Strings-->> It is a type of data type comprising of sequence of characters. These are immutable in nature.It can be implemented with inverted commas " ". It is case senstive in nature. There are various methods of strings-->>

1.upper() : The upper() method converts a string to upper case.

2.lower() The lower() method converts a string to lower case.

3.strip() : The strip() method removes any white spaces before and after the string.

4.rstrip() : the rstrip() removes any trailing characters

5.replace() : The replace() method replaces all occurences of a string with another string

6.split() : The split() method splits the given string at the specified instance and returns the separated strings as list items.

7.capitalize() : The capitalize() method turns only the first character of the string to uppercase and the rest other characters of the string are turned to lowercase. The string has no effect if the first character is already uppercase.

8.center() : The center() method aligns the string to the center as per the parameters given by the user.

9.count() : The count() method returns the number of times the given value has occurred within the given string. 10.endswith() : The endswith() method checks if the string ends with a given value. If yes then return True, else return False.

11.find() : The find() method searches for the first occurrence of the given value and returns the index where it is present. If given value is absent from the string then return -1.

12.index() : The index() method searches for the first occurrence of the given value and returns the index where it is present. If given value is absent from the string then raise an exception.

13.isalnum() : The isalnum() method returns True only if the entire string only consists of A-Z, a-z, 0-9. If any other characters or punctuations are present, then it returns False.

16.isalpha() : The isalnum() method returns True only if the entire string only consists of A-Z, a-z. If any other characters or punctuations or numbers(0-9) are present, then it returns False.

17.islower() : The islower() method returns True if all the characters in the string are lower case, else it returns False.

18.isprintable() : The isprintable() method returns True if all the values within the given string are printable, if not, then return False.

19.isspace() : The isspace() method returns True only and only if the string contains white spaces, else returns False.

20.istitle() : The istitile() returns True only if the first letter of each word of the string is capitalized, else it returns False.

21.isupper() : The isupper() method returns True if all the characters in the string are upper case, else it returns False.

22.startswith() : The endswith() method checks if the string starts with a given value. If yes then return True, else return False.

23.swapcase() : The swapcase() method changes the character casing of the string. Upper case are converted to lower case and lower case to upper case.

24.title() : The title() method capitalizes each letter of the word within the string.

25 .expandtabs()-->> It usesthe amount of space as argument to provide tab between each string word.For default it is 8 spaces.

Practice Question-->>

1. String Manipulation: Write a Python program that reverses a given string. Given a string, write a function to count the number of vowels in it. Write a program to check if a string is a palindrome. Implement a function to capitalize the first letter of each word in a sentence.

```
str1="Hello World"
print(str1[::-1])

def vowels(string):
  count=0
  for i in string:
    if(i=="a" or i=="e" or i=="i" or i=='o' or i=="u"):
      count+=1
  print(count)
vowels("aeiof")

def reverse(str1):
  str2=str1[::-1]
  if(str1==str2):
    print("Palindrome")
  else:
    print("Not Palindrome")
reverse("101")

def capitalize():
  str1="hello world"
  str2=str1.title()
  print(str2)
capitalize()
```

```
dlroW olleH
4
Palindrome
Hello World
```

1. String Formatting: Create a formatted string using f-strings that includes variables and their values. Write a program to left-align a list of strings with a specified width. Given a list of names, format them into a bulleted list using string concatenation or join method.

```
str1="hello"
str2="bye"
print(f"The variable is {str1} and value is {str2}")
str3="Hello"
print(str3.center(50))
names = ["Alice", "Bob", "Charlie", "David"]

bulleted_list = ""

for name in names:
    bulleted_list += f"• {name}\n"

print(bulleted_list)
```

```
The variable is hello and value is bye
                      Hello
• Alice
• Bob
• Charlie
• David
```

1. String Searching and Counting: Write a function to find the index of the first occurrence of a substring in a given string. Count the occurrences of a specific word in a paragraph. Write a program to extract all email addresses from a given text.

```
def show():
  str1="hello where are you are"
  print(str1.find("are"))
show()

def count():
  word="abacdcvxsa"
  print(word.count("a"))
count()

def email():
```

```
12
3
```

1. String Slicing and Concatenation: Given a sentence, extract the last three words. Concatenate two strings with a space in between. Write a program that removes leading and trailing whitespaces from a string.

```
 str1="Hello World"
print(str1[-3:])
str2="No where"
print(str1+" "+str2)
str3="   hello   "
print(str3.strip())
```

```
rld
Hello World No where
hello
```

OPERATORS-->Use to perform various operation on basis of certain conditions. 1.Assignment 2.Airthematic 3.Bitwise 4.Comparison 5.Identity 6.Logical--> is , is not 7.Membership Operator-->in, not in There is also a function called id() function use to find memory location of specifc string.

