

量化金融 python 基础 1-String

2023 年 5 月 9 日

1 字符串操作

```
[14]: # 以沪深 300 指数期货, 中证 500 指数期货, 上证 50 指数期货为例。
```

```
A = " if2209.CFFEX, Ic2209.CFFEX, IH2209.CFFEX "  
A
```

```
[14]: ' if2209.CFFEX, Ic2209.CFFEX, IH2209.CFFEX '
```

```
[4]: A.split(",") # 拆分后返回的是一个列表
```

```
[4]: ['if2209.CFFEX', ' Ic2209.CFFEX', ' IH2209.CFFEX']
```

```
[5]: B, C, D = A.split(",") # 将三个元素分别赋予 B, C, D  
B
```

```
[5]: 'if2209.CFFEX'
```

```
[6]: ", ".join([B, C, D]) # 将三个元素重新组合成 A
```

```
[6]: 'if2209.CFFEX, Ic2209.CFFEX, IH2209.CFFEX'
```

```
[7]: # index 返回第一个匹配的字符位置  
A.index("2")
```

```
[7]: 2
```

```
[8]: # str.find 也是返回第一个匹配的字符位置  
# 和 index 不同的是, 当寻找的字符不存在时, index 会报错, find 会返回 "-1"  
A.find("2")
```

```
[8]: 2
```

```
[9]: A.find("3")
```

```
[9]: -1
```

```
[10]: # string.count 返回给定字符出现的次数  
A.count("I")
```

```
[10]: 2
```

```
[12]: # string.replace 替换所有符合给定的字符  
print(A.replace("I", "Z")) # 将 I 替换成 Z  
print(A.replace("I", "Z", 1)) # 仅替换第一次出现的 I
```

```
if2209.CFFEX, Zc2209.CFFEX, ZH2209.CFFEX  
if2209.CFFEX, Zc2209.CFFEX, IH2209.CFFEX
```

```
[15]: #string.strip() 用于删除字符串开头和结尾处的空白字符。空白字符包括空格、制表符、换行符  
print(A) # A 的开头和结尾都有一个空格符  
print(A.strip())
```

```
if2209.CFFEX, Ic2209.CFFEX, IH2209.CFFEX  
if2209.CFFEX, Ic2209.CFFEX, IH2209.CFFEX
```

```
[17]: # strip 只能删除字符串开头和结尾的空白符号，如果想删除字符串中间的空白符，可以搭配  
split 和 join  
print(A) # 每个期货合约代码之间都有一个空格符  
print("".join(A.split(" ")))
```

```
if2209.CFFEX, Ic2209.CFFEX, IH2209.CFFEX  
if2209.CFFEX,Ic2209.CFFEX,IH2209.CFFEX
```

```
[20]: print(A)  
print(A.lower()) # 所有字母改成小写  
print(A.upper()) # 所有字母大写  
print(A.title()) # 字符串首字母大写
```

```
if2209.CFFEX, Ic2209.CFFEX, IH2209.CFFEX  
if2209.cffex, ic2209.cffex, ih2209.cffex  
IF2209.CFFEX, IC2209.CFFEX, IH2209.CFFEX  
If2209.Cffex, Ic2209.Cffex, Ih2209.Cffex
```

[23]: `# string.ljust()` 将字符串左对齐并在右侧填充指定的字符, 以达到指定的宽度。`l` 指 *left* 左对齐的意思。

```
print(B)
print(B.ljust(15, "*")) # 左对齐右侧填充至 15 个单位
```

```
if2209.CFFEX
if2209.CFFEX***
```

[24]: `#` 有 `ljust` 对应的就有 `rjust`。同理, `rjust` 将字符串右对齐, 向左填充。

```
print(B)
print(B.rjust(15, "*"))
```

```
if2209.CFFEX
***if2209.CFFEX
```

[28]: `# string.format`

```
print(B, C, D)
print("2022 年 9 月到期的沪深 300 指数期货是{}, 中证 500 期货是{}, 上证 50 期货是{}".format(B, C, D))
```

```
if2209.CFFEX  Ic2209.CFFEX  IH2209.CFFEX
2022 年 9 月到期的沪深 300 指数期货是 if2209.CFFEX, 中证 500 期货是 Ic2209.CFFEX, 
↪上证 50 期货是 IH2209.CFFEX
```

[33]: `#` 这里提一下制表符和换行符

```
# python 中制表符以 "\t" 表示, 用于输出水平对齐的列。
# 换行符以 "\n" 表示。
```

```
print("2022 年 9 月到期的股指期货有:\n沪深 300 指数期货{}\n中证 500 期货{}\n\n↪上证 50 期货{}\n".format(B, C, D))
print("加上制表符")
print("2022 年 9 月到期的股指期货有:\n沪深 300 指数期货\t{}\n中证 500 期货\t{}\n\n↪上证 50 期货\t{}\n".format(B, C, D))
```

```
2022 年 9 月到期的股指期货有:
沪深 300 指数期货 if2209.CFFEX,
中证 500 期货 Ic2209.CFFEX,
上证 50 期货 IH2209.CFFEX
```

```
加上制表符
```

2022 年 9 月到期的股指期货有：

沪深 300 指数期货 if2209.CFFEX,

中证 500 期货 Ic2209.CFFEX,

上证 50 期货 IH2209.CFFEX

1.1 字符串操作在程序化交易中经常被用于调整合约名称。

需要注意的是，各个期货交易所对合约名称有自己的规定，如下：

上期/能源所：小写 +4 个数字（rb1909 代表 rb 品种，19 年 9 月份到期）

中金所：大写 +4 个数字

郑商所：大写 +3 个数字 (TA001 代表 TA 品种，20 年 01 月份到期)

大商所：小写 +4 个数字

期权合约如下：标的期货合约 + 看涨/看跌 + 行权价

上期所：小写 + 四个数字 +C（或者 P）+ 行权价，如 cu1912C43000

大商所：小写 + 四个数字 + -C-（或者-P-）+ 行权价，如 c2001-C-1800

郑商所：大写 + 三个数字 +C（或者 P）+ 行权价，如 CF001C11200

中金所：大写 + 四个数字 + -C-（或者-P-）+ 行权价，如 IO1908-C-2100

2 Summary

- string.split()
- string.join()
- string.index()
- string.find()
- string.count()
- string.replace()
- string.strip()
- string.lower()
- string.ljust(), string.rjust()
- string.title()
- format using {}.format

- 换行符, 制表符