欢迎继续我们的学习。

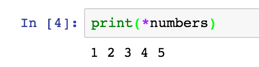
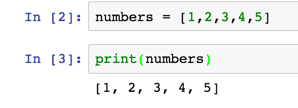
在本课的内容中，我们将继续关注Python中的函数，特别是如何packing和unpacking参数。

看到两个新的名词害怕了？Don’t panic，让我们用代码解释一切。

打开Jupyter Notebook，打开之前所创建的myfunc文件。

输入以下代码并运行：

这里，我们使用print函数，并传递了5个参数。这个没什么，很简单。



接着我们定义一个列表，然后使用print函数输出：

看起来似乎也很正常。

接下来我们换一种方式：

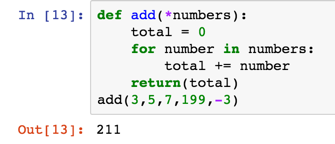
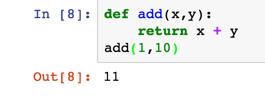
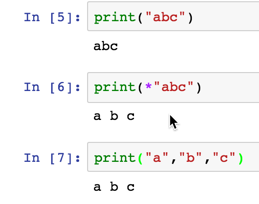
通过在列表变量的前面添加一个星号，我们就完成了unpacking的工作。此时输出的结果就不再是一个列表。

理论上来说，对于所有可遍历的变量类型，都可以使用unpacking的方式，将其变成单独的多个变量。

因为字符串属于可遍历的变量类型，所以也可以使用unpacking：

在上面的代码中，第一行print函数完整的输出了字符串，而第二行代码中则使用了unpacking，所以输出的是单个字母，其结果和第三行代码是一样的。

再看一个例子。



之前我们曾经定义过一个简单的加法函数：

这个函数运行起来没有任何问题，但遗憾的是它只支持两个数字相加，如果我们想要实现多个数字的相加，这个函数就无能为力了。

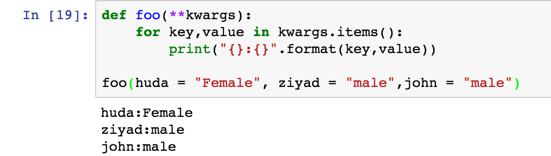
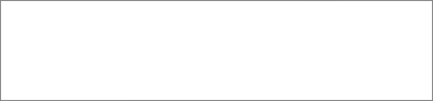
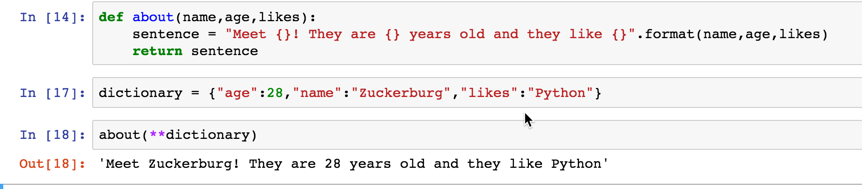
但是通过使用unpacking，我们可以轻松实现这个目的：

实际上，这个函数支持无穷多个数字的相加，比如最开始的add函数，有了质的飞跃。

在以上的代码中，\*号之后的numbers其实是一个turple。我们遍历了numbers中的每一个数字，然后将其添加到最终的结果中。

让我们再看一个例子：

在上一课的内容中我们创建过这个函数。



唯一不同的是，这里我们使用两个\*号来unpacking了关键词参数。

所以要记住，一个\*号用来unpacking普通的参数，两个\*号用来unpacking关键词参数。

再来看一个例子：

好了，这一课的内容就到此结束了，我们下一课再见~