PRACTICE -->> 1. Given a string containing a sentence, use a string method to convert the entire string to uppercase.

```
str1="Hello World"
print(str1.upper())
```

```
HELLO WORLD
```

2. You have a string with extra spaces at the beginning and end. Use a string method to remove these extra spaces.

```
str1="   Hello World   "
print(str1.strip())
```

```
Hello World
```

3. Given a string, find the first occurrence of the substring "Python" and print its position.

```
str1="Find the first occurrence of the substring Python"
print(str1.find("Python"))
```

```
43
```

4. You have a string with words separated by commas. Use a string method to split the string into a list of words.

```
string="Hello,World,Python"
print(string.split(","))
```

```
['Hello', 'World', 'Python']
```

5. Given a string containing a sentence, replace all occurrences of the word "Java" with "Python".

```
string="Hello,World,Python"
print(string.replace("Python","java"))

Hello,World,java
```

6. You have a string containing a mix of uppercase and lowercase characters. Use a string method to convert the entire string to lowercase.

```
string="Hello,World,Python"
print(string.lower())

hello,world,python
```

7. Given a string, check if it starts with the prefix "Hello".

```
string="Hello,World,Python"
print(string.startswith("Hello"))

True
```

8. You have a string with numbers and letters. Use a string method to check if the string is alphanumeric

```
string="Hello,World,Python"
print(string.isalnum())

False
```

9. Given a string containing multiple words, use a string method to join the words with a hyphen ("-").

```
words = ["Hello", "World", "Python"]

hyphen_separated_string = "-".join(words)

print(hyphen_separated_string)

Hello-World-Python
```

10. You have a string with different words separated by spaces. Use a string method to count the number of words in the string.

```
str1="Find the first occurrence of the substring Python"
print(len(str1.split()))

8
```

11. Given a string containing a sentence, use a string method to capitalize the first letter of each word.

```
str1="find the first occurrence of the substring Python"
print(str1.title())
```

```
Find The First Occurrence Of The Substring Python
```

12. You have a string containing a URL. Use a string method to check if the URL ends with ".com".

```
str="https://www.google.com"
print(str.endswith(".com"))
```

```
True
```

13. Given a string containing a list of names separated by commas, use a string method to sort the names alphabetically.

```
names="Alice,Bob,Charlie,David"
print(sorted(names.split(",")))
```

```
['Alice', 'Bob', 'Charlie', 'David']
```

14. You have a string containing a sentence. Use a string method to find the number of times the word "data" appears in the string.

```
str1="Find the first occurrence of the substring Python"
print(str1.count("data"))
```

```
0
```

16. You have a string containing multiple email addresses separated by semicolons. Use a string method to extract the domain of each email address.

```
string="Hello@gmail.com ; Wolrd2@gmail.com; Python@gmail.com"
string3=string.split(";")
for i in string3:
  print(i.split("@")[1])
```

```
gmail.com
gmail.com
gmail.com
```

7. Given a string containing a paragraph, use a string method to replace all newline characters ("\n") with spaces.

```
str1="Hello\nWorld\nPython"
print(str1.replace("\n"," "))
```

```
Hello World Python
```

18. You have a string that contains both letters and digits. Use a string method to extract only the digits from the string.

```python
str1="mehak123jacky234"
for i in str1:
  if i.isdigit():
    print(i,end=" ")
```

```
1 2 3 2 3 4
```

22. You have a string with multiple lines of text. Use a string method to split the string into a list of lines.

```python
str1="Hello\nWorld\nPython"
print(str1.split("\n"))
```

```
['Hello', 'World', 'Python']
```

