

```
In [1]: print('Hello there\n Im doing well\n how are you')
```

```
Hello there
 Im doing well
 how are you
```

```
In [3]: print('Hello there\nIm doing well\nhow are you')
```

```
Hello there
Im doing well
how are you
```

raw string

```
In [6]: print(r'Hello guys what a surprise after long time')
```

```
Hello guys what a surprise after long time
```

Multiline strings

```
In [9]: print('''Dear friend
I'm fine here and doing well
How are you there.
I Hope u r also doing well''')
```

```
Dear friend
I'm fine here and doing well
How are you there.
I Hope u r also doing well
```

```
In [31]: print(
...     """Dear Friend,
...
... I'm fine here and doing well.
... How are you there.
... I Hope u r also doing well"""
... )
```

```
Cell In[31], line 2
... """Dear Friend,
^
```

SyntaxError: invalid syntax. Perhaps you forgot a comma?

```
In [27]: print(
...     """Dear Alice,
...
... Eve's cat has been arrested for catnapping,
... cat burglary, and extortion.
...
... Sincerely,
... Bob"""
... )
```

Dear Alice,

Eve's cat has been arrested for catnapping,
cat burglary, and extortion.

Sincerely,
Bob

INDEXING AND SLICING

```
In [34]: b='Hello World!'
b[0]
```

```
Out[34]: 'H'
```

```
In [36]: b[-1]
```

```
Out[36]: 'l'
```

```
In [38]: b[5]
```

```
Out[38]: ' '
```

```
In [40]: b[0:9]
```

```
Out[40]: 'Hello Wor'
```

```
In [42]: b[0:17]
```

```
Out[42]: 'Hello World!'
```

```
In [44]: b[:]
```

```
Out[44]: 'Hello World!'
```

```
In [46]: b[::-1]
```

```
Out[46]: '!dlrow olleH'
```

```
In [48]: b[0:5]
```

```
Out[48]: 'Hello'
```

```
In [52]: 'Hi'+b[0:5]
```

```
Out[52]: 'HiHello'
```

```
In [54]: b.replace('hello','hi')
```

```
Out[54]: 'Hello World!'
```

```
In [56]: b
Out[56]: 'Hello World!'

In [58]: b.replace('Hello', 'Hi')
Out[58]: 'Hi World!'

In [62]: b.replace('World', 'Madam')
Out[62]: 'Hello Madam!'
```

IN AND NOT IN OPERATORS

```
In [65]: b
Out[65]: 'Hello World!'

In [71]: 'Hello' in b
Out[71]: True

In [73]: 'world' not in b
Out[73]: True

In [75]: 'World' not in b
Out[75]: False

In [77]: '' in b
Out[77]: True

In [79]: 'Madam' in b
Out[79]: False
```

UPPER LOWER TITLE

```
In [91]: a='Hello Madam'
a
Out[91]: 'Hello Madam'

In [83]: a.upper()
Out[83]: 'HELLO MADAM'
```

```
In [85]: a.lower()
```

```
Out[85]: 'hello madam'
```

```
In [94]: a.title()
```

```
Out[94]: 'Hello Madam'
```

isupper islower methods

```
In [97]: a
```

```
Out[97]: 'Hello Madam'
```

```
In [109... b='HELLO MADAM'  
b
```

```
Out[109... 'HELLO MADAM'
```

```
In [103... c='hello sir'  
c
```

```
Out[103... 'hello sir'
```

```
In [111... a.isupper()
```

```
Out[111... False
```

```
In [113... a.islower()
```

```
Out[113... False
```

```
In [115... a.istitle()
```

```
Out[115... True
```

```
In [117... b.isupper()
```

```
Out[117... True
```

```
In [119... b.islower()
```

```
Out[119... False
```

```
In [121... b.istitle()
```

```
Out[121... False
```

```
In [123... c.isupper()
```

Out[123... False

In [125... `c.islower()`

Out[125... True

startswith and endswith

In [128... `b.startswith('hello')`

Out[128... False

In [130... `b.endswith('MADAM')`

Out[130... True

In [132... `a.startswith('hello')`

Out[132... False

In [134... `a.startswith('Hello')`

Out[134... True

In [136... `a.endswith('Madam')`

Out[136... True

In [138... `p='abc123'`
`p`

Out[138... 'abc123'

In [140... `p.startswith('abc')`

Out[140... True

In [142... `p.endswith('23')`

Out[142... True

join and split

In [147... `' '.join(['My', 'Name', 'is', 'Vaishnavi'])`

Out[147... 'My Name is Vaishnavi'

