

Manipulating Strings

- escape characters

```
In [3]: '\'
```

```
Out[3]: ''
```

```
In [4]: '\"'
```

```
Out[4]: ''
```

```
In [5]: '\t'
```

```
Out[5]: '\t'
```

```
In [6]: '\n'
```

```
Out[6]: '\n'
```

```
In [7]: '\\'
```

```
Out[7]: '\\'
```

```
In [8]: '\b'
```

```
Out[8]: '\x08'
```

```
In [11]: '\000'
```

```
Out[11]: '\x00'
```

```
In [12]: '\r'
```

```
Out[12]: '\r'
```

```
In [15]: print("Hello there!\nHow are you?\nI'm doing fine.")
```

```
Hello there!  
How are you?  
I'm doing fine.
```

Raw strings

- A raw string entirely ignores all escape characters and prints any backslash that appears

in the string.

```
In [16]: print(r"Hello there!\nHow are you?\nI'm doing fine.")
```

Hello there!\nHow are you?\nI\'m doing fine.

Multiline Strings

```
In [18]: print(  
...     """Dear Anilkumar,  
...  
...     Eve's cat has been arrested for catnapping,  
...     cat burglary, and extortion.  
...  
...     Sincerely,  
...     Saikumar"""  
... )
```

Dear Anilkumar,

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

Sincerely,
Saikumar

Indexing and Slicing strings

```
In [20]: spam = 'Hello world!'
```

```
In [21]: spam[0]
```

```
Out[21]: 'H'
```

```
In [22]: spam[4]
```

```
Out[22]: 'o'
```

```
In [23]: spam[-1]
```

```
Out[23]: '!'
```

Slicing

```
In [24]: spam = 'Hello world!'
```

```
In [25]: spam[0:5]
```

```
Out[25]: 'Hello'
```

```
In [26]: spam[:5]
```

```
Out[26]: 'Hello'
```

```
In [27]: spam[6:-1]
```

Out[27]: 'world'

In [28]: `spam[:-1]`

Out[28]: 'Hello world'

In [29]: `spam[::-1]`

Out[29]: '!dlrow olleH'

In [30]: `fizz = spam[0:5]`
`fizz`

Out[30]: 'Hello'

In []: *# The in and not in operators*

In [31]: `'Hello' in 'Hello World'`

Out[31]: True

In [33]: `'Hello' in 'Hello'`

Out[33]: True

In [34]: `'HELLO' in 'Hello World'`

Out[34]: False

In [35]: `'' in 'spam'`

Out[35]: True

In [36]: `'cats' not in 'cats and dogs'`

Out[36]: False

upper(),lower() and title()

In [37]: `greet = 'Hello world!'`
`greet.upper()`

Out[37]: 'HELLO WORLD!'

In [38]: `greet.lower()`

Out[38]: 'hello world!'

In [39]: `greet.title()`

Out[39]: 'Hello World!'

isupper() and islower() methods

```
In [41]: spam = 'Hello world!'
spam.islower()
```

Out[41]: False

```
In [44]: spam.isupper()
```

Out[44]: False

```
In [45]: 'HELLO'.isupper()
```

Out[45]: True

```
In [46]: 'abc12345'.islower()
```

Out[46]: True

```
In [47]: '12345'.islower()
```

Out[47]: False

```
In [48]: '12345'.isupper()
```

Out[48]: False

startswith() and endswith()

```
In [50]: 'Hello world!'.startswith('Hello')
```

Out[50]: True

```
In [51]: 'Hello world!'.endswith('world!')
```

Out[51]: True

```
In [52]: 'abc123'.startswith('abcdef')
```

Out[52]: False

```
In [53]: 'abc123'.endswith('12')
```

Out[53]: False

```
In [54]: 'Hello world!'.startswith('Hello world!')
```

Out[54]: True

```
In [55]: 'Hello world!'.endswith('Hello world!')
```

Out[55]: True

```
In [ ]: # join() and split()
+ join()
```

```
In [60]: ''.join(['My','name','is','saikumar'])
```

