

# Conditionals

July 26, 2016

## 1 Conditionals

### 1.1 If Statements

Up until now, we've directly told what the computer to do, and it executes those commands every time. What if we only want the computer to run code *some* of the time? This is where conditionals come in.

Let's start by defining a variable a.

```
In [1]: a=1  
        print a
```

1

Now that that cell is executed, let's look at the following piece of trivial code:

```
In [2]: if a < 5 :  
        a = a + 1  
        print a
```

2

The first time you run the above code, you should see the number 2 printed. This code does pretty much what you would expected. First, the computer checks to see if a is less than 5. If it is, then it adds 1 to a, then stores the result in a. Next, it prints a. Since a was 1, it met the condition, so the computer added 1 to a and printed it. a is now 2. If you select the cell above and hit shift enter, it will run again. What do you think will happen? Try it out!

Let's look more closely at that last piece of code. The if statement looks for a boolean. If that boolean is True, then the indented code below will be executed. If that boolean is False, then the indented code will be skipped. In our code, we used `a < 5`. Luckily, this evaluates to a boolean! Since that boolean is True, the next line is run. If you experiment by continuing to execute the cell, eventually a will equal 5, and executing the cell will effectively do nothing. This is because `a < 5` now evaluates to False.

With this knowledge, we can see why stuff like this works:

```
In [5]: if True :  
        a = a*3  
  
        print a
```

Since the if statement expects a boolean, we could just put a boolean in there! Of course, this particular example is the same as not using an if statement at all, because this code will always run! So while you'll never see this actually used, it does demonstrate what is going on behind the scenes and some similar tricks with different keywords might be useful later.

## 1.2 The Else and Elif Keywords

If you want the computer to execute a different piece of code should the if condition fail, you use an else statement. Here's an example:

```
In [15]: if a < 5:
        a = a + 1
        else:
        a = a - 4

        print a
```

1

The else statement is partnered with the preceding if statement. In the case that the if statement fails to execute, the else statement will be executed instead. Note that else statements are optional, but can be very useful.

What if we have a long list of conditions? We can use the elif statement, which is short for "else if"

```
In [24]: if a < 5:
        a = a + 1
        elif a < 10:
        a = a + 2
        else:
        a = a - 10

        print a
```

2

Play with the above cell. Try to guess what number the computer will print before you run the cell each time. Think about how the computer goes about printing the appropriate result.

Strings can also be used as conditions for if statements. Given a string, the computer will try to convert it to a boolean. This boolean will be true unless the string is an empty string.

```
In [8]: if "hello":
        a = a+1
        print a
```

5

```
In [9]: if "":  
        a = a+1  
        print a
```

5

It works similarly with numbers. If the computer tries to convert a number to a boolean, it will be true unless that number is 0.

```
In [10]: if 0:  
         a = a+1  
         print a
```

5

```
In [11]: if 4:  
         a = a+1  
         print a
```

6

Practice problems:  
Even/Odd  
Quadratic Equation solver

```
In [ ]:
```