

CS30 - Assignment 1

Part A: Due Wednesday at the beginning of lab.

Part B: Due Thursday Jan. 29, at 11:59pm



©2008 lefthandedtoons.com

<http://www.lefthandedtoons.com/245/>

1 Part A: my first function [3 points]

In Wing, create a new file called `[first_name][last_name]-assignment1A.py` (don't include the square brackets, e.g. my filename would be `davekauchak-assignment1A.py`).

- Define a function called `sphere_surface_area` that takes the radius as a parameter and outputs the surface area of a sphere with that radius.

Remember the basic structure for defining a function is:

```
def function_name ( parameters ) :  
    statement1  
    statement2  
    ...  
    return something # not all functions will have return statements
```

Remember that the way that Python can tell what is part of the function is based on the indenting.

- Put in a `print` statement that prints out the surface area of a sphere of radius 25 at the end of your program and then run your program. You should NOT have any `print` statements inside your `sphere_surface_area` function. Instead, you should call your function and print out the value returned.

Remember, we can print anything that represents a value:

```
print sphere_surface_area(25)
```

This statement has multiple parts. `sphere_surface_area(25)` calls our defined function, which will in turn, execute the statements inside your function. When it returns, that value will then be printed by the `print` statement.

- After running your program, play with the function in the Python shell.

After you've run your program, you can still interact with the Python shell. For example, you can type:

```
>>> sphere_surface_area(12.4)  
1932.2035136000002
```

Once you're comfortable with running Python programs with your own defined functions and with playing with them in the Python shell, submit the current version of your `.py` file online using the courses submission mechanism.

2 Part B: A semester abroad in Europe [11 points]



<http://www.bildergeschichten.eu/euro-cartoon-witz.htm>

For the main part of this assignment, you will write some more functions. Create a new file called *[first_name][last_name]-assignment1B.py*.

You're going to do a semester abroad in Europe and have decided to write a few functions that will help you out with some common questions you might find yourself asking while you're there.

1. [2 points] Write a function called `euros_to_dollars`, with a single parameter, the price in euros, and the function will give you the price in dollars. Lookup the current price exchange from euros to dollars online. For example, after running your program you could type:

```
>>> euros_to_dollars(13.5)
18.97695
```

(Note: depending on the exchange rate you use, your value will be slightly different)

Make sure that you are using the `return` statement and not printing the answer in your function. In particular, try running the following and make sure you get something similar:

```
>>> dollars = euros_to_dollars(13.5)
>>> print dollars
18.97695
```

2. [2 points] Write a function called `welcome` that doesn't take any parameters and prints out "welcome" in some european language. For example:

```
>>> welcome()
te lutem
```

Remember that you can create functions with zero parameters. To call a function without any parameters, you still need to put the parenthesis at the end.

3. [2 points] Write a function called `kilometers_to_miles`, with a single parameter, the number of kilometers, and the function will give you the distance in miles. For example, after running your program you could type:

```
>>> kilometers_to_miles(100)
62.137
```

4. [3 points] Write a function called `mpg_from_metric` that takes *two* parameters: first the number of kilometers and second the number of liters. The function returns the miles per gallon (i.e. miles divided by gallons) by converting the kilometers and liters appropriately. Remember, to have multiple parameters for functions, you separate them with commas.

```
>>> mpg_from_metric(400, 30)
31.358472666666668
```

5. [2 points] Write a function of your own that takes one or more parameters and does something interesting/useful. Brownie points for creative functions :)
6. **Extra credit (1 point):** The right way to calculate the `mpg_from_metric` function is to utilize other functions you write that do some of the work for you. To calculate the mpg, write an additional function `liters_to_gallons` and then use this function and your `kilometers_to_miles` function inside the `mpg_from_metric` function.

3 Commenting and documentation

An important part of programming is making sure that your code is well documented so that when someone else looks at it (or, more likely, you look at it a month later) you can get information about the program without having to read all of the code.

You should have the following in your program:

- Comments at the very beginning of the file stating your name, course, assignment number and the date.
- Each function should have an appropriate docstring.

- Other miscellaneous comments to make things clear. Don't go overboard, but do include comments where things are complicated or if you have a block of code that does something. For this assignment, you won't need much since the functions should all be pretty short.

In addition, make sure that you've used good *style*. This includes:

- Following naming conventions, e.g. all variables and functions should be lowercase.
- Using good variable names.
- Proper use of whitespace, including indenting and use of blank lines to separate chunks of code that belong together.

4 When you're done

Submit your .py file online using the courses submission mechanism. Note, you can resubmit the same assignment as many times as you would like up until the deadline.

5 Grading

	points
section 1	3
section 2	11
comments and code style	3
extra credit	1
total	17 (+1)

