

在python2.x中，单引号和双引号能混合使用，但是在python3.x中，单引号和双引号不能混合使用，需要使用转义字符

一，字符串介绍（字符串不能被改变）

1，字符串是一种表示文本的数据类型，字符串中的字符可以是ASCII字符，各种符号已经各种

Unicode字符

表示形式：①使用单引号包含字符‘s’，单引号表示字符串里不能包含单引号

②使用双引号包含字符“a”，双引号表示字符串里不能包含双引

号，并且只能有一行

③使用三引号（三对单引号或者是三对双引号）包含字符，在这个字符串中可以包含换

行符、制表符或者是其他字符，通常情况下三引号出现的字符出现在函数声明的下一

行，用来注释函数

2，转义字符（\n换行，\r回车，\f换页，\'双引号，\'单引号，\\反斜杠）

二，字符串的输入与输出：

（在python中输出汉字，需要在该python文件的开头声明`#-*- coding:utf-8 -*-`）

1，字符串输出：使用print函数，字符串在单引号或者双引号甚至三引号括起来，

如：`print("姓名：%s"%name)`

常见的格式化符号：

①%s：通过str（）字符串转换来格式化

②%i：有符号十进制整数

③%d：有符号十进制整数

④%o：八进制整数

⑤%x：十六进制整数（小写字母）

⑥%X：十六进制整数（大写字母）

⑦%e：索引符号（小写e）

⑧%E：索引符号（大写E）

⑨%f：浮点实数

①%g：%f和%e的简写

②%G：%f和%E的简写

2，字符串输入：

在python2.x中，输入格式为：

```
myst = raw_input("请输入指令")  
print(myst)
```

在python3.x中，输入的格式为：

```
myst = input("请输入指令")  
print(myst)
```

注意：row_input和input函数获取的数据，即使是数字，也是以字符串的方式进行保存的

3，访问字符串中的值：

python不支持单字符类型，单字符在python中也相当于一个字符串，可以通过下标访问字符串

串，字符串的下标从0开始

①获取index处的字符：str = string[index]

②字符串切片：对操作对象截取其中一部分字符（字符串，元组、列表都支持切片）

格式：[起始:结束:步长]，选取的区间是左闭右开型，包含起始位置，但是不包含结束位

置其中步长为正数表示从左边起，步长为负数从右边起

例如：print(name1[0:3])

print(name1[::-2])

4，字符串内建函数：

①：find函数用于检测字符串中是否包含子字符串sub，如果指定start和end范围，则检测是否包

含在指定范围内，如果包含子字符串在内则返回索引值，否则返回-1

语法格式：str.find(sub[, start[, end]])其中，sub指定检索的字符串，start开始索引，默认值

为0，end结束索引，默认为字符串的长度

例：string = "hello world, i love learning evering"

print(string.find("love"))

②：index函数用于检测字符串中是否包含子字符串sub，如果指定start和end范围，则检测是否

包含在指定范围内，如果包含子字符串在内则返回索引值，否则抛出异常

语法格式：str.index(sub[, start[, end]])其中，sub指定检索的字符串，start开始索引，默认值

为0，end结束索引，默认为字符串的长度

```
例: string = "hello world,i love learing evering"  
print(string.index("love",0,10))
```

③: count函数用于统计字符串里面某个字符出现的次数，可选参数为在字符串搜索的开始

于结束位置，该函数返回子字符串在字符串中出现的次数

语法格式: str.index(sub[, start[, end]]) 其中，sub指定检索的字符串，start开始索引，默认值

为0，end结束索引，默认为字符串的长度

```
例: string = "hello world,i love learing evering"  
print(string.count("o"))
```

④: replace函数把字符串中的old替换成new，该函数返回的是字符串中old替换成new后

生成的新字符串，如果指定第三个参数count，则表示替换次数不超过count次

语法格式:str.replace(old,new[, count]) 其中old表示将被替换的子字符串，new

表示新字符串，用于替换old字符串，count表示替换不超过count次，是

一个可选参数

```
例: string = "hello world,i love learing  
evering"  
print(string.replace("i","you"))
```

⑤: split函数通过指定分隔符对字符串进行切片，如果maxsplit有指定值，则仅分割

maxsplit个字符串，该函数的返回值是分割后的字符串列表

语法格式: str.split(sep = none,maxsplit = -1) 其中sep是分隔符，默认为所有

空字符，包括空格，换行，制表符等，maxsplit表示分割次数

```
例: string = "hello world,i love learing evering"  
print(string.split(","))
```

