

基本语法

IlleeniumDillon

2024 年 6 月 2 日

1 文件编码

默认情况下，Python 源码文件的编码是 UTF-8。这种编码支持世界上大多数语言的字符，可以用于字符串面值、变量、函数名及注释。如果不使用默认编码，则要声明文件的编码，文件的第一行要写成特殊注释。句法如下：其中 encoding 是文件的编码，常见的编码有 utf-8、gbk、

```
# -*- coding: encoding -*-
```

gb2312 等。具体可见 Python 文档。具体地，如果文件的编码是 utf-8，则第一行应该写成：第一行的规则也有一种例外情况，源码以 UnixShell 脚

```
# -*- coding: utf-8 -*-
```

本开始，前两行可以写成：那么第一行的作用是告诉操作系统，这个脚本

```
#!/usr/bin/env python3  
# -*- coding: utf-8 -*-
```

用 Python3 解释器执行，并指出解释器的路径。在类 Unix 系统中，可以直接运行这个脚本，但是在 Windows 系统中，这两行会被忽略。

2 注释

Python 的注释以 `#` 开头，直到行尾结束。注释可以单独占一行，也可以跟在语句后面。注释用于解释代码的功能，提高代码的可读性。注释可以是单行注释，也可以是多行注释。单行注释以 `#` 开头，多行注释以三个单引号或三个双引号开始和结束。多行注释可以用于函数的文档字符串，也可以用于多行注释。

```
# 这是一个单行注释
print("Hello, World!") # 这也是一个单行注释
'''
这是一个多行注释
这是一个多行注释
这是一个多行注释
'''
```

3 缩进

Python 使用缩进来表示代码块，缩进是 Python 语法的一部分。缩进的空格数是可变的，但是同一个代码块的语句必须包含相同的缩进空格数。一般情况下，缩进使用 4 个空格，也可以使用 2 个空格或者 8 个空格。缩进的空格数不是固定的，但是同一个代码块的语句必须使用相同的缩进空格数。缩进的空格数不是固定的，但是同一个代码块的语句必须使用相同的缩进空格数。缩进的空格数不是固定的，但是同一个代码块的语句必须使用相同的缩进空格数。缩进的空格数不是固定的，但是同一个代码块的语句必须使用相同的缩进空格数。

4 行尾分号

Python 语句不需要使用分号结尾，但是如果一行中有多个语句，可以使用分号分隔。分号用于分隔同一行的多个语句，但是不推荐在同一行中

```
if True:
    print("True")
else:
    print("False")
```

写多个语句。如果一行中有多个语句，可以使用分号分隔，但是不推荐这样做。

```
print("Hello, World!"); print("Hello, Python!")
```

5 多行语句

Python 语句通常以新行开始，但是可以使用反斜杠 来实现多行语句。反斜杠 可以用于将一行的语句分成多行显示，但是不推荐使用反斜杠 来实现多行语句。如果一行的语句太长，可以使用反斜杠 将其分成多行显示。

```
total = 1 + 2 + 3 + \
        4 + 5 + 6 + \
        7 + 8 + 9
```

6 引号

Python 可以使用单引号、双引号和三引号来表示字符串。单引号和双引号的作用是相同的，三引号用于表示多行字符串。单引号和双引号可以用于表示字符串，三引号用于表示多行字符串。单引号和双引号的作用是相同的，三引号用于表示多行字符串。单引号和双引号的作用是相同的，三引号用于表示多行字符串。

```
print('Hello, World!')
print("Hello, Python!")
print('''Hello, World!
Hello, Python!''')
```
