

Week 1 Review

[Re-submit Assignment](#)

Due Jan 19 by 11:59pm **Points** 27 **Submitting** a text entry box or a file upload

Overview

When attempting these problems, please keep in my mind the nature of [Academic Honesty](#) in this course. This week you should submit your response to the Week 1 Review. This will support your review of the topics covered this week and prepare you for writing your program.

Programming Exercises

1. **[temp_converter.py]** Modify the temperature converter we developed in [1.10 Together Project: A Temperature Converter](#) to now convert from Fahrenheit to Celsius. A user should now input a temperature in Fahrenheit and the program should output the equivalent temperature in Celsius. Modify the input prompts and the output message as appropriate.

2. **[bitcoin_converter.py]** Write a program that converts bitcoin to dollars (usd). At the end of this description you will find an example of the inputs and outputs for a correctly written program.

The program should begin by outputting to the user the date and time at which you recorded the conversion rate for bitcoins to usd. In the example below I went to [Coinbase](https://www.coinbase.com/charts) (<https://www.coinbase.com/charts>) on 8/1/17 at 11:13 am and found the price of a single bitcoin to be \$2086 usd. I did not use any special Python libraries for concurrency conversion or dates nor should you.

After that output, the user should be prompted to input an amount of bitcoin they have. Notice in the example below the user's input value of **.5**. That value was typed in by the user as input, it is not a value printed out. We will use a convention of displaying user input in example interactions as **bold** text. Your program should allow the user to input any numeric value, not just **.5**. A user could enter **.33** bitcoins, or **2.81** bitcoins.

Finally the program should output the value of the input amount in dollars based on that conversion rate. For example, if you find the conversion rate that 1 bitcoin is worth 2086 usd, then half a bitcoin, or .5 bitcoin, would be worth 1043 usd.

Here is an example interaction with a correctly written program:

```
As of 8/1/17 at 11:13 am, bitcoin is currently trading at $2086 per bitcoin.  
Enter the bitcoin amount: .5
```

That is worth 1043 us dollars.

3. **[fight_song.py]** Write a program that outputs the following fight song. Your program is required to have a function named `sing_fight_song`. When the `sing_fight_song` function is called it should "sing" the song below by printing it out.

You should create other functions to show structure and to eliminate redundancy in your program. More precisely, your `sing_fight_song` function should not consist solely of print statements printing each line of the song. That would be highly redundant. It should call other functions that print reusable parts of the song.

```
Go, team, go!
Defeat your foe.

Go, team, go!
Defeat your foe.
Simply the best,
Better than the rest.
Go, team, go!
Defeat your foe.

Go, team, go!
Defeat your foe.
Simply the best,
Better than the rest.
Go, team, go!
Defeat your foe.

Go, team, go!
Defeat your foe.
```

4. **[phrase_repeater.py]** Write a program that repeats a phrase, given by a user, the number of times a user requests it be repeated.

For example, a user could input the phrase `'Lazy harp seal has no job'`. Then the user could input to repeat it `'3'` times. Given these inputs, the program should (1) output the phrase three times and (2) output which repetition this is by starting each line with the repetition number (note the `'1'`, `'2'`, and `'3'` below):

```
Input your phrase: Lazy harp seal has no job
How many times should it be repeated? 3
1 Lazy harp seal has no job
2 Lazy harp seal has no job
3 Lazy harp seal has no job
```

Again, note the bold text in the example above: this indicates a value that is typed and input by the user (i.e., **Lazy harp seal has no job**, and **3**). Your program should *not* be printing out the values you see in bold. The user should be able to type in any phrase where you see **Lazy harp seal has no job** and any number of times to repeat it where you see **3**. A correct program would also work like this for the following inputs:

Input your phrase: **You fill up my tummy with hugs**

How many times should it be repeated? 5

- 1 You fill up my tummy with hugs
- 2 You fill up my tummy with hugs
- 3 You fill up my tummy with hugs
- 4 You fill up my tummy with hugs
- 5 You fill up my tummy with hugs

Submission

Please post all necessary .py files to Canvas and include your answers to the questions under the "Canvas Submission" banner in the textbox provided.

Canvas Submission

When you submit this assignment here in Canvas, I would like you to answer the following question(s):

1. How many hours do you estimate you used completing this assignment?
2. What was easiest for you when completing this assignment?
3. What was the most difficult challenge you experienced when completing this assignment?

Week 1 Review

Criteria	Ratings			Pts
1. Temp Converter Input	2.0 pts Full Marks	1.0 pts Executing program provides input message, but message inaccurate	0.0 pts No Marks	2.0 pts
1. Temp Converter Output	2.0 pts Full Marks	1.0 pts Executing program provides output message, but message inaccurate	0.0 pts No Marks	2.0 pts
1. Temp Converter Correctness	2.0 pts Full Marks	1.0 pts Calculations modified, but incorrect	0.0 pts No Marks	2.0 pts
2. Bitcoin Converter Conversion Rate Message	2.0 pts Executing program provides conversion rate message with date, time, and rate	1.0 pts Executing program provides message but missing information	0.0 pts No Marks	2.0 pts
2. Bitcoin Converter Input	1.0 pts Executing program prompts for bitcoin amount input		0.0 pts No Marks	1.0 pts
2. Bitcoin Converter Output	2.0 pts Full Marks	1.0 pts Executing program provides output message, but message inaccurate	0.0 pts No Marks	2.0 pts
2. Bitcoin Converter Correctness	3.0 pts Full Marks	1.0 pts Executing program does Bitcoin to dollars computation, but is incorrect	0.0 pts No Marks	3.0 pts
2. Bitcoin Converter Style	2.0 pts Full Marks	1.0 pts Does not do one of the following: Use descriptive variable and/or function names	0.0 pts No Marks	2.0 pts
3. Fight Song: Output	2.0 pts Full Marks	1.0 pts Executing program provides output message, but message inaccurate	0.0 pts No Marks	2.0 pts

Criteria	Ratings				Pts
3. Fight Song: Use of Functions	3.0 pts Full Marks	2.0 pts Solution uses functions to reduce most redundancy	1.0 pts Solution still retains redundancy with single function	0.0 pts No Marks	3.0 pts
3. Fight Song: Style	1.0 pts Functions have descriptive names indicating what they do			0.0 pts No Marks	1.0 pts
4. Phrase Repeater: Phrase Input	1.0 pts Full Marks		0.0 pts No Marks		1.0 pts
4. Phrase Repeater: Num of Repetitions Input	1.0 pts Full Marks		0.0 pts No Marks		1.0 pts
4. Phrase Repeater: Output	2.0 pts Full Marks	1.0 pts Outputs phrase correct number of times, does not include repetition number		0.0 pts No Marks	2.0 pts
Answers Questions in Canvas Submission	1.0 pts Full Marks		0.0 pts No Marks		1.0 pts
Total Points: 27.0					