Python 基础 (四): 字符串

目录

```
1 简介
2 基本操作
2.1 访问
2.2 单个字符编码
3 转义符
4 运算符
5 格式化
```

1 简介

在之前的文章【Python 基础(一): 入门必备知识】中我们已经提到了: 字符串是 Python 的一种数据类型,它可以通过单引号 '、双引号 "、三引号 '' 或 """ 来 定义,本节我们来详细介绍一下。

2 基本操作

2.1 访问

访问单个字符

```
1 | s = 'Python'
2 | # 访问第一个字符 P
3 | print(s[0])
```

访问范围内字符

```
1 s = 'Python'

2 # 访问 yt

3 print(s[1:3])

4 # 访问 Pyt

5 print(s[:3])

6 # 访问 hon

7 print(s[3:])
```

2.2 单个字符编码

Python 使用了 ord() 函数返回单个字符的编码, chr() 函数把编码转成相应字符。如下所示:

```
1 | s = 'A'
2 | print(ord(s))
3 | print(chr(65))
```

输出结果:

1 65

3 转义符

之前我们说过可以通过反斜杠 \ 将一行语句分多行显示,其实就是 \ 来转义字符,一些常见的转义字符如下表所示:

转义字符	描述
\	在行尾使用时,用作续行符
d/	退格 (Backspace)
\000	空
\n	换行
\v	纵向制表符
\t	横向制表符
\r	回车

4运算符

之前我们已经介绍了大部分运算符,下面再来详细看一下字符串运算符。如下表所示:

运算符	描述
+	连接符
*	重复输出
0	通过索引获取字符串中字符
[:]	获取字符串中的一部分
in	字符串中是否包含指定字符
not in	字符串中是否不包含指定字符
r/R	字符串原样输出

使用示例如下所示:

```
1  s1 = 'Hello'
2  s2 = 'Python'
3  print('s1 + s2 -->', s1 + s2)
4  print('s1 * 2 -->', s1 * 2)
5  print('s1[0] -->', s1[0])
6  print('s1[0:2] -->', s1[0:2])
7  print('"H" in s1 -->','H' in s1)
8  print('"H" not in s1 -->','H' not in s1)
9  print('\\r -->', R'\\r')
```

输出结果:

```
1 | s1 + s2 --> HelloPython

2 | s1 * 2 --> HelloHello

3 | s1[0] --> H

4 | s1[0:2] --> He

5 | "H" in s1 --> True

6 | "H" not in s1 --> False

7 | \r --> \r
```

5 格式化

当我们需要输出的内容中含有变量时,比如: Hello xxx , xxx 为变量,此时便需要一种格式化字符串的方式,Python 使用 % 格式化字符串,常用占位符如下表所示:

占位符	描述
%s	格式化字符串
%d	格式化整数
%f	格式化浮点数

以字符串为例,如下所示:

```
1 | print('Hello %s' % 'Python')
```

输出结果:

```
1 Hello Python
```

我们也可以使用字符串的 format() 方法进行格式化, 先看下示例:

```
1 | print('{0} {1}'.format('Hello', 'Python'))
```

这种方式是用传入的参数依次替换字符串内的占位符{0}、{1}...

