

只要知道它们是同一种东西就可以了，不用在这个地方产生疑惑。

函数是用来运行代码的载体，你可以在一个函数里编写很多行代码，当运行这个函数时，函数中的所有代码会全部运行。像我们前面使用过的 `main()` 函数就是一个函数，只不过它比较特殊，是程序的入口函数，即程序一旦运行，就是从 `main()` 函数开始执行的。

但是只有一个 `main()` 函数的程序显然是很初级的，和其他编程语言一样，Kotlin 也允许我们自由地定义函数，语法规则如下：

```
fun methodName(param1: Int, param2: Int): Int {
    return 0
}
```

下面我来解释一下上述的语法规则，首先 `fun`（function 的简写）是定义函数的关键字，无论你定义什么函数，都一定要使用 `fun` 来声明。

紧跟在 `fun` 后面的是函数名，这个就没有什么要求了，你可以根据自己的喜好起任何名字，但是良好的编程习惯是函数名最好要有一定的意义，能表达这个函数的作用是什么。

函数名后面紧跟着一对括号，里面可以声明该函数接收什么参数，参数的数量可以是任意多个，例如上述示例就表示该函数接收两个 `Int` 类型的参数。参数的声明格式是“参数名：参数类型”，其中参数名也是可以随便定义的，这一点和函数名类似。如果不想接收任何参数，那么写一对空括号就可以了。

参数括号后面的那部分是可选的，用于声明该函数会返回什么类型的数据，上述示例就表示该函数会返回一个 `Int` 类型的数据。如果你的函数不需要返回任何数据，这部分可以直接不写。

最后两个大括号之间的内容就是函数体了，我们可以在这里编写一个函数的具体逻辑。由于上述示例中声明了该函数会返回一个 `Int` 类型的数据，因此在函数体中我们简单地返回了一个 0。

这就是定义一个函数最标准的方式了，虽然 Kotlin 中还有许多其他修饰函数的关键字，但是只要掌握了上述函数定义规则，你就已经能应对 80% 以上的编程场景了，至于其他的关键字，我们会在后面慢慢学习。

接下来我们尝试按照上述定义函数的语法规则来定义一个有意义的函数，如下所示：

```
fun largerNumber(num1: Int, num2: Int): Int {
    return max(num1, num2)
}
```

这里定义了一个名叫 `largerNumber()` 的函数，该函数的作用很简单，接收两个整型参数，然后总是返回两个参数中更大的那个数。

注意，上述代码中使用了一个 `max()` 函数，这是 Kotlin 提供的一个内置函数，它的作用就是返回两个参数中更大的那个数，因此我们的 `largerNumber()` 函数其实就是对 `max()` 函数做了一层封装而已。