

```
s='adas'  
type(s)
```

```
str
```

forward indexing

```
s[3]  
  
's'
```

backward indexing

```
s[-1]  
  
's'
```

String index out of Range

```
s[100]  
  
-----  
IndexError                                Traceback (most recent call last)  
<ipython-input-9-2a138df92e52> in <module>  
----> 1 s[100]  
  
IndexError: string index out of range
```

SEARCH STACK OVERFLOW

it starts from 0 and end to 9

```
a='my name is das'  
a[0:10]  
  
'my name is'
```

```
b="ineuron"  
b[0:3]  
  
'ine'
```

```
b[0:300]  
  
'ineuron'
```

```
b[300]  
  
-----  
IndexError                                Traceback (most recent call last)  
<ipython-input-58-6ff6ddb2b1a6> in <module>  
----> 1 b[300]  
  
IndexError: string index out of range
```

SEARCH STACK OVERFLOW

```
b[-1]  
  
'n'
```

```
b[-100]  
  
-----  
IndexError                                Traceback (most recent call last)  
<ipython-input-57-b20bba380709> in <module>  
----> 1 b[-100]  
  
IndexError: string index out of range
```

SEARCH STACK OVERFLOW

```
b
    'ineuron'

b[-1:-4]
    ''

b[-1:-4:-1]
    'nor'

b[0:4]
    'ineu'

a="kumar"
a[0:3]
    'kum'

a[0:300]
    'kumar'

a[0:300:2]
    'kmr'

a[0:300:3]
    'ka'

a[0:100:-1]
    ''

a[-1:-4]
    ''

a
    'kumar'

a[-1:-4:-1]
    'ram'

a[0:-10:-1]
    'k'

a[:, :]
    'kumar'

a[:8]
    'kumar'

a[-2:]
    'ar'

a
    'kumar'

a[-2:-1:1]
```

```
'a'

a[-2]

'a'

a[::-1]

'ramuk'

a[-1::-1]

'ramuk'

a='i am working with ineuron'
a[::-1]

'norueni htiw gnikrow ma i'

a

'i am working with ineuron'

a[-5:5:-1]

'eni htiw gnikro'

a[-2:-10:-1]

'orueni h'

a

'i am working with ineuron'

a[0:100:3]

'imoi tiun'
```

Repeat same string

```
"mdas"*3

'mdasmdasmdas'
```

concatenation string

```
"m"+"das"

'mdas'
```

```
a

'i am working with ineuron'
```

```
len(a)

25
```

```
"a"*"a"
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-50-58dabfd08b64> in <module>
----> 1 "a"*"a"

TypeError: can't multiply sequence by non-int of type 'str'
```

SEARCH STACK OVERFLOW

```
a

'i am working with ineuron'

#find particular string that is a and return value indexwise
a.find('a')

2

#i has many places but by default takes first position
a.find('i')

0

a.find('ia')

-1

a

'i am working with ineuron'

a.find('in')

9

#how manytimes a has appeared
a.count('i')

4

a

'i am working with ineuron'

a.count('x')

0

string to list

l=a.split()

type(l)

list

l

['i', 'am', 'working', 'with', 'ineuron']

l[0]

'i'

l[1]

'am'

l[3]

'with'

l[0:3]

['i', 'am', 'working']

a.split('w')

['i am ', 'orking ', 'ith ineuron']
```

```
#wo is splitting delimiter
a.split('wo')

['i am ', 'rking with ineuron']

a.upper()

'I AM WORKING WITH INEURON'

#using swap function we can convert upper to lower and lower to upper
s='adas'

s.swapcase()

'aDAS'

s='my faviourite mentor is Sudhanshu kumar'

#first character of word is upper cases and rest of the cases are lower cases
s.title()

'My Faviourite Mentor Is Sudhanshu Kumar'

s='adas'

#capitalize first letter
s.capitalize()

'Adas'

b='sudh'
c='ineuron'
b.join(c)

'isudhnsudhesudhusudhrsudhosudhn'

" ".join("sudh")

's u d h'

for i in reversed("sudh"):
    print(i)

h
d
u
s

s='sudh'
s[::-1]

'hdus'

s='sudh'
s.rstrip()

'sudh'

s.lstrip()

'sudh'

s.strip()

'sudh'

s.replace('u','i')

'sidh'

"sudh\tkumar".expandtabs()
```

```
'sudh    kumar '  
  
s='sudh'  
s.center(40,'#')  
  
'#####sudh#####'  
  
s  
  
'sudh'  
  
s.isupper()  
  
False  
  
s='ADAS'  
s.isupper()  
  
True  
  
s.islower()  
  
False  
  
s="  "  
s.isspace()  
  
True  
  
s="90000"  
s.isdigit()  
  
True  
  
s='sudhanshu sir'  
  
s.endswith('r')  
  
True  
  
s.istitle()  
  
False  
  
l=['anwasha','das',9,9+7j,True,3.98]  
type(l)  
  
list  
  
l[0]  
  
'a'  
  
l[-1]  
  
3.98  
  
l[0:4]  
  
['a', 'das', 9, (9+7j)]  
  
l  
  
['a', 'das', 9, (9+7j), True, 3.98]  
  
l[:-1]  
  
['a', 'das', 9, (9+7j), True]  
  
l
```

```
[ 'a', 'das', 9, (9+7j), True, 3.98]

l[-5]

'das'

l[0:4]

[ 'a', 'das', 9, (9+7j)]

l

[ 'a', 'das', 9, (9+7j), True, 3.98]

l[::-1]

[3.98, True, (9+7j), 9, 'das', 'a']

l[-1:6]

[3.98]

type(l[4])

bool

l[0]

'anwasha'

l

[ 'anwasha', 'das', 9, (9+7j), True, 3.98]

l[0][1]

'n'

l[3]

(9+7j)

l[3].real

9.0

l[3].imag

7.0

l1=["anwasha","das",3456]
l2=["xyz","pqr",90.0]
l1+l2

[ 'anwasha', 'das', 3456, 'xyz', 'pqr', 90.0]

l1+[ 'sudh' ]

[ 'anwasha', 'das', 3456, 'sudh' ]

l1*4

[ 'anwasha',
 'das',
 3456,
 'anwasha',
 'das',
 3456,
 'anwasha',
 'das',
 3456,
 'anwasha',
 'das',
 3456]
```

```
l1*2

['anwesha', 'das', 3456, 'anwesha', 'das', 3456]
```

```
l1

['anwesha', 'das', 3456]
```

```
l1[0]

'anwesha'
```

```
l1[0]='yourname'
l1

['yourname', 'das', 3456]
```

```
l1[0]=456899
l1

[456899, 'das', 3456]
```

```
l1[1]

'das'
```

