字符串类型

字符串类型

字符串的定义:

- 第一种方式:

str1 = 'our company is westos'

- 第二种方式:

str2 = "our company is westos"

- 第三种方式:

str3 = """our company is westos"""

转义符号

一个反斜线加一个单一字符可以表示一个特殊字符，通常是不可打印的字符

\n:	代表换行符	\":	代表双引号本身
\t:	代表tab符	\':	代表单引号本身

```
>>> say = 'let's go'
  File "<stdin>", line 1
    say = 'let's go'
              ^
SyntaxError: invalid syntax
>>> say = 'let\'s go'
>>> print say
let's go
>>> say = 'hello python\n'
>>> print say
hello python

>>> say = '\thello python\t'
>>> print say
        hello python
```

三重引号

- 块注释:多行代码注释
- 函数的doc文档:讲函数时会说到
- 字符串格式化

```
>>> say = """
...     hello world
... """
>>> print say

    hello world

>>> say
'\n\thello world\n'
```

字符串操作预览

字符串属于序列，序列支持的操作如下:

- 索引　　切片
- 判断子串　　重复
- 连接　　计算长度

索引

- 索引(s[i])：获取特定偏移的元素
- 给出一个字符串，可输出任意一个字符，如果索引为负数，

就是相当于从后向前数。

索引理解						
字符串	h	e	l	l	o	\n
索引	0	1	2	3	4	5
索引	-6	-5	-4	-3	-2	-1

索引

```
>>> str = 'hello'
>>> str[0]
'h'
>>> str[-1]
'o'
>>> str = 'hello '
>>> str[0]
'h'
>>> str[-1]
' '
>>> str[-2]
'o'
```

切片

切片S[i:j]提取对应的部分作为一个序列：

- 上边界并不包含在内;
- 如果没有给出切片的边界，切片的下边界默认为0，上边界为字符串的长度;
- 扩展的切片S[i:j:k],其中i,j含义同上，k为递增步长;

切片

s[:]获取从偏移量为0到末尾之间的元素，是实现有效拷贝的一种方法

```
>>> s = 'hello'
>>> s[1:3]
'el'
>>> s[1:]
'ello'
>>> s[:3]
'hel'
>>> s[:-1]
'hell'
>>> s[:]
'hello'
```

判断子串

判断一个sub字符串是不是属于s字符串：

- sub in s
- sub not in s

```
>>> s = 'westos'
>>> 's' in s
True
>>> 'hel' in s
False
>>> 'hel' not in s
True
```

重复、连接及计算长度

```
>>> print '===='*10
========================================
>>> print 'hello'+'world'
helloworld
>>> print 'hello'+' world'
hello world
>>> s = 'hello world'
>>> len(s)
11
```

字符串的类型转换

str(obj)　将其他类型内容转换为字符串

int(obj)　将字符串转换为为整数

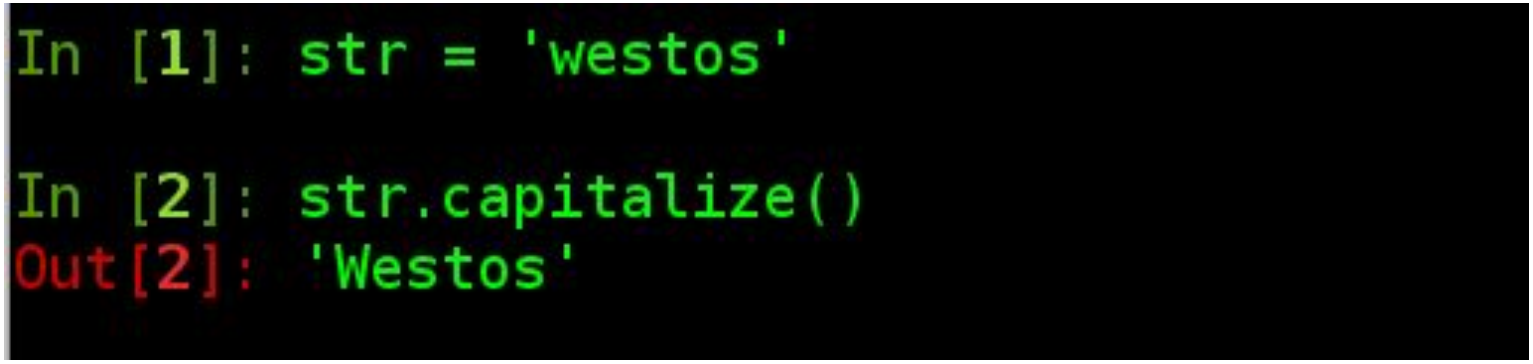
float(obj)　将字符串转换为浮点型

long(obj)　将字符串转换为长整型

字符串常用操作:

str.capitalize()

