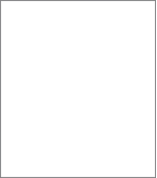
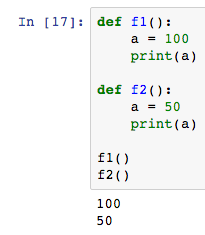
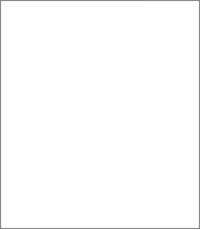
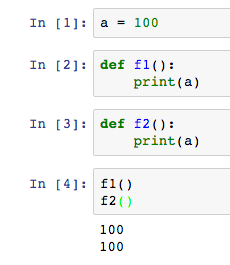
在这一课的内容中，我们将一起来学习variable scope，也就是变量的作用域。

听起来这个名词有点唬人，其实variable scope某种意义上来说就是变量的生命周期。和现实世界的生物一样，代码世界中的生命也有自己的生命周期。作为代码世界的重要组成部分，变量也有着自己特定的生命周期。

在Python中，变量的作用域（生命周期）有两种类型，分别是global和local。从单词的意思也可以看出来，global变量的作用域是整个代码，而local的作用域仅限于某一段代码。



为了让大家清楚这些概念，最简单的方式还是用代码来说明。

打开Jupyter Notebook，创建一个新的项目，并将其命名为scope。

接下来创建以下代码：

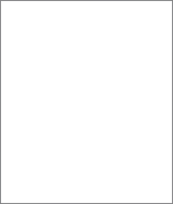
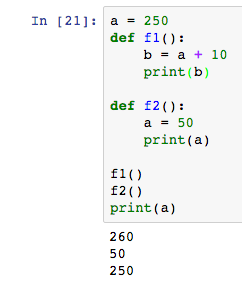
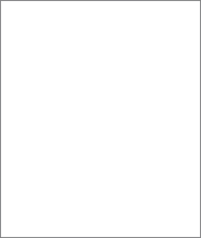
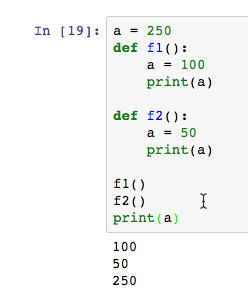
以上代码可以顺利运行。这是因为我们创建的a变量是一个global全局变量，所以在函数f1和f2中仍然可以使用。

好了，让我们换一种方式，再看看：

在以上的代码中，在f1和f2中都用到了变量a，但是因为在Python函数中的变量属于local局部变量，所以输出的结果仍然不同。

好了，如果我们再换一种方式：

在上面的代码中，函数之外的a是全局变量，而f1和f2中的a则是局部变量。在Python中，当我们在函数中使用和全局变量相同名称的变量时，会自动创建一个临时的局部变量。而这个局部变量的值不会影响全局变量的值。

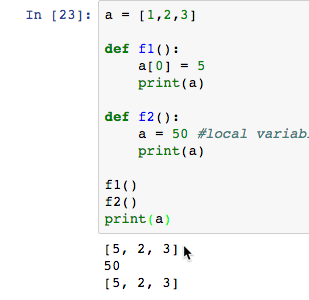
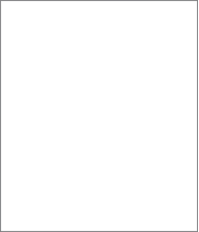
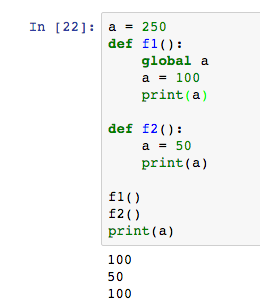


所以最后输出的结果是作为全局变量的a仍然是250。

接下来继续看一个例子：

在上面的例子中，在函数f1中并没有提供a的初始值，但是因为作为全局变量的a已经有初始值了，所以会直接调用250这个数值。

在某些情况下，我们可能希望用局部变量来overwrite全局变量的值，此时需要使用global关键词：



在以上的代码中，在f1函数中，我们使用global关键词，从而让其中的a也成为全局变量。注意f2函数中的仍然是局部变量。

再来看一个例子：

在上面的例子中，我们定义了一个列表类型的全局变量，然后在f1函数中更改了列表中的第一个元素的值。而f2函数中的a则是一个普通数值类型的局部变量。

在最后的输出结果中，全局变量的值也已经发生了变化。

需要注意的是，在f1中我们并没有使用global关键词，这是因为对于列表和dictionary来说，无需使用global关键词。

简单总结一下：

1.Python中存在两种类型的变量作用域-global和local

2.对于List和dictionary,可以无需使用global关键词，就在函数中更改其中元素的值。

好了，这一课的内容就到此结束了，我们下一课再见~