1. This function converts miles to kilometers (km). **1 / 1 point**

1. Complete the code to return the result of the conversion.

   NOTE: The following items occur outside of the function. Do not try to change the indentations on the associated code or you will receive an error.
2. Call the function to convert the trip distance from miles to kilometers.
3. Fill in the blank to print the result of the conversion.
4. Calculate the round-trip in kilometers by doubling the result, and fill in the blank to print the result.

```
# 1) Complete the function to return the result of the conversion
def convert_distance(miles):
    km = miles * 1.6  # approximately 1.6 km in 1 mile
    return km
# Do not indent any of the following lines of code as they are
# meant to be located outside of the function above

my_trip_miles = 55

# 2) Convert my_trip_miles to kilometers by calling the function above
my_trip_km = convert_distance(my_trip_miles)

# 3) Fill in the blank to print the result of the my_trip_km conversion
print("The distance in kilometers is " + str(my_trip_km))

# 4) Calculate the round-trip in kilometers by doubling the result of
#    my_trip_km. Fill in the blank to print the result.
print("The round-trip in kilometers is " + str(my_trip_km * 2))
```

Run

Reset

Correct

Woohoo! You've figured out how to make the functions do what they need to do, and remembered some things from the previous lessons. Way to go!.

2. This function compares two numbers and returns them in increasing order. **1 / 1 point**

1. Fill in the blanks, so the print statement displays the result of the function call in order.

Hint: if a function returns multiple values, don't forget to store these values in multiple variables

```
# This function compares two numbers and returns them
# in increasing order.
def order_numbers(number1, number2):
    if number2 > number1:
        return number1, number2
    else:
        return number2, number1

# 1) Fill in the blanks so the print statement displays the result
#    of the function call
smaller, bigger = order_numbers(100, 99)
print(smaller, bigger)
```

Run

Reset

Correct

Nice! You remembered how to accept multiple return values from a function. You're ready for the next lesson!

3. What are the values passed into functions as input called? 1 / 1 point

- ○ Variables
- ○ Return values
- ◉ Parameters
- ○ Data types

Correct
Nice job! A parameter, also sometimes called an argument, is a value passed into a function for use within the function.

4. Complete the first line of the "print_seconds" function so that it accepts three parameters: hours, minutes, and seconds. Remember to use the "def" keyword to tell the Python interpreter the block of code is intended to define a function. 1 / 1 point

```
1  def print_seconds(hours, minutes, seconds):
2      print(hours*3600+minutes*60+seconds)
3
4
5  print_seconds(1,2,3)
6  #output will print to the screen
7
```

Run

Reset

Correct

Here is your output:
3723

Correct. The formula should multiply the hours variable by 3600 and the minutes variable by 60, then add these two products to the seconds variable.

5. What is the purpose of the def keyword? 1 / 1 point

- ◉ Used to define a new function
- ○ Used to define a return value
- ○ Used to define a new variable
- ○ Used to define a new parameter

Correct
Awesome! When defining a new function, we must use the def keyword followed by the function name and properly indented body.