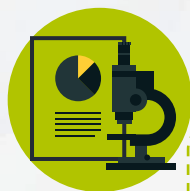


PYTHON

编程基础

字符串的其他常用操作

复制字符串



由于字符串是不可变类型，无法修改字符串中的某个元素值，不存在修改一个字符串值会影响另一个字符串的问题。

因此，直接用赋值运算符 “=” 实现字符串复制功能即可。



例：复制字符串示例。

```
str1='Java'
```

```
str2='C++'
```

```
str1='Python'
```

```
print('str1: %s, str2: %s'%(str1,str2))
```

```
str1: Python, str2: C++
```

连接字符串



- 作为一种序列数据，直接使用拼接运算（+）即可实现两个字符串的连接。
- 另外，还可以使用字符串中的join方法将序列中的元素以指定的字符连接成一个新的字符串，join方法的语法格式为：`str.join(seq)`
- 其中，seq是一个序列对象，str是使用的连接符。join方法返回连接后的字符串。



例：join方法使用示例。

```
str1=',' #仅包含一个逗号的字符串  
str2=' ' #仅包含一个空格的字符串  
str3="" #一个空字符串  
ls=['I','like','Python'] #列表  
print(str1.join(ls))  
print(str2.join(ls))  
print(str3.join(ls))
```



```
I,like,Python  
I like Python  
IlikePython
```

获取字符串长度



使用len函数可以计算一个字符串中包含的字符数量（即字符串长度），len函数的语法格式为：len(str)



例：获取字符串长度示例。

```
print('字符串 "Python" 的长度为：',len('Python'))  
print('字符串 "你好！" 的长度为：',len('你好！'))  
字符串 "Python" 的长度为： 6  
字符串 "你好！" 的长度为： 3
```

测试字符串的组成部分

01

如果需要判断一个字符串A是否是另一个字符串B的组成部分（即子串），可以直接使用前面介绍的字符串检索方法，检索成功，则A是B的子串，否则A不是B的子串。另外，也可以使用更简洁的in运算符。

02

例：测试字符串的组成部分示例。

```
str='cat dog cat'
print("'cat'是str的子串：",'cat' in str)
print("'mouse'是str的子串：",'mouse' in str)

'cat'是str的子串： True
'mouse'是str的子串： False
```