Crash Course in Python Session 2
Shmuel Jacobs
May 2, 2018

Last week, we dealt with variables of type int, float and string. We also wrote lines of code that run the same way, every single time. However, we need our code to make decisions on its own. Let's look at how Boolean variables can do that for us.

Boolean Variables

Open IDLE to a new python shell. Try typing some equations that are either true or false.

```
>> 2 + 2 == 5

False

>> 2 + 2 == 4

True

>> 2 + 2 < 5

True

>> 2 + 2 <= 5

True

>> 2 + 2 <= 4

True

>> 2 + 2 <= 3

False

>> 2 + 2 != 3

True
```

Remember to use the double equals '==' for true/false statements. The double equals is the 'checker' equals, while the single equals is the 'setter' equals, used to instruct the computer to set a variable. Using the wrong one ***will*** cause some problem.

```
>> 2 + 2 == 5

SyntaxError: can't assign to operator
```

Variables aren't just for storing numbers. We can use one to store True/False values. This is called Boolean data.

```
# Setup for next example
```

Crash Course in Python Session 2
Shmuel Jacobs
May 2, 2018

```
>> x = 2

>> y = 2

>> z = 5
```

Consider this example. I have integers stored in variables x and y, which we in a quiz question. z stores the response I gave. Instead of using several lines of logic to decide whether I pass the quiz, use a one liner (you'll have to set x, y and z before you use this):

```
>> passedMyQuiz = x + y == z

>> passedMyQuiz

False

>> not passedMyQuiz

True
```

In the shell, you can you the Boolean literal values `True` and `False`, which must be capitalized. Side Note: You may remember from last week that Python variables are case sensitive, so the names `true` and `false` are available for you to set as variables, if you want to make your code confusing.

Beginning Control Flow

In IDLE, create a new Python file. Put the following into it:

```
x = 1

y = 2

z = 3

passedMyQuiz = x + y == z

if(passedMyQuiz):

     print('Congrats on the quiz!')

else:

     print('You need to work on your math skills.')
```

Save and run the file. I assume you can figure out what it will do.

Sometimes, we need to say "If the last condition was false, check this condition, and then...". The keyword for that is `elif`, which stands for 'else if'. Try running this code, changing age to get different results.

```
age = 8

if(age > 75):

     print('This ride is very intense. Consult your cardiologist.')

if(age > 55):

     print('Senior Discount')

elif(age > 18):

     print('One adult')

elif(age > 8):

     print('One kid')

elif(age > 2):

     print('Small children ride free.')

else:

     print('You must be at least 3 years old to ride.')
```

Logical Operators

Let's look at the quiz program again. What if I have a more complicated decision to make? If you pass your quiz and your project, I congratulate you for both. If you only pass one, I congratulate you for that. Finally, if you passed neither, I suggest that you study. One way we could do this is by nesting an `if` inside another `if`.

```
passedMyQuiz = True

passedMyProject = True

if passedMyQuiz:

    if passedMyProject:

        print('You\'re killing it!')

    # Still inside the passedMyQuiz block of code

    else:

        print('You\'re doing fine.')
```

```
# Outside the passedMyQuiz block of code.

elif passedMyProject:

     # You can't reach this block if passedMyQuiz, so we know you're 1 for
2

     print('You\'re doing fine.')

else:

    print('You need to study.')
```

This is harder than it needs to be. Python gives us logical operators and and or.

```
>> 2 + 2 == 5

False

>> 2 + 2 == 5 or 2 + 2 == 4

True

>> 2 + 2 <= 4 or 2 + 2 <= 5

True

>> 2 + 2 == 5 and 2 + 2 == 4

False

>> 2 + 2 <= 4 and 2 + 2 <= 5

True

>> True and True

True
```

Let's use these for our program.

```
passedMyQuiz = True

passedMyProject = True

if(passedMyQuiz and passedMyProject):

    print('You\'re killing it!')
```

```
elif(passedMyQuiz or passedMyProject):

    print('You\'re doing fine.')

else:

    print('You need to study.')
```

Challenge

Okay, genius, think you're ready to move on? Go for it. But first, try to fix this programs.

```
weekend = True

vacation = False


'''

See if you can get this code to let me sleep in. It never does.

'''


if weekend and vacation:

     print('Sleep in')

if weekend or vacation:

     print('Wake me at 7:30')

else:

     print('Wake me at 6')
```

See if you can figure how to make decisions based on user input. Just remember, user input reaches you as a string, even when it looks like a number. Try the exercises on page 93 of *Python Crash Course*, included as a photocopy.