

Python拼接字符串的几种方式

1、%方式

```
print('%s %s' % ('Hello', 'world'))
>>> Hello world
```

%号格式化字符串的方式继承自古老的C语言，这在很多编程语言都有类似的实现。上例的%s是一个占位符，它仅代表一段字符串，并不是拼接的实际内容。实际的拼接内容在一个单独的%号后面，放在一个元组里。

类似的占位符还有：%d（代表一个整数）、%f（代表一个浮点数）、%x（代表一个16进制数），等等。

格式符号	转换
%c	字符
%s	字符串
%d	有符号十进制整数
%u	无符号十进制整数
%o	八进制整数
%x	十六进制整数（小写字母0x）
%X	十六进制整数（大写字母0X）
%f	浮点数
%e	科学计数法（小写“e”）
%E	科学计数法（大写“E”）
%g	%f和%e的简写
%G	%f和%E的简写

2、format()拼接方式

```

# 简洁版
s1 = 'Hello {}! My name is {}'.format('World', 'Python猫')
print(s1)
>>>Hello World! My name is Python猫.

# 对号入座版
s2 = 'Hello {0}! My name is {1}'.format('World', 'Python猫')
s3 = 'Hello {name1}! My name is {name2}'.format(name1='World', name2='Python猫')
print(s2)
>>>Hello World! My name is Python猫.
print(s3)
>>>Hello World! My name is Python猫.

```

这种方式使用花括号{}做占位符，在format方法中再转入实际的拼接值。容易看出，它实际上是对%号拼接方式的改进。

上例中，简洁版的花括号中无内容，缺点是容易弄错次序。对号入座版主要有两种，一种传入序列号，一种则使用key-value的方式。实战中，我们更推荐后一种，既不会数错次序，又更直观可读。

3、常用的+号方式

```

str_1 = 'Hello world! '
str_2 = 'My name is Python猫.'
print(str_1 + str_2)
>>>Hello world! My name is Python猫.
print(str_1)
>>>Hello world!

```