26. You have a string containing a sentence. Use a string method to split the sentence into individual words and then join them back together with spaces.

```python
str1="Hello World Python"
print(" ".join(str1.split()))
```

```
Hello World Python
```

27. Given a string containing a file path, use a string method to extract the file name.

```python
str1="/window/users/hello.js"
print(str1.split("/")[-1])
```

```
hello.js
```

30. You have a string containing a sentence with multiple spaces between words. Use a string method to remove all extra spaces, leaving only a single space between words.

```python


def remove_extra_spaces(sentence):

  words = sentence.split()


  return " ".join(words)


# Example usage:
```

```
sentence = "This  is  a    sentence    with    multiple    spaces."
print(remove_extra_spaces(sentence))

This is a sentence with multiple spaces.
```

Operators-->>Practice questions 1. You have two variables, a and b, containing integer values. Use arithmetic operators to calculate and print the sum, difference, product, and quotient of these variables.

```
a=10
b=20
print(a+b)
print(a/b)
print(a-b)
print(a%b)
print(a*b)

30
0.5
-10
10
200
```

2. Given a variable x containing an integer value, use the modulus operator to check if x is even or odd.

```
x=10
if x%2==0:
  print("Even")
else:
  print("Odd")

Even
```

3. You have two variables, a and b. Use comparison operators to check if a is greater than b, and print the result.

```
a=10
b=20
if a>b:
  print("a is greater than b")
else:
  print("b is greater than a")

b is greater than a
```

4. Given two boolean variables, p and q, use logical operators to evaluate and print the result of p AND q, p OR q, and NOT p.

```
p=10
q=20
print(p and q)
print(p or q)

20
10
```

5. You have a variable n containing an integer value. Use the bitwise AND, OR, and XOR operators to perform operations with another integer m.

```
n=10
m=20
print(n&m)
print(n|m)
print(n^m)

0
30
30
```

6. Given a variable y containing a floating-point number, use the floor division operator to divide y by 2 and print the result.

```
y=19.98
print(y//2)

9.0
```

7. You have two strings, str1 and str2. Use the concatenation operator to join these strings and print the result.

```
str1="Mehak"
str2="Jacky"
print(str1+" "+str2)

Mehak Jacky
```

8. Given a variable z containing an integer, use the increment operator to increase its value by 1 and print the result.

```
x=10
x=x+1
print(x)

11
```

9. You have a list my_list and a value val. Use the in operator to check if val is present in my_list and print the result.

```
my_list={"mehak":123,"jacky":234,"bob":456,"oggy":789}
value=input("Enter the value: ")
if value in my_list:
  print("present")
else:
  print("Not present")

Enter the value: mehak
present
```

10. Given two variables a and b, use the assignment operator to assign the value of b to a and print a.

```
a=10
b=20
a=b
print(a)

20
```

11. You have two variables, x and y, containing integer values. Use the compound assignment operator to add y to x and print the result.

```
x=10
y=20
x+=y
print(x)

30
```

12. Given a variable num containing an integer value, use the bitwise left shift operator to shift num by 2 bits and print the result.

```

num = 1010

print(num<<2)

4040
```

13. You have a string text and an integer n. Use the repetition operator to repeat text n times and print the result.

```

text = "Hello"
n = 3

repeated_text = text * n
```

```
print(repeated_text)

HelloHelloHello
```

14. Given two variables, a and b, use the comparison operators to check if a is equal to b and if a is not equal to b. Print the results.

```
a=10
b=20
print(a==b)
print(a!=b)

False
True
```

15. You have two variables, x and y, containing integer values. Use the bitwise right shift operator to shift x by 3 bits and print the result.

```
x=10
y=20
print(x>>3)

1
```

22. Given a variable num containing an integer value, use the bitwise NOT operator to invert its bits and print the result.

```
x=10
print(~x)

-11
```

24. Given two boolean variables, a and b, use the logical operator to evaluate a AND NOT b and print the result.

```
a = True
b = False

result = a and not b

print(result)

True
```

27. You have a string str1 and an integer n. Use the slicing operator to get the first n characters of the string and print the result.

```
str1="Hello World"
n=2
print(str1[:n])
```

```
He
```

30. Given a variable x containing an integer, use the decrement operator to decrease its value by 1 and print the result.

```
a=10
a-=1
print(a)
```

```
9
```