

小白必备字符串对象练习

39 Python 中有字符串类型吗？单个字符如何被解释？

Python 中没有像 C++ 那样表示的字符类型 (char)，不管是单个字符或多个字符都会被统一解释为 `str` 对象。

40 字符串方法 `strip`, `replace`, `join` 各自实现什么功能？

`strip` 用于去除字符串前后的空格：

```
In [1]: ' I love python\t\n '.strip()
Out[1]: 'I love python'
```

[复制](#)

`replace` 用于字符串的替换：

```
In [2]: 'i love python'.replace(' ', '_')
Out[2]: 'i_love_python'
```

[复制](#)

`join` 用于合并字符串：

```
In [3]: '_'.join(['book', 'store', 'count'])
Out[3]: 'book_store_count'
```

[复制](#)

41 子串判断的两种方法

判断 `a` 串是否为 `b` 串的子串：

方法 1，使用 `in`：

```
In [16]: a = 'our'
In [17]: b = 'flourish'

In [18]: r = True if a in b else False

In [19]: r
Out[19]: True
```

[复制](#)

方法 2，使用方法 `find`，返回字符串 `b` 中匹配子串 `a` 的最小索引。

```
In [16]: a = 'our'
In [17]: b = 'flourish'

In [21]: b.find(a)
Out[21]: 2
```

[复制](#)

42 去除字符串开始和结尾处空格

清洗字符串时，位于字符串开始和结尾的空格，有时需要去掉，`strip` 方法能实现。

如下字符串，使用 `strip`，清理字符串开头和结尾的空格和制表符。

```
In [24]: a = ' \tI love python \b\n'

In [25]: a
Out[25]: ' \tI love python \x00\n'

In [26]: a.strip()
Out[26]: 'I love python \x00'
```

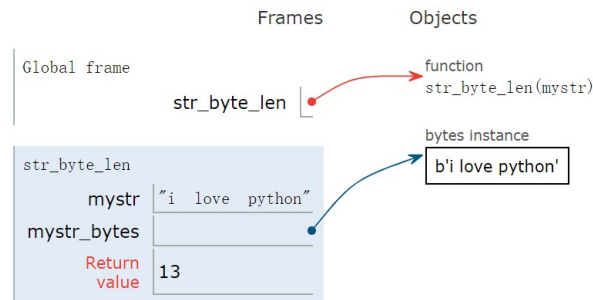
[复制](#)

43 求字符串的字节长度

`encode` 方法对字符串编码后

```
In [12]: def str_byte_len(mystr):
...:     mystr_bytes = mystr.encode('utf-8')
...:     return (len(mystr_bytes))

In [13]: str_byte_len('i love python')
Out[13]: 13
```

[复制](#)

44 反转字符串的两种方法

```
In [1]: s = "python"

# 方法1
In [6]: rs = ''.join(reversed(s))

In [7]: rs
Out[7]: 'nohtyp'
```

[复制](#)

```
#方法2
In [5]: s[::-1]
```

[复制](#)

39 Python 中有字...
40 字符串方法 strip...
41 子串判断的两种...
42 去除字符串开始...
43 求字符串的字节...
44 反转字符串的两...
45 字符串切片的案例
46 join 串联字符串
47 分割字符串
48 字符串替换
49 字符串处理成字典

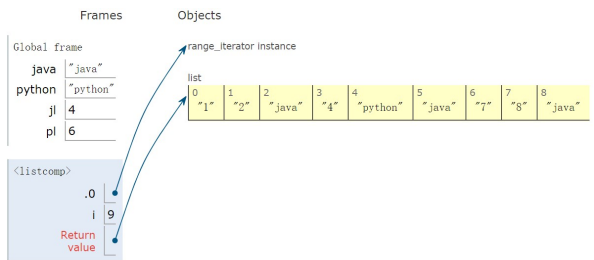
```
Out[5]: 'nohtyp'
```

45 字符串切片的案例

生成 1 到 15 的序列，并在满足条件的索引处，替换为 java 或 python 。

```
In [15]: java,python = "java", "python"
In [16]: jl,pl=len(java),len(python)
In [17]: [str(java[i%3*jl:] + python[i%5*pl:] or i) for i in range(1,10)]
```

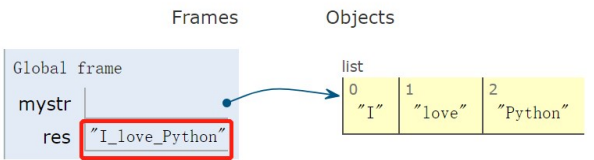
输出结果，如下图：



46 join 串联字符串

用下划线 _ 连接字符串 mystr：

```
In [8]: mystr = ['I','love','Python']
In [9]: res = '_'.join(mystr)
In [10]: res
Out[10]: 'I_love_Python'
```



47 分割字符串

根据指定字符或字符串，分割一个字符串时，使用方法 split

join 和 split 可看做一对互逆操作

```
In [23]: 'I_love_python'.split('_')
Out[23]: ['I', 'love', 'python']
```

48 字符替换

字符串替换，使用 replace 方法。

如下字符串，小写的 o 全部替换为大写的 O。

```
In [14]: s = 'i love python'.replace('o', 'O')
In [15]: s
Out[15]: 'i lOve pythOn'
```

49 字符串处理成字典

输入串："k0:10|k1:2|k2:11|k3:5"

输出字典：{'k0':10,'k1':2,...}

第一层 split，根据分隔符 |，分割出 k0:10，k1:2，k2:11，k3:5

第二层 split，根据分隔符：，分割出新字典的键值对

使用字典生成式，得到结果，也就是一个新字典

```
In [30]: m = map(lambda x: x.split(':'),'k0:10|k1:2|k2:11|k3:5'.split('|'))
In [31]: {mi[0]:int(mi[1]) for mi in m}
Out[31]: {'k0': 10, 'k1': 2, 'k2': 11, 'k3': 5}
```

分享交流

我们为专栏付费读者创建了微信交流群，以便更有针对性地讨论专栏相关的问题。入群方式
请添加 GitChat 小助手伽利略的微信号：linmicc（或扫描以下二维码），然后给小助手发「866」消息，即可拉你进群~





下一章

互动评论



说点什么

评论



阿拉斯加猪
亲好友频繁

14 小时前

鼓掌



weixin_46187354
day3打卡

2 个月前

鼓掌



The Scrapper
day3,不错的练习

2 个月前

鼓掌



存

