

# Python小练习：处理字符串

作者：凯鲁嘎吉 - 博客园 <http://www.cnblogs.com/kailugaji/>

介绍3种处理字符串的方式：1. 将英语名词单数转化为复数形式(仅适用于一般形式)，2. 将字符串(带有下换线\_)转化为驼峰化形式，3. 左对齐右对齐。

## 1. word\_test.py

```
1 # -*- coding: utf-8 -*-
2 # Author: 凯鲁嘎吉 Coral Gajic
3 # https://www.cnblogs.com/kailugaji/
4 # Python小练习：处理字符串
5
6 # 名词的复数形式
7 # 这是一个Python函数，用于将单词转换为其不同的形式。它接受一个单词作为参数，并返回该单词的不同形式。
8 # 该函数首先检查单词是否以“y”结尾，如果是，则返回单词的前半部分，即“ies”。
9 # 如果单词以“x”结尾，则返回单词的后半部分，即“es”。
10 # 如果单词以“an”结尾，则返回单词的前半部分，即“en”。
11 # 否则，返回单词本身，并在单词后添加“s”。
12 def _plural(word):
13     if word.endswith('y'):
14         return word[:-1] + 'ies'
15     elif word[-1] in 'sx' or word[-2:] in ['sh', 'ch']:
16         return word + 'es'
17     elif word.endswith('an'):
18         return word[:-2] + 'en'
19     else:
20         return word + 's'
21
22 # 这段代码定义了一个名为_under_score_to_camel的函数
23 # 它接受一个参数word，并返回该字符串的驼峰化形式。
24 # 该函数使用了split方法将字符串按照_分割成多个部分，并将每个部分转换成大写形式
25 # 然后使用join方法将所有部分连接起来，形成驼峰化的形式。
26 def _under_score_to_camel(word):
27     parts = word.split('_')
28     parts = [p[0].upper() + p[1:] for p in parts]
29     return ''.join(parts)
30
31 # 字符串对齐
32 # 使用了 rjust() 方法来对键值对进行右对齐
```

```

33 # 使用了 format() 方法来对键值对进行左对齐
34 # 使用了 round() 函数来对结果进行四舍五入
35 def format_for_process(params):
36     new_params = []
37     for k, v in params.items():
38         k = k.rjust(18) # 右对齐
39         # rjust() 方法是向字符串的左侧填充指定字符，从而达到右对齐文本的目的。
40         v = '{:<12}'.format(round(v, 6))[:12] # 左对齐，并且保留6位小数
41         new_params.append([k, v])
42     return new_params
43
44 word = 'agent'
45 print(word, '->', _plural(word)) # 输出: agents
46 word = 'policy'
47 print(word, '->', _plural(word)) # 输出: policies
48 word = 'lynx'
49 print(word, '->', _plural(word)) # 输出: lynxes
50 word = 'woman'
51 print(word, '->', _plural(word)) # 输出: women
52 print('-----')
53 word = 'book_story'
54 print(word, '->', _under_score_to_camel(word)) # 输出: BookStory
55 word = 'activity'
56 print(word, '->', _under_score_to_camel(word)) # 输出: Activity
57 word = 'lynx_woman_hello'
58 print(word, '->', _under_score_to_camel(word)) # 输出: LynxWomanHello
59 print('-----')
60 params={"name": 127.1234567587, "height": 185.123456789, "score": 10.123456852}
61 new_params = format_for_process(params)
62 print(new_params)

```

## 2. 结果

D:\ProgramData\Anaconda3\python.exe "D:/Python code/2023.3 exercise/time/word\_test.py"

```

agent -> agents
policy -> policies
lynx -> lynxes
woman -> women

```

```

-----
book_story -> BookStory
activity -> Activity
lynx_woman_hello -> LynxWomanHello

```

```

-----
[['name', '127.123457'], ['height', '185.123457'], ['score', '10.123457']]

```

Process finished with exit code 0

后续如果有更多字符串处理的案例，还会继续补充完整。