing object.
- Hence with replace() method also a new object got created but existing object won't be changed.

```
str1 = "abab"
str2 = str1.replace("a","b")

print(str1,"is available at :",id(str1))
```

```
print(str2,"is available at :",id(str2))
```

o/p

```
abab is available at : 2604224914096<br>
bbbb is available at : 2604225153904
```

## Splitting of Strings

- We can split the given string according to specified seperator by using split() method.
- l = s.split(seperator)
- The default seperator is space. The return type of split() method is List.

```
s="durga software solutions"

l=s.split()
for x in l:
    print(x)
# o/p
# durga
# software
# solutions

date = "22-07-2024"
date_list = date.split('-')

for val in date_list:
    print(val)
# o/p
# 22
# 07
# 2024
```

## Joining of Strings

We can join a Group of Strings (List OR Tuple) wrt the given Seperator.

- s = seperator.join(group of strings)

```
names = ('sunny', 'bunny', 'chinny')
list_name = '-'.join(names)
print(list_name) # sunny-bunny-chinny
```

## Changing Case of a String: We can change case of a string by using the following 4 methods

1. upper(): To convert all characters to upper case
   2. lower(): To convert all characters to lower case
   3. swapcase(): Converts all lower case characters to upper case and all upper case characters to lower case
   4. title(): To convert all character to title case. i.e first character in every word should be upper case and all remaining characters should be in lower case.
   5. capitalize(): Only first character will be converted to upper case and all remaining characters can be converted to lower case
   6. casefold(): Return a version of the string suitable for caseless comparisons.

```
line = 'learning Python is very Easy'

print(line.upper())      # LEARNING PYTHON IS VERY EASY
print(line.lower())      # learning python is very easy
print(line.swapcase())   # LEARNING pYTHON IS VERY eASY
print(line.title())      # Learning Python Is Very Easy
print(line.capitalize()) # Learning python is very easy
print(line.casefold())   # learning python is very easy
```

## Checking Starting and Ending Part of the String

Python contains the following methods for this purpose

1. s.startswith(substring)
2. s.endswith(substring)

```
line = 'learning Python is very easy'

print(line.startswith('learning')) # True
print(line.endswith('learning'))   # False
print(line.endswith('easy'))       # True
```

## To Check Type of Characters Present in a String: Python contains the following methods for this purpose

1. isalnum(): Returns True if all characters are alphanumeric( a to z , A to Z ,0 to9 )
2. isalpha(): Returns True if all characters are only alphabet symbols(a to z,A to Z)
3. isdigit(): Returns True if all characters are digits only( 0 to 9)
4. islower(): Returns True if all characters are lower case alphabet symbols
5. isupper(): Returns True if all characters are upper case aplhabet symbols
6. istitle(): Returns True if string is in title case
7. isspace(): Returns True if string contains only spaces

```
print('Durga786'.isalnum()) # True
print('durga786'.isalpha()) # False
print('durga'.isalpha()) # True
print('durga'.isdigit()) # False
print('786786'.isdigit()) # True
print('abc'.islower()) # True
print('Abc'.islower()) # False
print('abc123'.islower()) # True
print('ABC'.isupper()) # True
print('Learning python is Easy'.istitle()) # False
print('Learning Python Is Easy'.istitle()) # True
print(' '.isspace()) # True


s=input("Enter any character:")

if s.isalnum():
    print("Alpha Numeric Character")
    if s.isalpha():
        print("Alphabet character")
        if s.islower():
            print("Lower case alphabet character")
        else:
            print("Upper case alphabet character")
    else:
        print("it is a digit")
elif s.isspace():
    print("It is space character")
elif not s.isalnum():
    print("it is a float")
else:
    print("Non Space Special Character")
```

## Important Programs regarding String Concept

## Q1) Write a Program to Reverse the given String

- Input: durga
- Output: agrud

```
str1 = input("Enter Some String: ") # ashish
print(str1[::-1]) # hsihsa

str1 = input("Enter Some String: ") # Python
print(''.join(reversed(str1))) # nohtyP

str1 = input("Enter Some String: ") # Python

i = len(str1) - 1
target=''

while i >= 0:
    target = target + str1[i]
    i = i - 1

print(target) # nohtyP
```

"4st way"

```
s = "hello"
reversed_s = ""

for char in s:
    reversed_s = char + reversed_s

print(reversed_s)  # Output: "olleh"
```

"Using a List Comprehension

```
s = "hello"
reversed_s = ''.join([s[i] for i in range(len(s)-1, -1, -1)])
print(reversed_s)  # Output: "olleh"


def reverse_recursive(s):
    if len(s) == 0:
        return s
    else:
        return reverse_recursive(s[1:]) + s[0]

s = "hello"
reversed_s = reverse_recursive(s)
print(reversed_s)  # Output: "olleh"


from functools import reduce

s = "hello"
reversed_s = reduce(lambda acc, char: char + acc, s)
print(reversed_s)  # Output: "olleh"
```

## Q2) Program to Reverse Order of Words

- Input: Learning Python is very Easy
- Output: Easy Very is Python Learning

```
line=input("Enter Some String: ")  # Learning Python is very Easy

list_line = line.split()
l1 = []
i = len(list_line) - 1

while i >= 0:
    l1.append(list_line[i])
    output = ' '.join(l1)
    i = i - 1
```

```
print(output) # Easy very is Python Learning

line=input("Enter Some String: ")  # Learning Python is very Easy

list_line = line.split()
list_line.reverse()

print(" ".join(list_line)) # Easy very is Python Learning
```

## Write a Program to find the Number of Occurrences of each Character present in the given String?

- Input: ABCABCABBCDE
- Output: A-3,B-4,C-3,D-1,E-1

```
str1 = input("Enter the Some String: ")

d = {}
for x in str1:
    if x in d.keys():
        d[x] = d[x] + 1
    else:
        d[x] = 1

for k, v in d.items():
        print(f"{k} = {v} Times")

Enter the Some String: Ashsih
A = 1 Times
s = 2 Times
h = 2 Times
i = 1 Times


print(type(string_data))

value = bytes("ashish".encode())

print(value, type(value))
```