

String methods.

Strings have a lot of builtin methods here are a few useful ones.

1. str.find(sub) , Returns the lowest index in the str where sub is found, return -1 if sub is not found.

```
>>> str = 'this string contains some words'
>>> str.find('contain')
12
>>> str.find('foo')
-1
```

2. str.format(args) , Returns the formatted version of format string str.

```
>>> s = 'my name is {}'
>>> s.format('parham')
'my name is parham'
```

note that the original string is unchanged and a new string is created.

```
>>> s
'my name is {}'
```

format method is mostly used in string literals.

```
>>> '{} and {} are example of programming languages'.format('python', 'C++')
'python and C++ are example of programming languages'
```

3. str.replace(old, new) , Returns a copy of str where all occurrences of old is replaced by new.

```
>>> s = 'me and you and all of us'
>>> s.replace('and', ', ')
'me , you , all of us'
```

4. str.strip([char]) , Returns a copy of str with all leading and trailing characters from [char] string is removed.defaults to removing whitespace if no argument is passed.

```
>>> s = '    this has a lot of leading white space  '
>>> s.strip()
'this has a lot of leading white space'
>>> t = '(a + b)'
>>> t.strip('()')
'a + b'
```

5. str.join(iterable) , Returns a string of strings in iterable separated by str.

pyStringMethods.txt

```
>>> list = ['aaa', 'bb', 'c']
>>> ', '.join(list)
'aaa, bb, c'
>>> list = ['name', 'id', 'order']
>>> ''.join(list)
'nameidorder'
```

6. `str.split(sep = None)` , Returns a list of the words in `str`, using `sep` as the delimiter string. if `sep` is `None`, splits `str` on whitespace.

```
>>> '1,2,3'.split(',')
['1', '2', '3']
>>> 'Returns a list of the words'.split()
['Returns', 'a', 'list', 'of', 'the', 'words']
```

For a complete list of string methods visit:
<https://docs.python.org/3/library/stdtypes.html#string-methods>
