|  |
| --- |
| 字符串操作 |
| **作者：李昂**  **归档：学习笔记**  **日期：2018/11/7** |
| **快捷键：**  Ctrl + 1 标题1  Ctrl + 2 标题2  Ctrl + 3 标题3  Ctrl + 4 实例  Ctrl + 5 程序代码  Ctrl + 6 正文  Ctrl + q 正文注释 |
| **格式说明：**  蓝色字体：注释  黄色背景：了解  绿色背景：重要  红色背景：特别重要 |

目 录

[字符串操作 I](#_Toc529389307)

[第1章 字符串操作 1-2](#_Toc529389308)

[1.1 .capitalize 第一个字符大写 1-2](#_Toc529389309)

[1.2 .count 统计字符数量 统计某个字符出现的数量 1-2](#_Toc529389310)

[1.3 .center 自动填充 1-2](#_Toc529389311)

[1.4 .endswith 判断是否以xx结尾 1-3](#_Toc529389312)

[1.5 .find 查找首个字符的索引 1-4](#_Toc529389313)

[1.6 .format 格式化输出 1-4](#_Toc529389314)

[1.7 .isalnum 判断是否为阿拉伯字符 1-4](#_Toc529389315)

[1.8 .isalpha 判断是否为纯英文字符 1-4](#_Toc529389316)

[1.9 .isdecimal 判断是否为十进制 1-4](#_Toc529389317)

[1.10 .isdigit 判断是否为整数 1-4](#_Toc529389318)

[1.11 .isidentifier 判断是否为合法标识符（是不是正常变量名） 1-5](#_Toc529389319)

[1.12 .islower 判断是否为小写 1-5](#_Toc529389320)

[1.13 .isnumeric 判断是否为数字，只能有数字 1-5](#_Toc529389321)

[1.14 .isspace 是否为空格，tab也是空格 1-5](#_Toc529389322)

[1.15 .istitle 判断首字母大写 1-5](#_Toc529389323)

[1.16 .isprintable 判断是否可打印 1-5](#_Toc529389324)

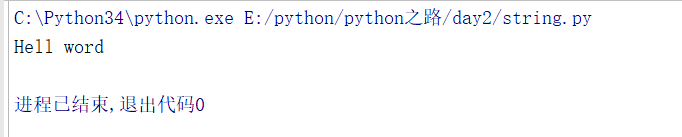
[1.17 .isupper 判断是否大写 1-5](#_Toc529389325)

[1.18 .join 添加分割符 1-5](#_Toc529389326)

# 字符串操作

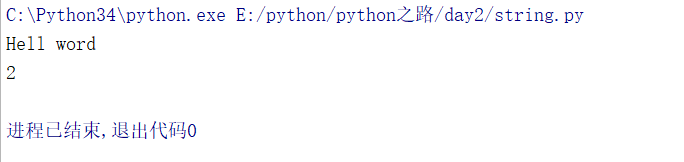
## .capitalize 第一个字符大写

name **=** "hell Word"  
print(name.capitalize())



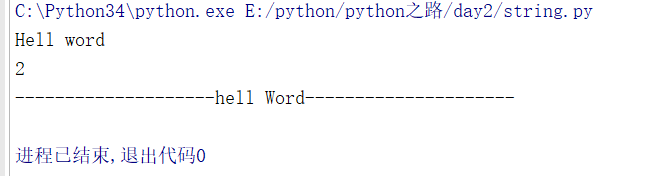
## .count 统计字符数量 统计某个字符出现的数量

name **=** "hell Word"  
print(name.capitalize())  
  
print(name.count("l"))

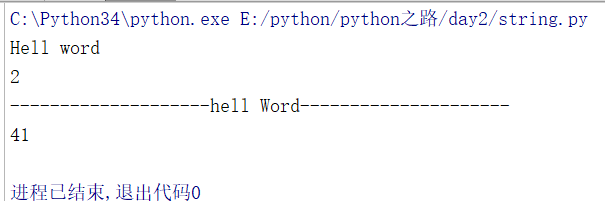


## .center 自动填充

name **=** "hell Word"  
print(name.capitalize())  
  
print(name.count("l"))  
  
print(name.center(50,"-")) #共50个字符，将 name 放在中间 不够左右填充-

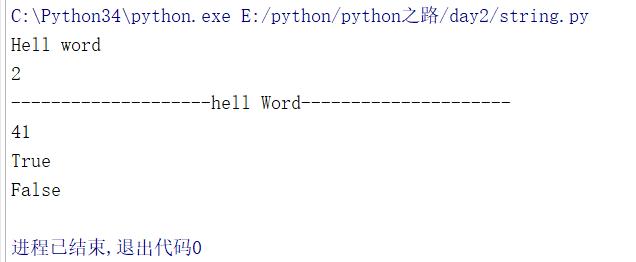


name **=** "hell Word"  
print(name.capitalize())  
  
print(name.count("l"))  
  
print(name.center(50,"-"))  
print(name.center(50,"-").count("-"))



