Phys 108 Name:_____

Loop and lists for Science!

One of the reason computers are so prevalent in their use, is that they can do very simple tasks very very quickly.

We will later play with a simulation that add vectors together. However, the simulation does not display the resultant vector. We will add code to the simulation to show the resultant vector with each new vector drawn.

Drawing a resultant vector means that we need to be able to add many vectors (as many as we draw!) together. While the computer can do this addition much faster than we can, we will need to tell it exactly what it must do!

To prepare us for the next computational thinking activity (adding code to a simulator) we will be creating a new function, "find_sum", that will take numbers from a list and add them all together/find the sum (this we be useful when we take vectors from a list and add them together to get the resultant!)

A function in python (the language we are using) is a small set of instructions that tell the computer what to do. This function can be used many times in many places in code. Think about buttons on your calculator. These buttons perform a specific task and can be called up many times at many different places!

Our function will take numbers from a list and add them all together/find the sum. A function has already been defined on line 5 with the line: def find_sum():

This is a much harder task than it appears, as a computer can only look at one item at a time and cannot look ahead to see what is coming. Therefore, to even go through a list, let alone add things on a list together, we will need to use a logical device called the "for loop"

Put simply a "for loop" will repeat an instruction for a given condition. So, we need to pick a condition that the computer will check. If this condition is true, the computer will perform a task, and check again to see if the condition is true or false. If the condition is false, it will not perform the task.

A real-world example of a for loop would be: Your instructor will hand out a worksheet for each student in the class without a worksheet. So the condition is if a student has a worksheet or not, and the action performed would be the instructor handing out the worksheet if the condition is met!

We have already determined that the task the computer will need to perform is to find the sum of all the numbers on the list. So we will need to use a variable called "sum". We also do not want this variable to have any value at the start! So we will need to start our function by setting this "sum" to zero. This is called initializing a variable.

On line 6 add: sum = 0 so that the code appears as below (the spacing is important!)

```
def find_sum():
  sum = 0
```

On line 7 is where we will start our for loop.

A for loop starts with: for _____ in _____:

Where we need to fill in the blanks. Our condition, before we add things together, is to make sure the item is in our list! A list (with its name) has been provided in the code, however, you are free to make your own too!

At this point your code should look like:

```
def find_sum():
  sum = 0
  for item in list1:
```

Or with “list1” substituted for your own new list.
You now have initialized the variable “sum”. Started the for loop, by setting up the condition that the item we are adding together must be in the list. The last step is to actually perform the addition.

On line 8 you add now take the “item” we have found in the list and add it to our variable sum (this will be the right side of our equation). The variable sum (the left side of the equation) will then be set equal to this addition in order to keep track of all added numbers! This gives:

```
 sum = sum + item
```

At this point all of the added code should now look like:

```
def find_sum():
  sum = 0
  for item in list1:
    sum = sum + item
```

We have one last line to add in (on line 9). We need to finish by having the function actually return with the answer we have asked it for. Think about plugging in a whole bunch of numbers to add into your calculator. To get the final answer you still need to hit “return” or equals to, for the final number.

Therefore the last line we need to add is : return sum

This will allow the computer to call up our function “find_sum(): and get back the number that represents the sum!