

String Method #2

`s.ljust(width[,fill])`
`s.rjust(width[,fill])`
`s.center(width[,fill])`

- These methods are useful for text alignment

`s = 'python '`

- If you want to write in extra spaces like 10 spaces

`s.ljust ()`

`P y t h o n _ _ _ _`

`s.rjust ()`

`_ _ _ _ P y t h o n`

`s.center ()`

`_ _ P y t h o n _ _`

- If the string is larger than the spaces given by using

`s.ljust(3)`

- It will take the entire string it will not just take 3 letters
- If you want bigger space you mention the width bigger than the length of the string
- All these have one more parameter that is fill

`s.center(10 , ' * ')`

- Python has only 6 alphabets but we want 10 spaces . * will be filled in empty spaces
- 10 space vacant spaces with * otherwise it will fill with spaces

`s = python`

- String is immutable so it will not modify it will create a new string

`s.strip ([chars])`
`s.lstrip ([chars])`
`s.rstrip ([chars])`



this is useful for removing the characters from the string

- They remove leading char , trailing characters and characters from both sides ... by default they will remove spaces

`s.lstrip` - it will remove leading char

`s.rstrip` - trailing removes character

`s.strip` - removes spaces from both the side

S = ' ++aaapython '

s.lstrip(' . ') — — —> it will remove leading dots and stops when there is no dot

O/p - '++aaapython '

s.lstrip(' . + ') — — —> it will remove dot spaces and +

O/p - 'aaapython'

- All this methods will return new string they will generate new string after performing the operations