In [155... `' , '.join(['apple', 'banana', 'carrot'])`

```
Out[155... 'apple , banana , carrot'
```

```
In [157... ' / '.join(['Maybe','Maynotbe','unfortunately'])
```

```
Out[157... 'Maybe / Maynotbe / unfortunately'
```

```
In [169... 'ABC'.join('Currently')
```

```
Out[169... 'CABCuABCrABCrABCeABCnABCTABClABCy'
```

```
In [171... 'ABC'.join(['My' , 'Name', 'is', 'Vaishnavi'])
```

```
Out[171... 'MyABCNameABCisABCVaishnavi'
```

```
In [173... 'currently I am looking for job'.split()
```

```
Out[173... ['currently', 'I', 'am', 'looking', 'for', 'job']
```

```
In [181... 'I eat apple everyday'.split()
```

```
Out[181... ['I', 'eat', 'apple', 'everyday']
```

```
In [183... 'I @eat@ apple@ @everyday@'.split('@')
```

```
Out[183... ['I ', 'eat', ' apple', ' ', 'everyday', '']
```

```
In [185... 'I gave my presentation'.split('e')
```

```
Out[185... ['I gav', ' my pr', 's', 'ntation']
```

```
In [191... 'I had my breakfast'.split(' ')
```

```
Out[191... ['I', '', 'had', '', 'my', '', 'breakfast']
```

rjust() ljust() center()

```
In [196... 'Goodmrng'.rjust(10)
```

```
Out[196... ' Goodmrng'
```

```
In [198... 'Goodmrng'.rjust(20)
```

```
Out[198... ' Goodmrng'
```

```
In [200... 'Hello'.ljust(15)
```

```
Out[200... 'Hello '
```

```
In [202... 'hello'.ljust(20)
```

Out[202...] `'hello'`

In [208...] `'world'.center(10)`

Out[208...] `' world '`

In [214...] `'Hello'.rjust(10, '@')`

Out[214...] `'@@@@@Hello'`

In [216...] `'World'.ljust(20, '=')`

Out[216...] `'World====='`

In [220...] `'Goodmorning'.center(30, '_')`

Out[220...] `'_____Goodmorning_____'`

Removing whitespace with strip(), rstrip(), and lstrip()

In [227...] `b='Hello ' 'world'`
`b.strip()`

Out[227...] `'Hello world'`

In [229...] `b.rstrip()`

Out[229...] `'Hello world'`

In [231...] `d=' hello planets '`
`d`

Out[231...] `' hello planets '`

In [233...] `d.strip()`

Out[233...] `'hello planets'`

In [235...] `d.rstrip()`

Out[235...] `' hello planets'`

In [237...] `d.lstrip()`

Out[237...] `'hello planets '`

In [239...] `q='abcabcdabcdefabcdefg'`
`q`

Out[239... 'abcabcdabcdefabcdefg'

In [241... `q.strip('abc')`

Out[241... 'dabcdefabcdefg'

In [245... `r='abcabcabcabcabcdeabcde'`
`r.strip('abc')`

Out[245... 'dabcdeabcde'

In [253... `r='abcabcabc'`
`r.strip('ab')`

Out[253... 'cabcab'

In [255... `r.rstrip('abc')`

Out[255... ''

In [261... `q.rstrip('efg')`

Out[261... 'abcabcdabcdefabcd'

In [259... `q.lstrip('abc')`

Out[259... 'dabcdefabcdefg'

count method

In [264... `s='I tried my level best'`
`s.count('e')`

Out[264... 4

In [266... `s.count('tried')`

Out[266... 1

In [268... `s.count('tired')`

Out[268... 0

In [277... `s.count('e',2)`

Out[277... 4

In [281... `s.count('e',8)`

Out[281... 3


```
In [283... s.count('e',12)
```

```
Out[283... 3
```

```
In [285... s.count('e',16)
```

```
Out[285... 1
```

replace method

```
In [293... z='mangoes,banana,apples,oranges,mangoes '  
z
```

```
Out[293... 'mangoes,banana,apples,oranges,mangoes '
```

```
In [295... z.replace('mangoes','grapes')
```

```
Out[295... 'grapes,banana,apples,oranges,grapes '
```

```
In [297... z.replace('mangoes','grapes',1)
```

```
Out[297... 'grapes,banana,apples,oranges,mangoes '
```

```
In [299... z.replace('mangoes','grapes',5)
```

```
Out[299... 'grapes,banana,apples,oranges,grapes '
```

```
In [301... z
```

```
Out[301... 'mangoes,banana,apples,oranges,mangoes '
```

```
In [303... z.replace('mangoes','grapes',5)
```

```
Out[303... 'grapes,banana,apples,oranges,grapes '
```

```
In [ ]:
```