

附录—缩进和注释

2023 年 9 月 18 日

附录——缩进和注释

1. 缩进

Python 是一种计算机编程语言，它和我们日常使用的语言，有很多相似之处。它有语法和语句块，对应我们自然语言的主谓宾，段落等概念。中文的书面语，每段开头的规则是空两格；而 Python 则是语句块第二行空 4 格，这就是“缩进”。

```
# print absolute value of an integer:  
a = 100  
if a >= 0:  
    print(a)  
else:  
    print(-a)
```

缩进有利有弊。好处是强迫你写出格式化的代码，但没有规定缩进是几个空格还是 Tab。按照约定俗成的惯例，应该始终坚持使用 4 个空格的缩进。



Python 的缩进规则如下：- 使用空格或制表符进行缩进，但不要混合使用两者。- 推荐使用四个空格进行缩进，这也是官方推荐的缩进风格。- 每一级缩进都应该增加一个缩进层次。- 缩进的空格数必须在同一个代码块中保持一致。

下面是一些示例代码，展示了 Python 中缩进的使用：

```
if x > 0:
    print("x 是正数")
    print(" 这是 if 代码块的一部分")
else:
    print("x 是负数或零")
    print(" 这是 else 代码块的一部分")
print(" 这是 if-else 语句之外的代码")

def greet(name):
    print("Hello, " + name)
    print(" 欢迎使用 Python")
```

在上面的示例中，`if` 语句和 `else` 语句的代码块都使用了缩进，它们是根据条件执行的不同代码分支。同样，函数 `greet` 的定义也使用了缩进。

请注意，Python 对缩进非常敏感，正确的缩进是保证代码能够正常运行的关键。如果缩进不正确，可能会导致语法错误或逻辑错误。

2. 注释

在 Python 中，注释是用来解释代码的文本，它们对于理解代码和提供说明非常重要。Python 支持两种类型的注释：

1. 单行注释：以井号符号“#”开头，后面跟着注释内容。单行注释只在同一行有效，用于解释该行代码的作用或提供相关说明。例如：

```
# 这是一个单行注释，用于解释下面这行代码的作用
x = 10 # 将值 10 赋给变量 x
```

2. 多行注释：以三个连续的引号（单引号或双引号）开始和结束，可以跨越多行。多行注释通常用于提供更详细的解释、函数或类的文档字符串（docstring）等。例如：

```
"""
这是一个多行注释的示例，可以用来提供更详细的代码解释和说明。
"""
```

这里可以写多行的注释内容，可以跨越多行。

```
"""
```

多行注释也可以使用单引号来表示，例如：''' 注释内容'''。

请注意，注释不会被解释器执行，它们只是用来提供说明和解释代码的作用。编写清晰、明确的注释可以帮助他人理解您的代码，并使代码更易于维护和修改。