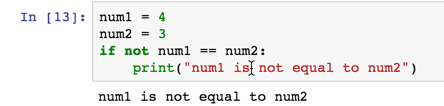
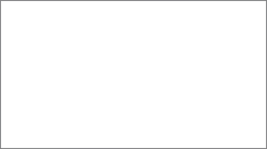
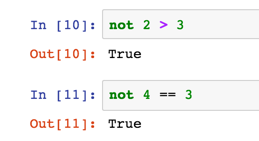
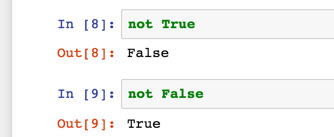
在上一课的内容中，我们学习了Python中逻辑判断语句if的基本用法。

在这一课的内容中，我们将学习基本的逻辑操作符，也就是之前提到的布尔代数中的and, or, not。

首先来看not，大家都知道英文中not的意思是否定，所以not True就代表False，而not False就代表True。



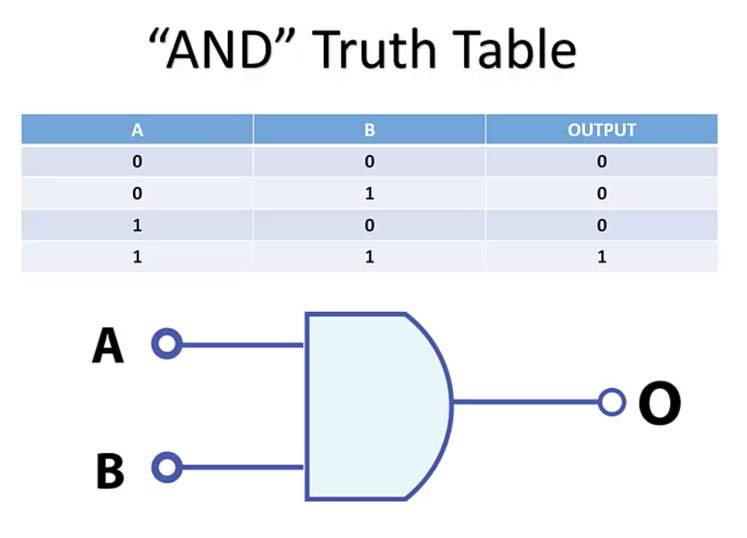
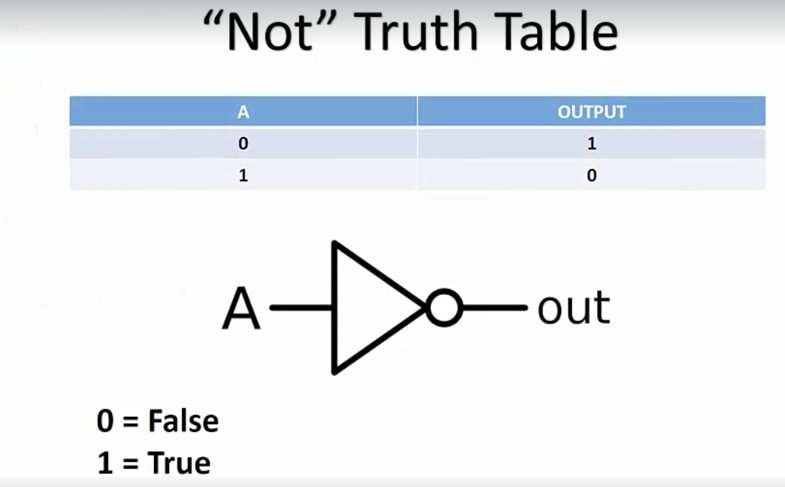
我们可以在if语句中组合使用not和条件判断，比如：

