

Karthik Nair | 3EA | 00229802021

# Strings

---

1. Write a Python program to get a string made of the first 2 and the last 2 chars from a given string.

If the string length is less than 2, return "Empty String".

Sample String: 'Hello Python'

Expected Result : 'Heon'

Sample String : 'Hp'

Expected Result : 'HpHp'

Sample String : 'H'

Expected Result: Empty String

```
def string_first_and_last_two_chars(s):  
    return s[:2]+s[-2:] if len(s)>=2 else "Empty string"  
  
test_cases=["Hello Python","Hp","H"]  
for i in test_cases:  
    print(string_first_and_last_two_chars(i))
```

2. Write a program in Python to count lower, upper, numeric and special characters in a string."

def count\_chars(s):

return {"lower":len([i for i in s if i.islower()]),"upper":len([i for i in s if i.isupper()]),"numeric":len([i for i in s if i.isnumeric()]),"special":len([i for i in s if not i.isalnum()])}

for x,y in count\_chars("Karthik@cosmic42").items():

print(f"Number of {x} characters: {y}")

3. Write a program to create a string made of the first, middle and last character. If the number of characters in the string is even, consider first letter out of the two middle characters as middle one.

Sample String: 'Plane'

Expected Result: 'Pae'

Sample String: 'Planer'

Expected Result: 'Par'

```
def get_middle_char(s):  
    if len(s)%2==0:  
        return s[0]+s[len(s)//2-1]+s[-1]  
    else:  
        return s[0]+s[len(s)//2]+s[-1]  
  
print(get_middle_char("Plane"))  
print(get_middle_char("Planer"))
```

4. Given two strings, s1 and s2. Write a program to create a new string s3 made of the first char of s1, then the last char of s2, Next, the second char of s1 and second last char of s2, and so on. Any leftover chars go at the end of the result.

Sample String: 'Good', 'Morning'

Expected Result: 'GgonoidnMor'

```
s1="Good"  
s2="Morning"  
s3=""  
  
e=-1  
for i in range(len(s1)):  
    s3+=s1[i]+s2[e]  
    e-=1  
  
if len(s2)>len(s1):  
    s3+=s2[0:e+1]  
  
print(s3)
```

5. Given two strings, s1 and s2. Write a program to create a new string s3 made of the first char of s1, then the last char of s2, Next, the second char of s1 and second last char of s2, and so on. Any leftover chars go at the end of the result.

Sample String: 'Good', 'Morning'

Expected Result: 'GgonoidnMor'

```
s1="Good"
s2="Morning"
s3=""

for i in range(len(s1)):
    s3+=s1[i]+s2[0-1-i]

if len(s2)>len(s1):
    s3+=s2[0:0-1-i]

print(s3)
```

6. Write a program to check if two strings are balanced. For example, strings s1 and s2 are balanced if all the characters in the s1 are present in s2. The character's position doesn't matter.

```
s1="Good"
s2="Morning"

print("balanced" if set(s1).issubset(set(s2)) else "not balanced")

s1="Good"
s2="God"

print("balanced" if set(s1).issubset(set(s2)) else "not balanced")
```

7. Write a program to calculate the sum and average of the digits present in a string.

```
def sum_avg(s):
    sum=0
    d=0
    for i in s:
        if i.isdigit():
            d+=1
            sum+=int(i)
    return sum, sum/d

print(sum_avg("abc123"))
```

8. Write a program to reverse a string if it's length is a multiple of 4.

```
def reverse(s, n):  
    if len(s) % n != 0:  
        return  
    return s[::-1]  
  
print(reverse('abcd', 4))
```

9. Write a Python program to reverse words in a string.

Sample String: 'Here is a school'

Expected Result: 'school a is Here'

```
def reverse_words(s):  
    return ' '.join(s.split()[::-1])  
  
print(reverse_words('Here is a school'))
```

10. Write a Python program to move spaces to the front of a given string.

```
def move_spaces(s):  
    return s.count(" ")*" " +"".join(s.split())  
  
print(move_spaces('Here is a school'))
```

11. Write a Python program to capitalize first and last letters of each word of a given string.

```
def capitalize(s):  
    return ' '.join([w[0].upper() + w[1:-1] + w[-1].upper() for w in s.split()])  
  
print(capitalize('Here is a school'))
```

12. Write a Python program to replace each character of a word of length five and more with hash character (#).

```
data='''What's the difference between a dead dog in the road and a dead  
lawyer in the road?'''
```

```
A: There are skid marks in front of the dog. '''

# l=data.split()
# for i in l:
#     if len(i)>=5:
#         print( i.replace(i, "#"*len(i)), end=" ")
#     else:
#         print(i, end=" ")

print(" ".join([i.replace(i, "#"*len(i)) if len(i)>=5 else i for i in data.split()
]))

# data=" ".join([i.replace(i, "#"*len(i)) for i in data.split() if len(i)>5 ])
# print(data)
```

13. Write a Python program to get a string from a given string where all occurrences of its first char have been changed to '#', except the first char itself.

Sample String : 'extent'

Expected Result : 'ext#nt'

```
s=input("Enter a string: ")
print(s[0]+s[1:].replace(s[0], '#'))
```

14. Write a Python program that returns a string sorted alphabetically by the first character of a given string of words.

```
# string0="Roses apple Banana penguin"

# print(sorted([string0.split()[i] for i in range(len(string0.split()))]))

string0="Roses apple Banana penguin"
print(" ".join(sorted(string0.split())))
```

15. Write a Python program to get a single string from two given strings, separated by a space and swap the first two characters of each string.

Sample String : 'good', 'evening'

Expected Result : 'evod goening'

```
a= input("Enter string: ")
b= input("Enter string: ")
print(b[:2]+a[2:]+ " "+a[:2]+b[2:])
```

16. Write a Python program to add 'ing' at the end of a given string (length should be at least 3). If the given string already ends with 'ing' then add 'ly' instead. If the string length of the given string is less than 3, leave it unchanged.

```
# Sample String : 'practice'
# Expected Result : 'practiceing'
# Sample String : 'eating'
# Expected Result : 'eatingly'

s=input('Enter a string: ')
if len(s)<3:
    print('Input string is too short')
    print(s)
else:
    if s[-3:]=='ing':
        print(f"{s}" 'ly')
    else:
        print(s+'ing')
```

17. Write a program to count vowels and consonants in a string.

```
s=input("Enter a string: ")
vc=0
cc=0
for i in s:
    if i in ['a','e','i','o','u']:
        vc+=1
    else:
        cc+=1
print("No of vowels: ",vc)
print("No of consonants: ",cc)
```

18. remove duplicates in a string

```
a= input ("Enter a string: ")
b= ""
for i in a:
    if i not in b:
        b+=i

print (b)
```

19. Write a program to count the occurrence of each character in a word.

```
def count_char_occurence(s):
    return {i:s.count(i) for i in s}

for x,y in count_char_occurence("Hogwarts is Home").items():
    print(f"{x} Appears {y} times")
```

20. Write a program to convert lower letter to upper and upper letter to lower in a string.

Sample String: 'Hello Python'

Expected Result: 'hELLO pYTHON'

```
def convert_case(s):
    return s.swapcase()

print(convert_case("Hello Python"))
```

21. Write a program to sort letters of word by lower to upper case format.

Sample String: 'I am gOIIng HoMe'

Expected Result: 'amngngoe IOIHM'

```
def sort_basis_of_case(s):
    return str([i for i in s if i.islower()]+[i for i in s if not i.isupper()])

sort_basis_of_case("I am gOIIng HoMe")
```