Manipulating string

In [1]:
```
print("hello there! \nhow are you?\nI\'m doing fine.")
```

```
hello there!
how are you?
I'm doing fine.
```

Raw String

- raw string are mostly used for regular expression

In [2]:
```
print(r'hello there!\nhow are you?\nI\'m doing fine')
```

```
hello there!\nhow are you?\nI\'m doing fine
```

multiline strings

In [3]:
```
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 strings

```
In [4]: spam = 'hello world!'

In [5]: spam[0]

Out[5]: 'h'

In [6]: spam[4]

Out[6]: 'o'

In [7]: spam[-1]

Out[7]: '!'

In [8]: spam[0:5]

Out[8]: 'hello'

In [9]: spam[0:5]

Out[9]: 'hello'

In [10]: spam[6:-1]

Out[10]: 'world'

In [11]: spam[:-1]

Out[11]: 'hello world'

In [12]: spam[::-1]

Out[12]: '!dlrow olleh'

In [13]: fizz=spam[0:5]
         fizz
```

```
Out[13]: 'hello'
```

the in and not in operators

```
In [14]: 'hello' in 'hello world'

Out[14]: True

In [15]: 'hello' in 'hello'

Out[15]: True

In [16]: 'HELLO' in 'hello world'

Out[16]: False

In [17]: '' in 'spam'

Out[17]: True

In [18]: 'cats' not in 'cats and dogs'

Out[18]: False
```

upper(),lower(),and title()

```
In [19]: greet ='hello world!'
         greet.upper()

Out[19]: 'HELLO WORLD!'

In [20]: greet.lower()

Out[20]: 'hello world!'

In [21]: greet.title()
```

```
Out[21]: 'Hello World!'
```

isupper() , islower() methods

```
In [22]: spam = 'Hello world!'
```

```
In [23]: spam.islower()
```

```
Out[23]: False
```

```
In [24]: spam.isupper()
```

```
Out[24]: False
```

```
In [25]: 'HELLO'.isupper()
```

```
Out[25]: True
```

```
In [26]: 'abcd1234'.islower()
```

```
Out[26]: True
```

```
In [27]: '1234'.islower()
```

```
Out[27]: False
```

```
In [28]: '12345'.isupper()
```

```
Out[28]: False
```

The isX string methods

- Method Description
- isalpha() returns `True` if the string consists only of letters.
- isalnum() returns `True` if the string consists only of letters and numbers.

- isdecimal() returns `True` if the string consists only of numbers.
- isspace() returns `True` if the string consists only of spaces, tabs, and new-lines.
- istitle() returns `True` if the string consists only of words that begin with an uppercase letter followed by only lowercase characters

startswith() and endswith()

```
In [29]: 'Hello world!'.startswith('Hello')
```

```
Out[29]: True
```

```
In [30]: 'Hello world!'.endswith('world!')
```

```
Out[30]: True
```

```
In [31]: 'abcd123'.startswith('abcdef')
```

```
Out[31]: False
```

```
In [32]: 'abc123'.endswith('12')
```

```
Out[32]: False
```

```
In [33]: 'Hello world!'.startswith('Hello world!')
```

```
Out[33]: True
```

```
In [34]: 'Hello world!'.endswith('Hello world!')
```

```
Out[34]: True
```

join()

```
In [35]: ''.join(['my','name','is','simon'])
```

```
Out[35]: 'mynameissimon'
```

```
In [36]: ', '.join(['cats','rats','bats'])
```

```
Out[36]: 'cats, rats, bats'
```

```
In [37]: 'ABC'.join(['my','name','is','simon'])
```

```
Out[37]: 'myABCnameABCisABCsimon'
```

spilt()

```
In [38]: 'my name is simon'.split()
```

```
Out[38]: ['my', 'name', 'is', 'simon']
```

```
In [39]: 'myABCnameABCisABCsimon'.split('ABC')
```

```
Out[39]: ['my', 'name', 'is', 'simon']
```

```
In [41]: 'My name is simon'.split('m')
```

```
Out[41]: ['My na', 'e is si', 'on']
```

```
In [42]: ' My name is Simon'.split()
```

```
Out[42]: ['My', 'name', 'is', 'Simon']
```

```
In [43]: ' My name is Simon'.split(' ')
```

```
Out[43]: ['', 'My', 'name', 'is', 'Simon']
```

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

```
In [44]: 'Hello'.rjust(10)
```

```
Out[44]: '     Hello'

In [45]: 'Hello'.rjust(20)

Out[45]: '               Hello'

In [46]: 'Hello world'.rjust(10)

Out[46]: 'Hello world'

In [47]: 'hello'.ljust(10)

Out[47]: 'hello     '

In [48]: 'hello'.ljust(10)

Out[48]: 'hello     '

In [49]: 'hello'.center(20)

Out[49]: '       hello        '

In [50]: 'hello'.rjust(20, '*')

Out[50]: '***************hello'

In [51]: 'hello'.ljust(20,'-')

Out[51]: 'hello---------------'

In [52]: 'hello'.center(20,'=')

Out[52]: '=======hello========'
```

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

```
In [53]: spam = '   hello world!'
         spam.strip()

Out[53]: 'hello world!'
```

```
In [54]: spam.lstrip()

Out[54]: 'hello world!'
```

```
In [55]: spam.rstrip()

Out[55]: '   hello world!'
```

```
In [56]: spam = 'spamspambaconeggssapamspam'
         spam.strip('amps')

Out[56]: 'baconegg'
```

count method

```
In [57]: sentence = 'one sheep two sheep three sheep four'
         sentence.count('sheep')

Out[57]: 3
```

```
In [58]: sentence.count('e')

Out[58]: 9
```

```
In [59]: sentence.count('e',6)

Out[59]: 8
```

replace method

```
In [60]: text = 'hello world!'
         text.replace('world','planet')

Out[60]: 'hello planet!'

In [61]: sentence = 'apple,banana,cherry,apple'
         sentence.replace('apples','oranges')

Out[61]: 'apple,banana,cherry,apple'
```