

```
var k kelvin = 294.0
var c celsius

c = kelvinToCelsius(k)
c = k.celsius()
```

调用 kelvinToCelsius 函数

调用 celsius 方法

跟调用其他包中的函数一样，调用方法也需要用到点记号。以上面的代码为例，在调用方法的时候，程序首先需要给出正确类型的变量，接着是一个点号，最后才是被调用方法的名字。

既然温度转换操作现在已经是 kelvin 类型的方法，那么继续使用 kelvinToCelsius 这样的名字就没有必要了。在同一个包里面，如果一个名字已经被函数占用了，那么这个包就无法再定义同名的类型，因此在使用函数的情况下，我们将无法使用 celsius 函数返回 celsius 类型的值。然而，如果我们使用的是方法，那么每种温度类型都可以具有自己的 celsius 方法，就像以下展示的 fahrenheit 类型一样：

```
type fahrenheit float64

// celsius 方法会将华氏度转换为摄氏度
func (f fahrenheit) celsius() celsius {
    return celsius((f - 32.0) * 5.0 / 9.0)
}
```

通过让每种温度类型都具有相应的 celsius 方法以转换为摄氏温度，我们可以创造出一种完美的对称。

### 速查 13-3

请标识出这个方法声明中的接收者：func (f fahrenheit) celsius() celsius。



## 13.4 小结

- 使用自定义类型能够让代码变得更易读且更可靠。
- 方法就像是跟特定类型相关联的函数，其中被关联的类型将通过方法名前面的接收者来指定。方法跟函数一样，都可以接受多个形参并返回多个值，但是一个方法必须并且只能有一个接收者。在方法体内部，接收者的行为就跟其他形参一样。
- 调用方法需要用到点记号，排在最前面的是适当类型的变量，之后是一个点号和方

### 速查 13-3 答案

方法声明 func (f fahrenheit) celsius() celsius 中的接收者为 f，它的类型为 fahrenheit。