find and replace methods

in operator	This operator search a given string within string and returns True if string found else False
find()	This method returns index of search string (left to right)
replace()	This method replace string within string and return replaced string as new string
rfind()	This method returns index of search string (right to left)

Example:

```
doj="date of joining 14/6/2025"
b="14/6/2025" in doj
print(b)
b="13/6/2025" in doj
print(b)
names=["naresh","ramesh","kishore","rajesh"]
b="naresh" in names
print(b)
email_list=["naresh@gmail.com",
        "suresh@nareshit.com",
        "rajesh@gmail.com"]
b="suresh@nareshit.com" in email_list
print(b)
```

Output

```
True
False
True
True
```

Example:

```
>>> str1="python easy programming language"
>>> index=str1.find("easy")
>>> print(index)
7
>>> str2="python java c++ java oracle"
>>> index=str2.find("java")
>>> print(index)
7
```

```
>>> index=str2.rfind("java")
>>> print(index)
16
>>> index=str1.find("mysql")
>>> print(index)
-1
```

Example:

```
# Write a program to count a given string existing
# how many times within string

str1="java python java oracle java .net java mysql"
str2="java"

i=0
c=0
while True:
    index=str1.find(str2,i)
    if index!=-1:
        c+=1
    else:
        break
    i=index+1

print(str1)
print(c)
```

Output

```
java python java oracle java .net java mysql
4
```

Using predefined methods

```
# Write a program to count a given string existing
# how many times within string

str1="java python java oracle java .net java mysql"
str2="java"
```

```
c=str1.count("java")
print(c)
```

count() method can be applied to any sequence data type

Example:

```
# Write a program to print the following output
# str1="abcaaabcd"
# output:4a2b2c1d

str1="abcaaabcd"
str2=""
for ch in str1:
    if ch not in str2:
        str2=str2+ch

print(str1)
print(str2)
str3=""
for ch in str2:
    c=str1.count(ch)
    str3=str3+str(c)+ch
print(str3)
```

Output

```
abcaaabcd
abcd
4a2b2c1d
```

Example:

```
# Write a program to print the following output
# str1="4a2b2c1d"
# output:aaaabbccd

str1="4a2b2c1d"
str2=""
for i in range(0,len(str1),2):
```

```
    n=int(str1[i])
    ch=str1[i+1]
    str2=str2+(n*ch)

print(str1)
print(str2)
```

Output

```
4a2b2c1d
aaaabbccd
```

Example of replace

```
>>> str1="python java oracle java .net"
>>> str2=str1.replace("java","jython")
>>> print(str1)
python java oracle java .net
>>> print(str2)
python jython oracle jython .net
>>> str3=str1.replace("java","jython",1)
>>> print(str3)
```

Partition methods

partition()	This method split string into 3 values (tuple) It search for separator from left to right
rpartition()	This method split string into 3 values (tuple) It search for separator from right to left

```
>>> str1="a,b,c,d,e"
>>> t1=str1.partition(",")
>>> print(t1)
('a', ',', 'b,c,d,e')
>>> t2=str1.rpartition(",")
>>> print(t2)
('a,b,c,d', ',', 'e')
```

Sets