Out[60]: 'Mynameissaikumar'

```
In [57]: ', '.join(['cat','rat','bats'])
```

Out[57]: 'cat, rat, bats'

```
In [59]: ' '.join(['My', 'name', 'is', 'saikumar'])
```

Out[59]: 'My name is saikumar'

```
In [61]: 'Sairam'.join(['My', 'name', 'is', 'saikumar'])
```

Out[61]: 'MySairamnameSairamisSairamsaikumar'

split()

- the 'split()' method splits a into a list . By default, it will use whitespace to separate the items , but you can set another xet character of choice:

```
In [63]: 'My name is Saikumar'.split()
```

Out[63]: ['My', 'name', 'is', 'Saikumar']

```
In [64]: 'MyABCnameABCisABCSaikumar'.split('ABC')
```

Out[64]: ['My', 'name', 'is', 'Saikumar']

```
In [65]: 'My name is Saikumar'.split('m')
```

Out[65]: ['My na', 'e is Saiku', 'ar']

```
In [66]: ' My name is Saikumar'.split()
```

Out[66]: ['My', 'name', 'is', 'Saikumar']

```
In [67]: ' My name is Saikumar'.split(' ')
```

Out[67]: ['', 'My', '', 'name', 'is', '', 'Saikumar']

Justifying text with rust(),ljust(),and center()

```
In [68]: 'Hello'.rjust(10)
```

```
Out[68]: '    Hello'
```

```
In [69]: 'Hello'.rjust(20)
```

```
Out[69]: '                Hello'
```

```
In [70]: 'Hello World'.rjust(20)
```

```
Out[70]: '        Hello World'
```

```
In [71]: 'Hello'.ljust(10)
```

```
Out[71]: 'Hello      '
```

```
In [72]: 'Hello'.center(20)
```

```
Out[72]: '    Hello        '
```

```
In [73]: # An optional second argument to rjust() and ljust() will specify a fill charact
```

```
In [74]: 'Hello'.rjust(20, '*')
```

```
Out[74]: '*****Hello'
```

```
In [75]: 'Hello'.ljust(20, '-')
```

```
Out[75]: 'Hello-----'
```

```
In [77]: 'Hello'.center(20, '=')
```

```
Out[77]: '====Hello===='
```

```
In [ ]: # Removing whitespace with strip(),rstrip(),and lstrip()
```

```
In [83]: spam = '    Hello World    '  
spam.strip()
```

```
Out[83]: 'Hello World'
```

```
In [84]: spam.lstrip()
```

```
Out[84]: 'Hello World    '
```

```
In [85]: spam.rstrip()
```

```
Out[85]: '    Hello World'
```

```
In [90]: spam = 'SpamSpamBaconSpamEggsSpamSpam'  
spam.strip('ampS')
```

```
Out[90]: 'BaconSpamEggs'
```

```
In [91]: # The Count Method
```

```
In [92]: sentence = 'one sheep two sheep three sheep four'
        sentence.count('sheep')
```

Out[92]: 3

```
In [93]: sentence.count('e')
```

Out[93]: 9

```
In [94]: sentence.count('e', 6)
```

Out[94]: 8

```
In [95]: sentence.count('e', 7)
```

Out[95]: 7

```
In [ ]: # Replace Method
```

```
In [96]: text = "Hello, world!"
        text.replace("world", "planet")
```

Out[96]: 'Hello, planet!'

```
In [97]: fruits = "apple, banana, cherry, apple"
        fruits.replace("apple", "orange", 1)
```

Out[97]: 'orange, banana, cherry, apple'

```
In [98]: sentence = "I like apples, Apples are my favorite fruit"
        sentence.replace("apples", "oranges")
```

Out[98]: 'I like oranges, Apples are my favorite fruit'

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```