Python不需要将源代码编译成二进制码，你只需要直接从源代码运行该程序。在程序内部，Python会将源代码转换为称为字节码的中间形式，尔后再转换成你的电脑所使用的语言，并运行它。

Python同时支持面向过程编程和面向对象编程。

***可扩展性***

如果希望代码的某一部分快速的运行或者希望算法的某些部分不被公开，你可以在C或C++语言中编写这些程序，然后再将其运用到你的Python程序中。

***可嵌入性***

你可以在你的C或C++程序中嵌入Python，从而向你的程序用户提供脚本功能。

***注释***

使用#进行单行注释

***字面常量***

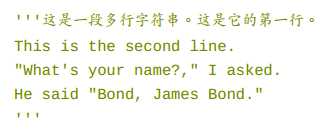
诸如，5、1.23这样的数字，或者是如“这是一串文本”或“This is a string”这样的文本。用这样的称呼是因为它们是字面上的------你用的就是它字面意义上的值或者内容。

***单引号、双引号***

被单引号包括的字符串和被双引号包含的字符串是一样的。

三引号

你可以通过使用三个引号------“””或’’’来指定多行字符串。也可以在三引号之间自由的使用单引号和多引号，比如，



***字符串是不可变的***

一旦创建了一个字符串，就不可能再改变它。

***格式化方法***

使用format()方法来格式化字符串

age = 20  
name = "Paul"  
print("{0} is {1} years old." . format(name, age))

输出

Hello, World!

Paul is 20 years old.

这里要注意，“{}”中的数字是可选项，去掉之后仍然会得到相同的输出结果。

还有更详细的使用方法



print()方法会默认的在最后使用“\n”来进行换行，可以使用end指定其应该以空白结尾。

print(‘a’, end=’’)

print(‘b’, end=’’)

输出结果为ab

或者通过end指定以空格结尾，

print(‘a’, end=’ ’)

print(‘b’, end=’ ’)

print(‘c’, end=’ ’)

输出结果为a b c

***转义序列***

使用反斜杠“\”进行转义。在一个字符串中，一个放置在末尾的反斜杠表示字符串将在下一行继续，例如，

“This is the first sentence.\

This is the second sentence.”

输出结果如下，

“This is the first sentence. This is the second sentence.”