

## Python Journal Template

**Directions:** Follow the directions for each part of the journal template. Include in your response all the elements listed under the Requirements section. Prompts in the Inspiration section are not required; however, they may help you to fully think through your response.

Remember to review the Touchstone page for entry requirements, examples, and grading specifics.

**Name:** Sarah Gillard

**Date:** 01/29/2024

**Final Replit Program Join Link:** <https://replit.com/join/eqtbwdxlwd-sarahg21>

Complete the following template. Fill out all entries using complete sentences.

## PART 1: Defining Your Problem

In the United States, our body weight is recorded in pounds and the weights we buy for weightlifting are stated in pounds. However, when abroad or when following exercise programs that are not based in the U.S. it can be confusing as weights are stated in kilograms. The opposite problem will occur for those traveling to the United States and being asked for their weight in pounds or going to a gym and needing to select the appropriate weight in pounds instead of their normal kilogram value.

The problem to solve is I want to help people be able to more quickly and efficiently convert between kilograms and pounds using a program coded in the Python language. When executed, the program will ask the user how they are converting (kilograms to pounds, or pounds to kilograms). Next, the program will output what the weight is in the newly selected units. To solve the problem, the program will perform the calculations, for example, to convert 150 pounds to kilograms, the program will divide 150 by 2.20462, resulting in 68.0389 kilograms. The program will round answers to the nearest tenths place.

## PART 2: Working Through Specific Examples

Scenario 1: A weightlifter wants to convert 180 pounds to kilograms.

- Input: The starting weight is 180 lbs and the user wants that converted to kilograms.
- Solve: Solve for kilograms by dividing 180 by 2.20462, getting 81.6467237.
- Result: Output “The approximate converted weight is: 81.6 kg.”

Scenario 2: A model traveling to Europe needs to convert their weight in pounds to kilograms.

- Input: The starting weight is 110 pounds and they want that converted to kilograms.
- Solve: Solve for kilograms by dividing 110 by 2.20462, getting 49.89522004.
- Result: Output “The approximate converted weight is: 49.9 kg.”

Scenario 3: A personal trainer wants to convert 70 kilograms to pounds to select the appropriate weight for their client.

- Input: The starting weight is 70 kilograms and they want that converted to pounds.
- Solve: Solve for weight in pounds by multiplying 70 by 2.20462.
- Result: Output “The approximate converted weight is: 154.3 lbs.”

Scenario 4: An international student in the U.S. needs to report their weight in pounds.

- Input: The starting weight is 61 kilograms and they want that converted to pounds.
- Solve: Solve for weight in pounds by multiplying 61 by 2.20462.
- Result: Output “The approximate converted weight is: 134.5 lbs.”

## PART 3: Generalizing Into Pseudocode

Function (pounds\_to\_kilograms)

Input will be the starting weight in pounds

This function will take the weight in pounds and divide it by 2.20462 to convert it to kilograms

Function (kilograms\_to\_pounds)

Input will be the starting weight in kilograms

This function will take the weight in kilograms and multiply it by 2.20462 to convert it to pounds

Main program

Prompts user to enter starting weight

Prompts user to enter 'lbs' or 'kg' to specify what units the starting weight is in

If the user enters 'lbs' for starting weight, the program will run the function pounds\_to\_kilograms.

Else, if the user enters 'kg' for starting weight, the program will run the function kilograms\_to\_pounds.

Else, prompt that the entry is invalid and ask the user to enter 'lbs' or 'kg'.

Run the main program to start sending the output to the user

## PART 4: Testing Your Program

While originally running the program, the prompts were not showing as I defined the main function but did not call the main function to actually run. To get the program to start running, I researched how to do this and found that I could use the `__name__` variable. I found this information on Free Code Camp at the following link: <https://www.freecodecamp.org/news/whats-in-a-python-s-name-506262fe61e8/>. I defined my function with the name “main”, so to get this function to run I added this to the end of my code:

```
if __name__ == "__main__":  
  
    main()
```

While testing my program I also ran into an issue when I entered ‘Kg’ with a capital k. In order to fix this, I used the `.lower()` method, which returns a string of all lowercase characters. This will allow the program to run efficiently whether the user uses capitals or lowercase letters for ‘kg’ and ‘lbs’ when prompted to enter the starting units.

**Originally the code looked like this:**

```
if unit == 'lbs':  
  
    converted_weight = pounds_to_kilograms(starting_weight)  
  
    print("The approximate converted weight is: ", converted_weight, "kg.")  
  
elif unit == 'kg':  
  
    converted_weight = kilograms_to_pounds(starting_weight)  
  
    print("The approximate converted weight is: ", converted_weight, "lbs.")
```

**I updated the code using the `.lower()` method built into Python to produce the following code:**

```
if unit.lower() == 'lbs':  
  
    converted_weight = pounds_to_kilograms(starting_weight)  
  
    print("The approximate converted weight is: ", converted_weight, "kg.")  
  
elif unit.lower() == 'kg':
```

```
converted_weight = kilograms_to_pounds(starting_weight)

print("The approximate converted weight is: ", converted_weight, "lbs.")
```

## PART 5: Commenting Your Program

```
#define function for converting pounds to kilograms
def pounds_to_kilograms(weight):
    #calculate the weight in kilograms, rounded to 1 place after the decimal point
    return round(weight / 2.20462, 1)

#define function for converting kilograms to pounds
def kilograms_to_pounds(weight):
    #calculate the weight in pounds, rounded to 1 place after the decimal point
    return round(weight * 2.20462, 1)

#define function for the main program which will interact with the user
def main():
    #ask the user to enter their starting weight, stating it's numerical because they should only
    #enter a number value here
    starting_weight = float(input("Enter your numerical starting weight: "))
    #ask the user if the starting weight is in kilograms or pounds
    unit = input("Is your starting weight in pounds (lbs) or kilograms (kg)? Enter 'lbs' or 'kg': ")

    #if the user selects lbs(pounds), the variable converted_weight will be equal to the return
    #value when running the pounds_to_kilograms function. The program will run that function
    #and output to the user the weight in kilograms, rounded to 1 decimal place.
    if unit.lower() == 'lbs':
        converted_weight = pounds_to_kilograms(starting_weight)
        print("The approximate converted weight is: ", converted_weight, "kg.")
    #if the user selects kg(kilograms), the variable converted_weight will be equal to the return
    #value when running the kilograms_to_pounds function. The program will run that function
    #and output to the user the weight in pounds, rounded to 1 decimal place.
    elif unit.lower() == 'kg':
        converted_weight = kilograms_to_pounds(starting_weight)
        print("The approximate converted weight is: ", converted_weight, "lbs.")

    #if the user does not enter lbs or kg (capital or lowercase is fine because of use of the
    #.lower() function), the program will output an error message.
    else:
        print("Invalid unit entered. Please enter 'lbs' or 'kg'.")

    #call the main function to start the program
if __name__ == "__main__":
    main()
```

## PART 6: Your Completed Program

<https://replit.com/join/eqtbwdxlwd-sarahg21>