在许多编程语言中都包含有格式化字符串的功能，比如C和Fortran语言中的格式化输入输出。Python中内置有对字符串进行格式化的操作%。

**模板**

格式化字符串时，Python使用一个字符串作为模板。模板中有格式符，这些格式符为真实值预留位置，并说明真实数值应该呈现的格式。Python用一个tuple将多个值传递给模板，每个值对应一个格式符。

比如下面的例子：

print("I'm %s. I'm %d year old" % ('Vamei', 99))

上面的例子中，

"I'm %s. I'm %d year old" 为我们的模板。%s为第一个格式符，表示一个字符串。%d为第二个格式符，表示一个整数。('Vamei', 99)的两个元素'Vamei'和99为替换%s和%d的真实值。   
在模板和tuple之间，有一个%号分隔，它代表了格式化操作。

整个"I'm %s. I'm %d year old" % ('Vamei', 99) 实际上构成一个字符串表达式。我们可以像一个正常的字符串那样，将它赋值给某个变量。比如:

a = "I'm %s. I'm %d year old" % ('Vamei', 99)

print(a)

我们还可以用词典来传递真实值。如下：

print("I'm %(name)s. I'm %(age)d year old" % {'name':'Vamei', 'age':99})

可以看到，我们对两个格式符进行了命名。命名使用()括起来。每个命名对应词典的一个key。

**格式符**

格式符为真实值预留位置，并控制显示的格式。格式符可以包含有一个类型码，用以控制显示的类型，如下:

%s    字符串 (采用str()的显示)

%r    字符串 (采用repr()的显示)

%c    单个字符

%b    二进制整数

%d    十进制整数

%i    十进制整数

%o    八进制整数

%x    十六进制整数

%e    指数 (基底写为e)

%E    指数 (基底写为E)

%f    浮点数

%F    浮点数，与上相同

%g    指数(e)或浮点数 (根据显示长度)

%G    指数(E)或浮点数 (根据显示长度)

%%    字符"%"

可以用如下的方式，对格式进行进一步的控制：

%[(name)][flags][width].[precision]typecode

(name)为命名

flags可以有+,-,' '或0。+表示右对齐。-表示左对齐。' '为一个空格，表示在正数的左侧填充一个空格，从而与负数对齐。0表示使用0填充。

width表示显示宽度

precision表示小数点后精度

比如：

print("%+10x" % 10)

print("%04d" % 5)

print("%6.3f" % 2.3)

上面的width, precision为两个整数。我们可以利用\*，来动态代入这两个量。比如：

print("%.\*f" % (4, 1.2))

Python实际上用4来替换\*。所以实际的模板为"%.4f"。

**总结**

Python中内置的%操作符可用于格式化字符串操作，控制字符串的呈现格式。Python中还有其他的格式化字符串的方式，但%操作符的使用是最方便的。

原文出处：

<http://www.cnblogs.com/vamei/archive/2013/03/12/2954938.html>