



python基础字符串处理

readme



本文档主要介绍如何使用python进行字符串的处理

ASCII、UNICODE、UFT-8（了解即可）

计算机智能识别0和1，为了让计算机能够识别文字，可以给字母、数字进行编号，比如'A'就是65，转成二进制就是：0100 0001。

- ASCII：字节表示方式是：0XXX XXXX，第一位为0，剩下7位二进制范围是0~127。仅仅支持字母、数字和标点符号
- UNICODE：ASICII编码仅仅考虑了英文，那么为了统一所有文字（比如中文、韩文等）的编码，就引入了UNICODE编码。UTF-8、UFT-16、UTF-32都是UNICODE的编码方式。UFT-8：8-bit Unicode Transformation Format

字符串前面+r、f

常用转义字符串

转义字符	描述
\	续航符(下有示例)
\\	反斜杠符号
\'	单引号
\"	双引号
\n	换行
\t	横向制表符（学到这里的人可以在下面补充下\t和\v的用法）
\v	纵向制表符

```
1 #续航符 \ 示例
2 Q = '你好
3 有什么需要帮助的吗'
4 print(Q)
5
6 # 会报错
7
8 # 正确输入
9 Q = '你好\
10 有什么需要帮助的吗'
11 print(Q)
12 # output: 你好有什么需要帮助的吗
13
14 # 用于将一段非常长的文本时换行，下面给一个例子
15 result = '如果这样实现，\
16 编辑器就会报错，\
17 因为这个字符串是用单引号括起来的，\
18 若果中间又出现一个单引号，\
19 到底三个单引号中哪两个单引号构成一个字符串呢？\
20 这里面就出现了歧义，\
21 而计算机最怕的就是歧义。\'
22 为了解决这种情况，\
23 我们就需要用到转义字符了。\'
24
25
26 # 如果不用 \ 的话就需要下面特别长的一行
27 result = '如果这样实现，编辑器就会报错，因为这个字符串是用单引号括起来的，若果中间又出现
28 一个单引号，到底三个单引号中哪两个单引号构成一个字符串呢？这里面就出现了歧义，而计算机最怕
的就是歧义。为了解决这种情况，我们就需要用到转义字符了。'
```

1 # \t 和 \v 的用法

字符串前面+f

以 `f` 或 `F` 开头的字符串，其中以 `{}` 包含的表达式会进行值替换，**实际生活中使用较多**

```
1 name = '小明'
2 age = '15'
3 output = f'我叫{name}，今年{age}岁'
4 print(output)
5 # output: 我叫小明，今年15岁
6
7 name = '小明'
8 age = '15'
9 output = '我叫{name}，今年{age}岁'
10 # output: 我叫{name}，今年{age}岁
```

字符串前面+r

在字符串前加 `r` 可防止字符串转义

```
1 Q = '你好\n有什么需要帮助的吗'
2 print(Q)
3
4 # output:
5 # 你好
6 # 有什么需要帮助的吗
7
8 Q = r'你好\n有什么需要帮助的吗'
9 print(Q)
10 # output:
11 # 你好\n有什么需要帮助的吗
```

python字符处理

字符串切片

```
1 str1 = 'hello world'
2 str1[0]
3 # output : 'h'
4
5 str1[1:5]
6 # output : 'ello'
7
8 str1[-2]
9 # output : 'l'
10
11 str1[-1:0:-1]
12 # output : 'dlrow olle'
```

将两个字符串拼接 str1+str2

```
1 a = 'hello'
2 b = 'world'
3 a+b
4 # output: 'helloworld'
5
6 a+' '+b
7 # output: 'hello world'
8
9 a+' the '+b
10 # output: 'hello the world'
```

字符串分割 str.split('\n')

```
1 # 按空格分割
2 str1 = 'hello the world'
3 str1.split(' ')
4 # output: ['hello', 'the', 'world']
5
6 # 按换行符分割
```

```

7 str1 = '''hello
8 the
9 world'''
10 str1.split('\n')
11 # output: ['hello', 'the', 'world']
12
13 # 按逗号分割
14 str1 = 'hello,the,world'
15 str1.split(',')
16 # output: ['hello', 'the', 'world']

```

合并字符串 ''.join(list1)

```

1 list1 = ['hello', 'the', 'world']
2 ''.join(list1)
3 # output: 'hellotheworld'
4
5 list1 = ['hello', 'the', 'world']
6 ','.join(list1)
7 # output: 'hello,the,world'

```

字符串替换 str.replace('str1','str2')

```

1 str1 = 'hello a world, a world'
2 str1.replace('a','the')

```

- output: 'hello the world, the world'
- 可以将语句中不要的词语去掉，比如str1.replace('a','')
- 这个函数被正则表达式里的 re.sub()完爆

小写转大写 str.upper()

```

1 str1 = 'hello a world, a world'

```

```
2 str1 = str1.upper()
3 str1
```

- output: 'HELLO A WORLD, A WORLD'

大写转小写 str.lower()

```
1 str1 = 'HELLO A WORLD, A WORLD'
2 str1 = str1.lower()
3 str1
```

- output: 'hello a world, a world'

字符串首字母大写 str.capitalize()

```
1 str1 = 'how are you? i am fine'
2 str1.capitalize()
```

- output: 'How are you? i am fine'
- 注意：只转换整个字符串的第一个字母的大小写，而不是每一句的首字母都转换

判断是否以某个字符开头 str.startswith('ap')

```
1 str1 = 'apple is my favorite'
2 str1.startswith('ap')
```

- output: True

判断是否以某个字符结尾 str.endswith('te')

```
1 str1 = 'apple is my favorite'
```

```
2 str1.endswith('te')
```

- output: True
- 这两个函数往往会结合使用，结合cot和正则表达式超级好用

将字符串居中str1.center(50,'-')

```
1 str1 = 'hello world'
2 str1.center(50,'-')
```

- '-----hello world-----'

结语

- 这里只包含了python字符的基本处理方法，下一步请前往学习[正则表达式](#)