对于逻辑操作符来说，存在着所谓的真值表。当然，不要被这个名词吓倒了，大家只需要知道这个表的作用是给定某个输入，输出特定的结果。

对于not这个逻辑运算符来说，真值表非常简单。

接下来我们看看稍微复杂一点的逻辑运算符and，也就是“与”。

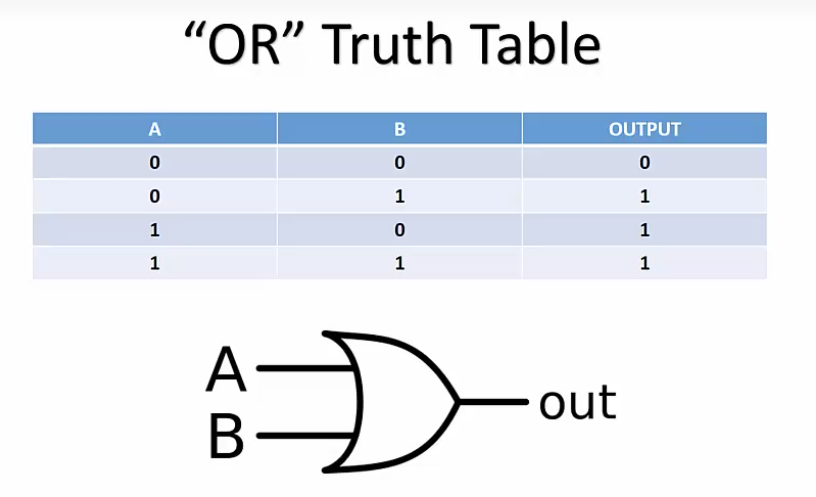
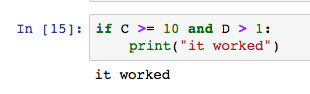
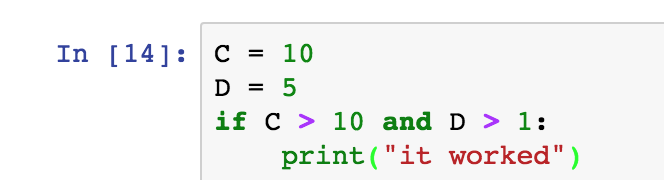
对于and的真值表，当且仅当两个输入的值都为1（True）时，才会输出1（True），其它情况下输出的结果都是0。



这么说可能有点抽象，我们还是用代码来展示比较直观。

在上面的代码中，我们定义了两个变量C和D，仅当C >10 和D >1两个条件同时满足时，才会输出对应的结果。

但是实际上仅满足一个条件，所以最终没有任何输出结果。



以上代码只需要稍微修改一下就OK了。

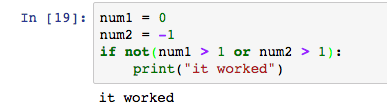
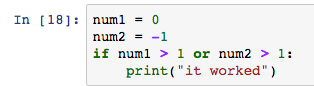
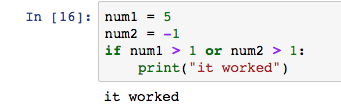
在了解了and的用法之后，最后我们再来了解一下另外一个逻辑操作符or，也就是或。

首先来看一下or的所谓真值表如下：

可以看到，or的意思是，当输入条件中只要有一个为1(True）时，输出的结果就是1(True）。

我们还是用实际的例子来说明。

在上面的代码中，我们定义了两个数字变量，然后使用if条件语句，以及or逻辑运算符进行条件判断，只要满足num1 >1或者num2 >1中的任一个条件（或者同时满足），就执行下面的语句。



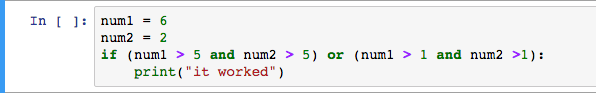
在上面的代码中，因为num1 > 1和num2 >1两个条件均不满足，所以就不会执行下面的语句。

好了，我们还可以进一步把not逻辑运算符结合使用：

小练习：

猜一猜下面代码的运算结果是怎样的，是否会输出it worked。

所以，只要我们能充分利用not,and,or这几个逻辑运算符，以及条件判断语句，就可以给Python生命体无穷的可能性。



好了，本课的内容就到此结束了，让我们期待后续的精彩旅程~