Lab Introduction

For this lab and future labs, you will be working with a partner. When you and your partner have completed the lab, notify a TA or instructor, who will ask you questions about your lab and check you both off as completed. Turn in a copy of your lab02.py file on Gradescope to the Lab02 assignment. Lab grades are based on putting forth a good effort and working with a partner.

Lab Task

1. Function Review
2. Cylinders
3. Mileage
4. Conceptual Questions

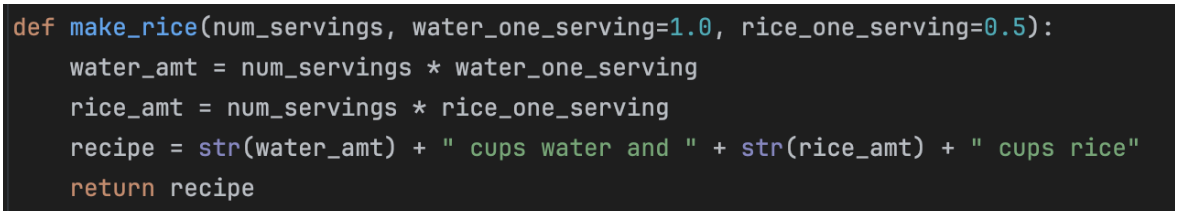
Function Review

A function is a block of reusable code that performs some action. Similar to mathematical functions, programming functions receive inputs and produce corresponding outputs. To get a function to do its job, you “call” it, supplying parameters, if required. Functions are helpful tools that let you divide your code into smaller parts. They make it easy to use the same code again without repeating the same lines again in your code.

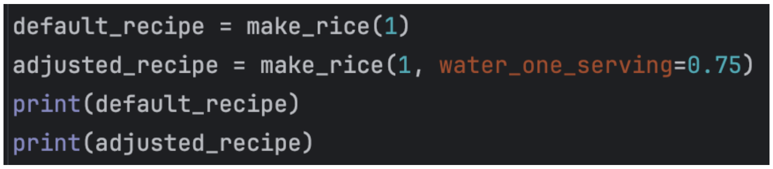
The components of a function are:

- Name of the function
- Parameters or arguments (input), if any
- Return values (output), if any

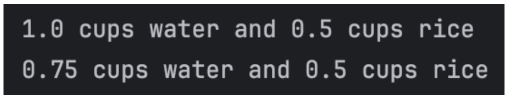
Here is an example function that takes in one required parameter, with two optional parameters, and returns a string. Remember, optional/default parameters must always come last!



```python
def make_rice(num_servings, water_one_serving=1.0, rice_one_serving=0.5):
    water_amt = num_servings * water_one_serving
    rice_amt = num_servings * rice_one_serving
    recipe = str(water_amt) + " cups water and " + str(rice_amt) + " cups rice"
    return recipe
```

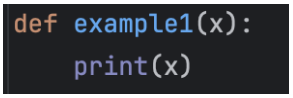


```python
default_recipe = make_rice(1)
adjusted_recipe = make_rice(1, water_one_serving=0.75)
print(default_recipe)
print(adjusted_recipe)
```

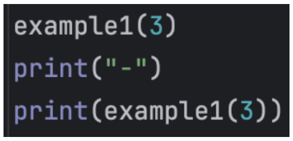


```
1.0 cups water and 0.5 cups rice
0.75 cups water and 0.5 cups rice
```

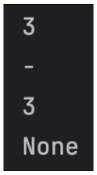
We run the function by calling its name, passing the required arguments into parentheses. Notice that a variable set equal to a function call has the value of whatever the function returns.



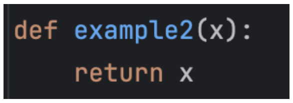
```
def example1(x):
    print(x)
```



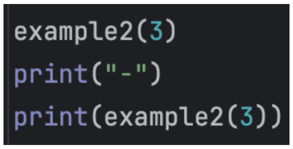
```
example1(3)
print("-")
print(example1(3))
```



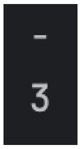
```
3
-
3
None
```



```
def example2(x):
    return x
```



```
example2(3)
print("-")
print(example2(3))
```



```
-
3
```

There is a key difference between returning a value in a function and printing a value within the function. Observe the following examples. In the first example, the function contained a “print” statement instead of a “return” statement, you can see that when the function call is printed, both “3” and “None” showed up in the output console. The reason that “3” appeared was because it was printed inside the function. The “None” appears because the function did not return any value. Therefore, the function does not evaluate to any value once it is run and when you try to print the function call, “None” is printed as it is the default return value of all Python functions. Therefore, if you ever want to print or store the result of a function call to a variable, you should return rather than print within the function.

