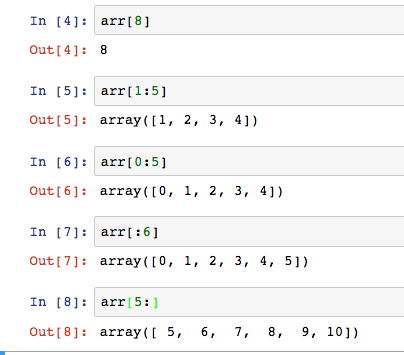
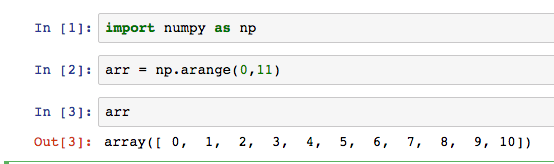
欢迎继续我们的学习。

在这一课的内容中，我们将重点了解Numpy的index和元素选择。

首先打开Anaconda和Jupyter Notebook，创建一个新的notebook，将其命名为NumpyIndexingAndSelection。



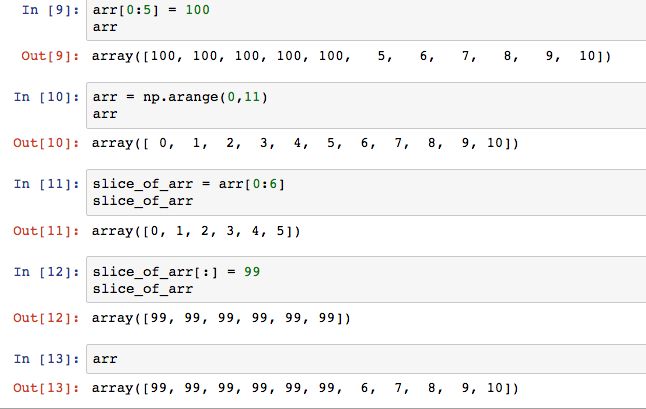
好了，现在可以开始我们的表演了~

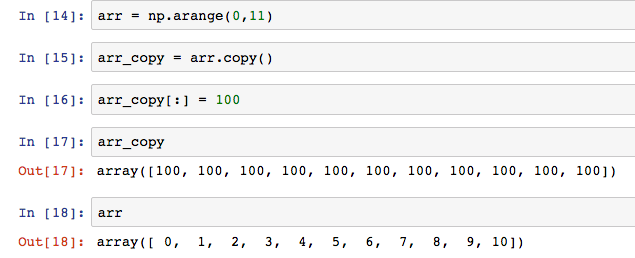
首先来一波常规操作，

在以上代码中，我们使用numpy中的arange函数创建了一个从0到10的一维数组。

接下来就是用到index的时候了，一维数组的index和列表有点类似。

在以上代码中，Numpy array的index操作跟python的普通列表的index操作几乎是一样的，这里就不再一一解释了。如果大家对这些代码还有疑问，不妨回过头复习下python列表的index操作。

接下来要说的是numpy array跟列表不一样的地方：



在以上代码中，我们使用index的方式将特定元素的值更改为指定的数值。

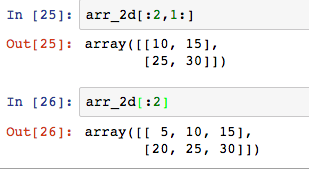
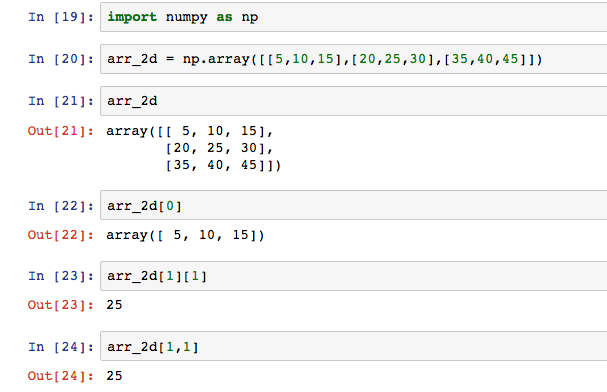
需要注意的是，在这种情况下，初始的数组也被改变了。

如果不希望初始的数组被改变，就可以考虑使用copy函数：

好了，刚才我们了解的都是关于一维数组的indexing，那么二维数组又是怎样的呢？

在以上代码中，我们创建了一个二维数组。

如果只使用单个index，那么获得的是对应的行。如果想获得某个具体的元素，就需要使用两个方括号，或者是在方括号中用逗号隔开的方式：



array[rowIndexNumber][columnIndexNumber]

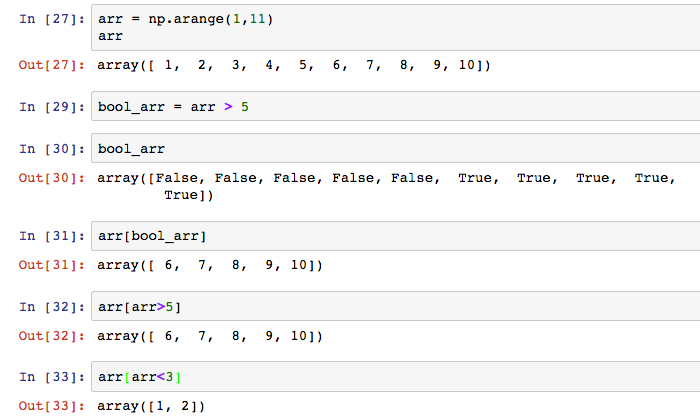
array[rowIndexNumber,columnIndexNumber]

接下来看一下相对复杂的操作：

在以上代码中，需要注意的是，如果没有逗号，那么index的方式和列表类似，只是此时提取的是行，而不是单个的元素。

如果有逗号，那么逗号之前的index是行的index，而逗号之后的index则是列的index。

接下来看看numpy中数组的条件选择，简单来说，也就是只选择满足条件的元素：



在以上的代码中，我们定义了一个布尔数组，从而在需要的时候只提取满足条件的元素（使得布尔值为True）。

最后两行代码的作用跟arr[bool\_arr]类似，只是为了让大家看的更清楚明白而已。

好了，这一课的内容就先到这里，我们下一课再见~