

## 1.4 设计函数

::: details INFO

译者: [Mancuoj](#)

来源: [1.4 Designing Functions](#)

对应: Lab 01

:::

函数是所有程序（无论大小）的基本组成部分，并且是我们使用编程语言来表达计算过程的主要媒介。之前我们已经讨论过了函数的形式及其调用方式，而本节我们将讨论“什么是一个好函数”。从根本上说，好函数共有的品质就是：它们都强化了“函数就是抽象”的理念。

- 每个函数应该只负责一个任务，且该任务要用一个简短的名称来识别，并在一行文本中进行描述。按顺序执行多个任务的函数应该分为多个函数。
- 不要重复自己（Don't repeat yourself）是软件工程的核心原则。这个所谓的 DRY 原则指出，多个代码片段不应该描述重复的逻辑。相反，逻辑应该只实现一次，为其指定一个名称后多次使用。如果你发现自己正在复制粘贴一段代码，那么你可能已经找到了进行函数抽象的机会。
- 定义通用的函数。比如作为 `pow` 函数的一个特例的平方函数就不在 Python 库中，因为 `pow` 函数可以将数字计算为任意次方。

这些准则提高了代码的可读性，减少了错误的数量，并且通常最大限度地减少了编写的代码总量。将复杂的任务分解为简洁的功能是一项需要经验才能掌握的技能。幸运的是，Python 提供了多种特性来支持你的工作。

### 1.4.1 文档

函数定义通常包括描述函数的文档，称为“文档字符串 docstring”，它必须在函数体中缩进。文档字符串通常使用三个引号，第一行描述函数的任务，随后的几行可以描述参数并解释函数的意图：

```
>>> def pressure(v, t, n):
    """计算理想气体的压力，单位为帕斯卡

    使用理想气体定律: http://en.wikipedia.org/wiki/Ideal\_gas\_law

    v -- 气体体积，单位为立方米
    t -- 绝对温度，单位为开尔文
    n -- 气体粒子
    """
    k = 1.38e-23 # 玻尔兹曼常数
    return n * k * t / v
```

当你使用函数名称作为参数调用 `help` 时，你会看到它的文档字符串（键入 `q` 以退出 Python help）。

```
>>> help(pressure)
```

编写 Python 程序时，除了最简单的函数之外，都要包含文档字符串。要记住，虽然代码只编写一次，但是会在之后阅读多次。Python 文档包含了 [文档字符串准则](#)，它会在不同的 Python 项目中保持一致。

注释：Python 中的注释可以附加到 `#` 号后的行尾。例如，上面代码中的注释 `玻尔兹曼常数` 描述了 `k` 变量的含义。这些注释不会出现在 Python 的 `help` 中，而且会被解释器忽略，它们只为人类而存在。

## 1.4.2 参数默认值

定义通用函数的结果是引入了额外的参数。具有许多参数的函数可能调用起来很麻烦并且难以阅读。

在 Python 中，我们可以为函数的参数提供默认值。当调用该函数时，具有默认值的参数是可选的。如果未提供，则将默认值绑定到形参上。例如，如果程序通常用于计算“一摩尔”粒子的压力，则可以提供此值作为默认值：

```
>>> def pressure(v, t, n=6.022e23):  
    """计算理想气体的压力，单位为帕斯卡  
  
    使用理想气体定律: http://en.wikipedia.org/wiki/Ideal\_gas\_law  
  
    v -- 气体体积，单位为立方米  
    t -- 绝对温度，单位为开尔文  
    n -- 气体粒子，默认为一摩尔  
    """  
  
    k = 1.38e-23 # 玻尔兹曼常数  
    return n * k * t / v
```

= 符号在此示例中表示两种不同的含义，具体取决于使用它的上下文。在 `def` 语句中，= 不执行赋值，而是指示调用 `pressure` 函数时使用的默认值。相比之下，函数体中对 `k` 的赋值语句中将名称 `k` 与玻尔兹曼常数的近似值进行了绑定。

```
>>> pressure(1, 273.15)  
2269.974834  
>>> pressure(1, 273.15, 3 * 6.022e23)  
6809.924502
```

`pressure` 函数的定义接收三个参数，但上面的第一个调用表达式中只提供了两个。在这种情况下，`n` 的值取自 `def` 语句中的默认值。如果提供了第三个参数，默认值将被忽略。

作为准则，函数主体中使用的大多数数据值都应该表示为具名参数（named arguments）的默认值，这样会使它们更易于检查，并且可以被函数调用者更改。一些永远不会改变的值，例如基本常量 `k` 可以绑定在函数体或全局帧中。