## .endswith 判断是否以xx结尾

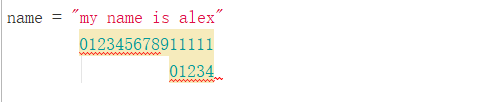
name **=** "hell Word"  
print(name.capitalize())  
  
print(name.count("l"))  
  
print(name.center(50,"-"))  
print(name.center(50,"-").count("-"))  
  
print(name.endswith("d")) #以d结尾返回True  
print(name.endswith("b"))



## .find 查找首个字符的索引

name **=** "my name is alex"

从0开始 ，空格也计入其中。



print(name.find("a"))

print(name.find("alex"))

4

11

应用：字符串切片

例如取出name （n-e之间的内容）

print(name[name.find("n")**:**name.find("e")])

nam

## .format 格式化输出

name **=** "my name is {name} , {year} old." #将变量中需要替换的内容{}

print(name.format(name**=**"alex",year**=**"33")) #print 时 输入值



## .isalnum 判断是否为阿拉伯字符

print('abe12dd'.isalnum()) #abe12dd 为阿拉伯字符,所以结果为True; 判断对象也可以为变量

True

## .isalpha 判断是否为纯英文字符

print('abc'.isalpha()) #同上

## .isdecimal 判断是否为十进制

print('12'.isdecimal())

## .isdigit 判断是否为整数

print('123'.isdigit())

## .isidentifier 判断是否为合法标识符（是不是正常变量名）

print('AC'.isidentifier())

## .islower 判断是否为小写

## .isnumeric 判断是否为数字，只能有数字

## .isspace 是否为空格，tab也是空格

## .istitle 判断首字母大写

print('My Name Is s'.istitle())

False

## .isprintable 判断是否可打印

设备文件，终端文件等不可打印为False 如：tty文件

## .isupper 判断是否大写

## .join 添加分割符

print('--'.join(['1','2','3']))



print('+'.join(['a','b','c','d']))



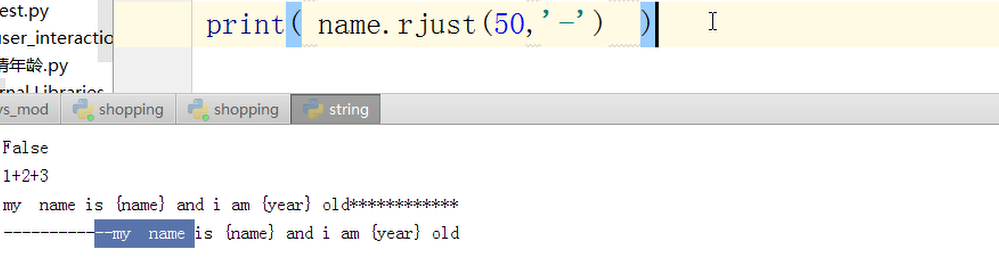
## .ljuest 向右填充

print(name.ljust(50,'\*'))



#不满足输出长度，在结尾补上

## rjuest 左填充



## lower upper 大小写转换

print("Name".lower()) *#转换为小写*print("Name".upper()) *#转换为大写*

## 回车空格处理strip

print("\n Name \n".lstrip()) *#去掉最左边的回车空格*print("\n Name \n".rstrip()) *#去掉最右边的回车空格*print("\n Name \n is \n".strip()) *#去掉两头的回车空格*print("-------")



## . maketrans 按照对应号码解释

p **=** str.maketrans("abcdef","123456") #定义的对照表必须长度一致  
  
print("af b".translate(p))



## replace 字符替换

print("My Name is ali ".replace("a","X"))

print("My Name is ali ".replace("a","X",1)) *#替换1个字符*

## rfind 从左往右0开始，找到最后字符串的值

print("My Name is is".rfind('i'))  
*# 0123456789*



## split 按照空格转换成列表

print("My Name is".split()) *#转换列表*



### 指定分割符

print("My Name is".split("a"))



## 强制大小写转换 swapcase

print("My Name is".swapcase())



## title 首字母大写

