Functions Exercises 2

More Calculator Stuff

Write the code for the following inside of the ~/intro_class/calculator.py file.

1. Set a variable called age to `add(30,4)`
2. Set a variable called height to `subtract(78, 2)`
3. Set a variable called weight to `multiply(6, 24)`
4. Set a variable called iq to `divide(100, 2)`
5. `print "Age: %d, Height: %d, Weight: %d, IQ: %d" % (age, height, weight, iq)`

More Survey Stuff

Write the code for the following inside of the ~/intro_class/survey.py file.

1. "Comment out" (preface each line with a # symbol) the lines that print:
   [name]'s favorite color is [color].
   [name]'s favorite hobby is [hobby].
   [name]'s favorite movie is [movie].
   etc.
2. Write a function called `print_survey_results` that takes in four arguments: `name`, `color`, `hobby`, `movie`.
3. The print_survey_results function should print:
   ```
   [name]'s favorite color is [color].
   [name]'s favorite hobby is [hobby].
   [name]'s favorite movie is [movie].
   ```
4. Call `print_survey_results` with the `name`, `color`, `hobby`, and `movie` variables as arguments.
5. Run your `survey.py` file. What happens?
6. Add parameters to `print_survey_results` for the two questions you made up.
7. Call `print_survey_results` with arguments for those new parameters.
8. Run your `survey.py` file to make sure it works.

Recursion

Create a file called recursion.py inside of ~/intro_class. Write the code for the following in the recursion.py file.

1. Write this function:
   ```
   def countdown(count):
       if(count == 0):
           print "Blastoff!"
       else:
           print count
           countdown(count-1)
   ```
2. Call the function with `countdown(5)` What happens?

3. Call the function with `countdown(10)` What happens?
4. See if you can understand what is happening in the function definition.
    a. What would happen if the conditional was removed?
    b. What would happen if the *recursive* call to countdown was `countdown(count+1)` instead of `countdown(count-1)`?
    c. What would happen if you called countdown with `countdown(-5)` instead of `countdown(5)`?
    d. `countdown` is called a *recursive* function because it calls itself inside of its function definition. It is important that recursive functions have a *base case* that stops execution of the function when it reaches a certain point. The other case in the conditional is called the *recursive case*. It is where the function calls itself.
        i. What is countdown's *base case*?
        ii. What is countdown's *recursive case*?
5. Create a function called countup that has one parameter, count. countup should *recursively* print the numbers from 1 up to 10.

Challenge:
A function object is a value you can assign to a variable or pass as an argument. For example, do_twice is a function that takes a function object as an argument and calls it twice:

```
def do_twice(f):
    f()
    f()
```

Here's an example that uses do_twice to call a function named print_spam twice.

```
def print_spam():
    print 'spam'

do_twice(print_spam)
```

1. Type this example into a python file called fun_challenge.py inside of ~/intro_class and test it.
2. Modify `do_twice` so that it takes two arguments, a function object and a value, and calls the function twice, passing the value as an argument.
3. Write a more general version of `print_spam`, called `print_twice`, that takes a string as a parameter and prints it twice.
4. Use the modified version of `do_twice` to call `print_twice` twice, passing `'spam'` as an argument.
5. Define a new function called `do_four` that takes a function object and a value and calls the function four times, passing the value as a parameter. There should be only two statements in the body of this function, not four.