Cylinders

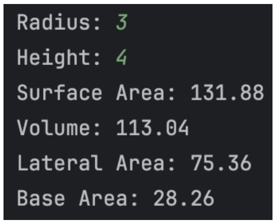
The first task is to create a cylinder stats calculator. Create a file called “lab02.py” to house your lab work. For this section of lab, you will write five different functions. By the end of this lab, you should be more comfortable creating and using functions. You will also see that functions help make your life easier by allowing you to repeat calculations very easily!

The first four formulas which you will write a function for are: Surface Area, Volume, Lateral Area, and Base Area. The formulas for each are…

- Surface Area = $2\pi rh + 2\pi r^2$
- Volume = $\pi r^2 h$
- Lateral Area = $2\pi rh$
- Base Area = πr^2

In these formulas, r is the radius of the base of the cylinder and h is the height of the cylinder. For your formulas, make a global variable for pi and set it equal to 3.14. You can name each function whatever you would like, but make sure they help tell you what they calculate (for example you can call the Surface Area function surface_area). Each function should take two required parameters, the radius and height, calculate the relevant value, and return it.

You will write one more function called “print_cylinder” where you will need to print out all the “stats” of a cylinder with the given height and radius. Rather than re-calculating each of the formulas, it is much easier to call each of the formula functions you just wrote. In your “print_cylinder” function, call each of the previously written functions to get the relevant values. Then print them out with some context using statements such as “Surface Area: ” followed by the calculated surface area. This function will not take in any parameters, but instead will prompt the user for input to be used to compute the cylinder statistics. The function should first prompt the user for the radius and then the height.



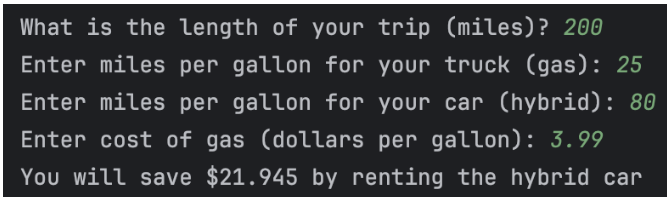
```
Radius: 3
Height: 4
Surface Area: 131.88
Volume: 113.04
Lateral Area: 75.36
Base Area: 28.26
```

Call print_cylinder in your program and test it out with the following example. This shows what should be printed when the user inputs a radius of 3 and height of 4.

Remember, the only function that should have print() in it is the print_cylinder function. Similarly, the only time input() should be called is within the print_cylinder function. Also, note that Python denotes exponents using **, so 2^3 would be written 2 ** 3.

Mileage

For this exercise, imagine you need to rent a car and are given two options, a hybrid and a truck. As the hybrid is more fuel efficient, you want to know how much money you would save by choosing the hybrid instead of the truck. In your "lab02.py" file, write a function that allows you to input the MPG (miles per gallon) of both vehicles, the length of your trip in miles, and the cost per gallon of gas and use it to output the total amount you would save if you chose the hybrid over the truck. Your function should not take in any variables, but instead utilize the input function within it to prompt users to enter the four values. Then, calculate the number of gallons each car would need to drive the given length of trip. Then, use the cost of gas in dollars per gallon to calculate how much it would cost to purchase that number of gallons for each car type. Finally, return a string stating the difference between the two prices.



```
What is the length of your trip (miles)? 200
Enter miles per gallon for your truck (gas): 25
Enter miles per gallon for your car (hybrid): 80
Enter cost of gas (dollars per gallon): 3.99
You will save $21.945 by renting the hybrid car
```

Here is an example of the printed output from running created function for the green inputted values. In this example we computed the following calculations to obtain the answer of $21.945 saved.

Number of gallons for truck: 200 / 25 = 8.0
Number of gallons for hybrid car: 200 / 80 = 2.5
Cost for truck gas: 8.0 * 3.99 = 31.92
Cost for hybrid car gas: 2.5 * 3.99 = 9.975
Difference between costs: 31.92 - 9.975 = 21.945

Challenge: This function returns out a long value for the amount saved, rather than just the rounded dollar amount. This doesn't make sense, as you cannot save a portion of a cent. How can you return the amount rounded to the nearest cent? Ask a TA for help!

Conceptual Questions

When checking you off, a TA will ask you one of the following questions. Discuss the following questions with your partner as you are working through the lab.

- What is the difference between printing and returning?
- Why should we use functions? How are they important?
- What is the difference between required and optional parameters?
- What is a variable? Why are they important?
- What are the differences between the string, int, and float data types?
- How can you tell what type a variable has? How can you change the type?