String is immutable

```
l1[1][0]='m'
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-35-9040588ffdc6> in <module>
----> 1 l1[1][0]='m'

TypeError: 'str' object does not support item assignment
```

SEARCH STACK OVERFLOW

```
l1

[456899, 'das', 3456]
```

```
s='sudh'
```

address has changed using replace operation. it does not mean that string is mutable

```
s.replace('s','v')

'vudh'
```

```
l1
len(l1)

3
```

```
'kumar' in l1

False
```

```
345 in l1

False
```

```
l = ['okay','kumar', 78,45,789,5.5,True, 5+8j]
l[-1:6:-1]

[(5+8j)]
```



```
l2.append('sudh')  
l2
```

```
['xyz', 'pqr', 90.0, 'sudh', 'sudh']
```

```
l2.pop()
```

```
'sudh'
```

```
l2
```

```
['xyz', 'pqr', 90.0, 'sudh']
```

```
l2.pop(4)
```

```
-----  
IndexError                                Traceback (most recent call last)  
<ipython-input-11-f3647d784fc4> in <module>  
----> 1 l2.pop(4)
```

```
IndexError: pop index out of range
```

SEARCH STACK OVERFLOW

```
l2.pop(2)
```

```
90.0
```

```
l2
```

```
['xyz', 'pqr', 'sudh']
```

```
l2.insert(1,"yuijj")
```

```
l2
```

```
['xyz', 'yuijj', 'pqr', 'sudh', 9, 9]
```

```
l2
```

```
['xyz', 'pqr', 90.0]
```

```
l2.insert(3,[67,90,78,56])
```

```
l2
```

```
['xyz', 'yuijj', 'pqr', [67, 90, 78, 56], 'sudh', 9, 9]
```

```
l2.append(9)
```

```
l2
```

```
['xyz', 'pqr', 'sudh', 9, 9]
```

```
l2
```

```
[90.0, 'pqr', 'xyz']
```

```
l2[::-1]
```

```
[90.0, 'pqr', 'xyz']
```

```
l2.reverse()
```

```
l2
```

```
['xyz', 'pqr', 90.0]
```

```
l2[1]
```

```
'pqr'
```

```
l2.count('xyz')
```

```
1
```

```
l2.append('vhjgvgvvggg')
```

```
l2
```

```
['xyz', 'pqr', 90.0, 'vhjgvgvvggg']
```

```
l2.append([7,8,9,10])
```

```
l2
```

```
['xyz', 'pqr', 90.0, 'vhjgvgvvggg', [7, 8, 9], [7, 8, 9], [7, 8, 9, 10]]
```

```
l1.extend([8,9,6])
```

```
l1
```

```
['anwesha', 'das', 3456, 8, 9, 6, 8, 9, 6]
```

one string related problem

```
s = input()
res = []
for i in range(len(s)):
    if not res or s[i] != res[-1]:
        res=res+ [s[i]]
    else:
        res.pop()
if res:
    print (''.join(res))
else:
    print ('Empty String')

baab
Empty String
```

✓ 0s completed at 12:43 AM