- 将字符串首字母大写，并返回新的首字母大写后的字符串;



字符串常用操作：

str.center(width[,fillchar])

- 返回一个长为width的新字符串，在新字符串中原字符居中，其他部分用fillchar指定的符号填充，未指定时通过空格填充。

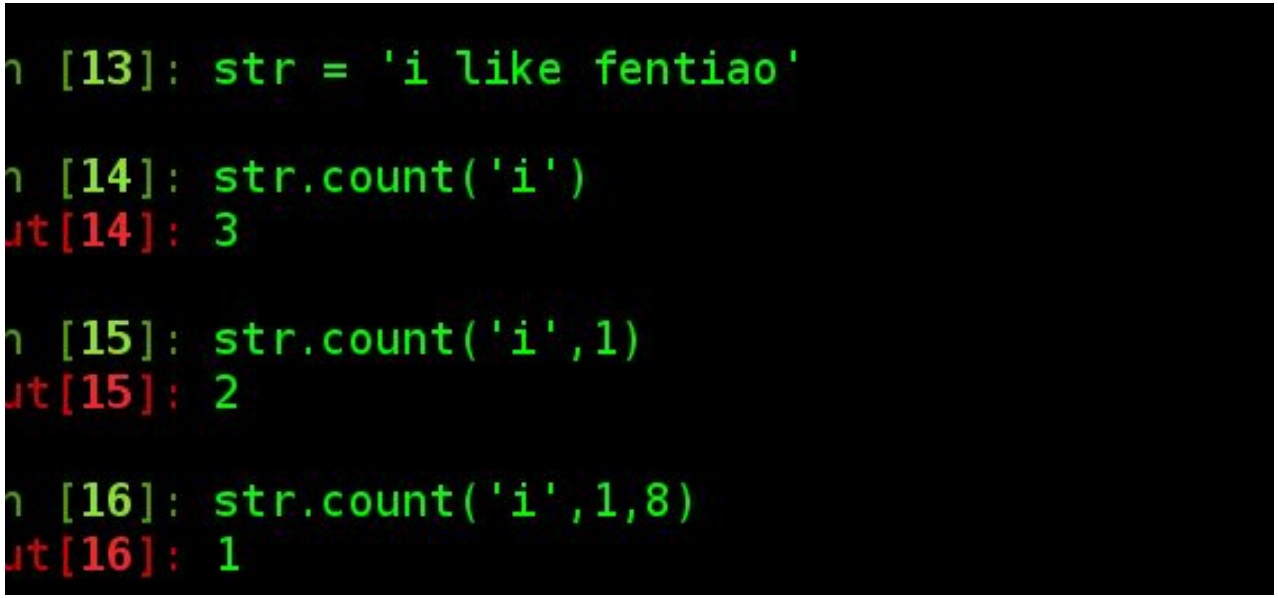
```
In [10]: str = 'westos'

In [11]: str.center(50,'*')
Out[11]: '**********************westos**********************'
```

字符串常用操作：

str.count(sub[, start[, end]]) -> int

- 返回sub在str中出现的次数,如果start与end指定，则返回指定范围内的sub出现次数。



字符串常用操作：

str.endswith(suffix[, start[, end]])

- 判断字符串是否以suffix结束，如果start和end指定，则返回str中指定范围内str子串是否以suffix结尾，如果是，返回True；否则返回False

str.startswith(prefix[, start[, end]])

```
In [20]: str = 'i like fentiao'

In [21]: str.endswith('iao')
Out[21]: True

In [22]: str.endswith('iao',1,5)
Out[22]: False
```

字符串常用操作:

str.find(sub[,start[,end]])

- 判断sub是否在str中，存在返回索引值，不存在返回-1.

str.index(sub[,start[,end]])

- 与find方法函数功能相同，如果sub不存在时抛出ValueError异常;

字符串常用操作:

str.isalnum()　　//判断是否都是字母或数字

str.isalpha()　　//判断是否都是字母

str.isdigit()　　//判断是否都是数字

str.islower()　　//判断是否都是小写

str.isspace()　　//判断是否都是英文空格

str.istitle()　　//判断是不是都是标题(有大小写)

str.isupper()　　//判断是不是都为大写字母

字符串常用操作:

str.join(seq)

- 以str作为分隔符，将序列seq中的所有元素合并为一个新的字符串。

str.replace(old,new[,count])

- 将str中的old字符串替换为new字符串，并将替换后的新字符串返回，如果count指定，则只替换前count个字符串

字符串常用操作:

str.split([sep[,maxsplit]])

- 以sep字符串作为分割符对str进行切割，默认为空格;

- maxsplit代表切割的此处

字符串常用操作：

str.strip([chars])

- 返回一字符串，将str中首尾包含指定的chars字符删除的字符串，未指定时，删除首尾的空格。

over !