Exercise 30: Else and If

In the last exercise you worked out some `if-statement` and then tried to guess what they are and how they work. Before you learn more I'll explain what everything is by answering the questions you had from Study Drills. You did the Study Drills right?

1. What do you think the `if` does to the code under it? An `if-statement` creates what is called a "branch" in the code. It's kind of like those choose your own adventure books where you are asked to turn to one page if you make one choice, and another if you go a different direction. The `if-statement` tells your script, "If this boolean expression is True, then run the code under it, otherwise skip it."
2. Why does the code under the `if` need to be indented four spaces? A colon at the end of a line is how you tell Python you are going to create a new "block" of code, and then indenting four spaces tells Python what lines of code are in that block. This is *exactly* the same thing you did when you made functions in the first half of the book.
3. What happens if it isn't indented? If it isn't indented, you will most likely create a Python error. Python expects you to indent *something* after you end a line with a `:` (colon).
4. Can you put other boolean expressions from Ex. 27 in the `if-statement`? Try it. Yes you can, and they can be as complex as you like, although really complex things generally are bad style.
5. What happens if you change the initial values for `people`, `cats`, and `dogs`? Because you are comparing numbers, if you change the numbers, different `if-statements` will evaluate to `True` and the blocks of code under them will run. Go back and put different numbers in and see if you can figure out in your head what blocks of code will run.

Compare my answers to your answers, and make sure you *really* understand the concept of a "block" of code. This is important for when you do the next exercise where you write all the parts of `if-statements` that you can use.

Type this one in and make it work too.

```
1  people = 30
2  cars = 40
3  buses = 15
4
5
6  if cars > people:
7      print "We should take the cars."
8  elif cars < people:
9      print "We should not take the cars."
10 else:
11     print "We can't decide."
12
13 if buses > cars:
14     print "That's too many buses."
15 elif buses < cars:
16     print "Maybe we could take the buses."
17 else:
18     print "We still can't decide."
19
20 if people > buses:
21     print "Alright, let's just take the buses."
22 else:
23     print "Fine, let's stay home then."
```

What You Should See

```
$ python ex30.py
We should take the cars.
Maybe we could take the buses.
Alright, let's just take the buses.
```

Study Drills

1. Try to guess what `elif` and `else` are doing.
2. Change the numbers of `cars`, `people`, and `buses` and then trace through each `if-statement` to see what will be printed.

3. Try some more complex boolean expressions like `cars > people and buses < cars`.
4. Above each line write an English description of what the line does.

Common Student Questions

What happens if multiple `elif` blocks are `True`?

Python starts and the top at runs the first block that is `True`, so it will run only the first one.