

String Methods #2

`S.replace(old, new,[,count])`

`S.join(iterable)`

`S.split([sep[, max split]])`

`S.rsplit([sep[, max split]])`

`s.splitlines([keepends])`

`s='a-b-c-d-e'` want to replace all these hyphens with commas using these methods

`S.replace('-', ', 3)`

So the first one is old and the next one is new

`'a, b, c, d, e' ->` This is a new string modify the old string and generate new string

We can mention the count but it is optional

```
s='a-b-c-d-e'
s
'a-b-c-d-e'
s.replace('-', ',')#replacing it with comma
'a,b,c,d,e'
s
'a-b-c-d-e'
s.replace('k','m')
'a-b-c-d-e'
s='abcd@gmail.com'
s.replace('gmail','yahoo')#replacing it with yahoo
'abcd@yahoo.com'
```

This is the replace method

`s.join(iterable)`

The join method:

```
s1='xyz'
s2='abc'
s1.join(s2)# calling s1 method and joining upon s2
'axyzbxyz'
s1='/'
s1.join(s2)# letters of s1 and s2 as a separator
'a/b/c'
```

So, the parameter is string so its taking letters of a string

The split method:

```
s='john smith ajay'
s.split()# it splits
```

```
['john', 'smith', 'ajay']  
s.split('h')  
['jo', 'n smit', ' ajay']  
s.split(',')  
['john smith ajay']  
s='john-smith-ajay-khan-james'  
s.split('-')  
['john', 'smith', 'ajay', 'khan', 'james']
```

The s. rSplit method:

The s. Split means splitting is done from right hand side

S.splitlines works same as split only

```
s='aaa\n' bbb\t ccc\t
```