

【Python】序列对象——字符串string

一、字符串

```
1 """
2 字符串支持单引号、双引号、三引号
3 """
4 message1 = 'Hello!'
5 message2 = "Hello Python World!"
6 message3 = """Hello
7             Python
8             World!"""
```

```
>>> message3 = """Hello
...             Python
...             World!"""
>>> message3
'Hello\n                Python\n                World!'\n
>>> print(message3)
Hello
                Python
                World!
```

CSDN @fftx_00

```
1 """
2 实现存储单or双引号:
3 1.混用
4 2.反斜杠转义
5
6 """
7 message = "The language 'python' is named after Monty Python, not the snake."
8
9 message = "The language \'python\' is named after Monty Python, not the snake."
```

```
>>> s = 'this\
is\
a\
test.\
'
>>> s
'thisisa test. '\n
>>> print(s)
thisisa test.
```

使用\表示行未完

```
>>> s = r'this\nis\na\ntest/'
>>> s
'this\\nis\\na\\ntest/'
>>> print(s)
this\nis\na\ntest/
```

使用r不转义

二、字符串函数

字符串是不可变对象，不支持元素赋值和修改

因此所有的字符串函数str.name() 并不会修改原字符串，而是产生一个新字符串

1.重复 *

```

1 | # 字符串片段重复多次
2 | str1 = 'abcd '
3 | str2 = str1*5
4 | print(str2)

```

2.连接 +

```

1 | # 字符串片段拼接
2 | str1 = 'abcd+'
3 | str2 = 'efgh'
4 | str3 = str1+str2
5 | print(str3)

```

3.大小写

```

1 | # 每个单词首字母大写
2 | name = "ada lovelace"
3 | print(name.title())
4 | # 字符串全部大写or小写
5 | print(name.upper())
6 | print(name.lower())

```

4.空白

```

1 | favorite_language = ' python '
2 | print(">"+favorite_language+"")
3 | # 删除字符串末尾的空白
4 | print(">"+favorite_language.rstrip()+"")
5 | # 删除字符串开头的空白
6 | print(">"+favorite_language.lstrip()+"")
7 | # 删除字符串两头的空白
8 | print(">"+favorite_language.strip()+"")

```

5.查找子串 in,str.find(),str.index(),str.count()

```

1 | # 查找子串
2 | str1 = 'abcd '
3 | str2 = str1*20+'aabbd'
4 |
5 | 'aabb' in str2           # 是否存在
6 |
7 | str2.find('aabb')        # 在哪里
8 | str2.index('aabb')       # 在哪里
9 | str2.rindex('aabb')      # 从右边找，在哪里(最后出现的位置)
10 |
11 | str2.find('aabb',3,5)    ##从3位置开始找,5位置结束，找不到返回-1
12 | str2.index('aabb',3,5)   ##从3位置开始找,5位置结束，找不到报错
13 |
14 | str2.count('aabb')       # 重复几次

```

6.字符串长度 len(str)

```

1 | # 字符串长度
2 | str1 = 'abcd '
3 | str2 = str1*20+'aabbd'
4 | print(len(str2))

```

7.字符串求值 eval(str)

```

1 | # 字符串求值eval()
2 | print(eval('3+4j'))          # 对字符串求值得到复数
3 | print(eval('8**2'))          # 计算表达式8**2的值
4 | print(eval('[1, 2, 3, 4, 5]')) # 对字符串形式求值得到列表
5 | print(eval('{1, 2, 3, 4}'))  # 对字符串求值得到集合

```

8.字符串索引、切片 [::]

```

>>> month = ['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP', 'OCT', 'NOV', 'DEC']
>>> month
['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP', 'OCT', 'NOV', 'DEC']
>>> month[3:6]
['APR', 'MAY', 'JUN']
>>> month[3:5]
['APR', 'MAY']
>>> '26C'[0:-1]
'26'
>>> '8C'[0:-1]
'8'
>>> '26C'[:-1]
'26'
>>> 'Zhejiang'[3:8]
'jiang'
>>> 'Zhejiang'[8]

```

CSDN @fftx_00

```

1 | """
2 | 取最后一个元素
3 | """
4 | str[-1]
5 | str[len(str)-1]

```

```

1 | """
2 | 切片倒转
3 | """
4 | str[::-1]

```

9.替换 str.replace()

```

>>> s
' Hello World '
>>> s.replace(' ', '-')
'-Hello-World-'
>>> s.replace(' ', '--')
'--Hello--World--'
>>> s.replace(' ', '')
'HelloWorld'

```

10.str()

```

>>> str(123)
'123'
>>> str(123.45)
'123.45'

```

11.字符串绑定变量

```
>>> 'I am %d year old.' % 18
'I am 18 year old.'
>>> 'I am %d years old and weight %d kg.' % (18, 50)
'I am 18 years old and weight 50 kg.'
>>> 'it is %.1fC.' % 30.5
'it is 30.5C.'
>>> 'it is %.1fC.' % 30.5123
'it is 30.5C.'
>>> 'it is %.1fC.' % 30.579
'it is 30.6C.'
>>> 'I am %10d year old' % 18
'I am          18 year old'
>>> 'it is %10.1fC.' % 30.5123
'it is          30.5C.'
```

%d

%.1f

%10.1f

截图(A)

CSDN @fftx_00

```
>>> val=1.2554654654
>>> 'sdfahohafhjoi%10.1fsafhoif' % val
'sdfahohafhjoi          1.3safhoif'
```

12.str.split()

string自动识别空格转成list

```
>>> s = 'this is a test'
>>> s.split()
['this', 'is', 'a', 'test']
>>> list(s)
['t', 'h', 'i', 's', ' ', ' ', 'i', 's', ' ', ' ', 'a', ' ', ' ', 't', 'e', 's', 't']
>>> s = '12:35'
>>> s.split(':')
['12', '35']
>>> s = '12::35'
>>> s.split('::')
['12', '35']
>>> s
'12::35'
>>> s.split(':')
['12', '', '35']
```

13.str.join()

str用来join列表

```
>>> s = 'this is a test'
>>> s
'this is a test'
>>> t = s.split()
>>> t
['this', 'is', 'a', 'test']
>>> ' '.join(t)
'this is a test'
```

14.str.maketrans(),str.translate()

```
>>> table = ''.maketrans('0123456789', '零一二三四伍陆柒捌玖')
>>> print('Tel:30647359'.translate(table))
Tel:三零陆四柒三伍玖
```

CSDN @fftx_00

15.ljust(), rjust(), center()

```
>>> print('居左'.ljust(20)+'结束')
居左                结束
>>> print('居右'.rjust(20, '#'))      # 左侧使用井号填充
#####居右
>>> print('居中'.center(20, '='))      # 两侧使用等号填充
=====居中=====
```

CSDN @fftx_00