⑥: capitalize函数用于将字符串的第一个字母变成大写字母，其他字母变成小写字母，

返回一个数字字母为大写字母的字符串

语法格式为: `str.capitalize()`

例: `string = "hello world,i love learing evering"`

`print(string.capitalize())`

⑦: `title`函数返回“标题化”的字符串,也就是说所有单词都是以大写字母开头,其余

字母均为小写字母

语法格式为: `str.title()`

例: `string = "hello world,i love learing evering"`

`print(string.title())`

⑧: `startswith`函数用于检查字符串是否是以指定子字符串开头,如果是,返回`true`,如

果不是,返回`false`,如果指定`start`和`end`的值,则表示在指定范围内进行检查

语法格式: `str.startswith(prefix[, start[, end]])`其中`prefix`表示检测的字符

串, `start`表示参数用于设置字符串检测的起始位置。`end`表示用于设置

字符串检测的结束位置

例: `string = "hello world,i love learing evering"`

`print(string.startswith("hello"))`

⑨: `endswith`函数用于检查字符串是否是以指定子字符串结束,如果是,返回`true`,如

果不是,返回`false`,如果指定`start`和`end`的值,则表示在指定范围内进行检查

语法格式: `str.endswith(suffix[, start[, end]])`其中`suffix`表示检测的字符

串或者是一个元素, `start`表示参数用于设置字符串检测的起始位置。

`end`表示用于设置字符串检测的结束位置

例: `string = "hello world,i love learing evering"`

`print(string.endswith("hello"))`

⑩: upper函数将字符串中的小写字母转换为大写字母, 返回转换为大写字母后的字符串

语法格式: `str.upper()`

例: `string = "hello world,i love learing evering"`
`print(string.upper())`

①: 在python2.x中, lower函数将字符串中的大写字母转换为小写字母, 返回转换为小写

字母后的字符串

语法格式: `str.lower()`

例: `string = "hello world,i love learing evering"`
`print(string.lower())`

②: ljust函数返回一个原字符串左对齐, 并使用空格填充至指定长度的新字符串, 如果指

定长度小于原字符串的长度, 则返回原字符串

语法格式: `str.ljust(width[, fillchar])`

例: `string = "hello world,i love learing evering"`
`print(string.ljust(40, "_"))`

③: rjust函数返回一个原字符串左对齐, 并使用空格填充至指定长度的新字符串, 如果指

定长度小于原字符串的长度, 则返回原字符串

语法格式: `str.rjust(width[, fillchar])`

例: `string = "hello world,i love learing evering"`
`print(string.rjust(40, "_"))`

④: center函数返回一个宽度为width, 原字符串居中, 以fillchar (默认空格) 填充左右

的字符串

语法格式: `str.center(width[, fillchar])`

例: `string = "hello world,i love learing evering"`
`print(string.center(50))`

⑤: lstrip函数用于截掉字符串左边的空格或者指定字符, 返回一个新字符串

语法格式: `str.lstrip([chars])` 其中chars表示截取的字符

例: `string = "hello world,i love learing evering"`
`string2 = string.lstrip(60)`

```
print(string2.lstrip())
```

⑥: rstrip函数用于删除字符串末尾的指定字符（默认为空格），返回一个新字符串

语法格式: str.rstrip([chars]) 其中chars表示删除的字符

例: string = "hello world,i love learing evering"

```
string2 =string.center(60)
```

```
print(string2.rstrip())
```

⑦: strip函数用于移除字符串头尾的字符（默认为空格），返回一个新字符串

语法格式: str.strip([chars]) 其中chars表示移除字符串头尾指定的字符

例: string = "hello world,i love learing evering"

```
string2 =string.center(60)
```

```
print(string2.strip())
```

注意: ①在python2中, 普通字符串是以8位ASCII码进行存储的, 而Unicode字符串中的字符按

照Unicode存储, 这样能够表示更多的字符集, 使用的语法是在字符串前面加上前缀u

②在python3中, 所有的字符串都是Unicode字符串

5, 字符串运算:

+表示字符串连接; *表示重复输出字符串; []通过索引获取字符串中的字符; [:]截取字符串中的一部分; in成员运算符, 如果包含给定的字符, 则返回true; not in成员运算符, 如果不包含给定的字符串, 则返回true; r/R