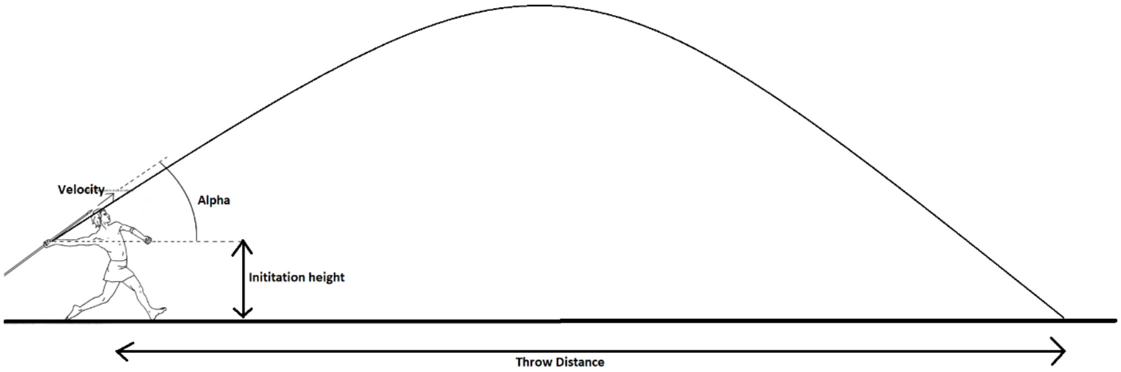
Python Exercises

Functions:

1.
    a. Go to ex2_str exercise 1 (related to ex1_print_and_var exercise 2) and encapsulate the code in a function called “say_my_name()” which has no input and no output, but should print your name in a cool format.
       Call the function and verify that it works.
    b. Do a minimal effort documentation of the function autogenerated by your editor, or formatted as we showed in class.
2.
    a. Go back to ex2_str exercise 3. And again, encapsulate the code for exercise a), in function called “bart_cheat_code” which should take two inputs, “numb_of_repetition” and “punishment_text” and return the string.
       Call the function with a variety of inputs and verify that it works.
    b. Set a default for “numb_of_repetition” to 10.
       test that it works.

3. Go back to ex4_if_else_bool exercise 1 and encapsulate the calculator logic in a function called “my_calculator”. The function should take 3 inputs, “first_numb”, “operator” and “second_number”
   and return the calculated value.
   Make sure the code still takes inputs by the user and prints the result, but both the inputs and print should be outside the function.
   Verify the code still works as usual with the function.
4.
   a. Go back to ex3_numb_input exercise 3 (NOT the numpy exercise from ex6_set_tuple_numpy exercise 3) and encapsulate the calculation logic into a function called “throw_distance” which takes 3 inputs, “velocity”, “angle”, “init_height”.

      Set init_height default to 1.8. The function should return throw length.
   b. Write a new function called “kmh_to_ms” which takes as input a velocity in km/h and return the m/s.

Change the code in “throw_distance” such that it uses the “kmh_to_ms” function.

c. Write a new function called “velocity_decomposition”. The function should take two inputs “velocity” and “angle”, and return two values, the horizontal and vertical velocity.
Change the code in “throw_distance” such that it uses the “velocity_decomposition” function.

d. Write a new function called “airtime” which takes as input the velocity in vertical direction “velocity_y”, “init_height” and gravitation “g” (with default 9.81) and returns the airtime.
Change the code in “throw_distance” such that it uses the “airtime” function.

e. Create a new function called “throw_possition”, which takes the same inputs as “throw_distance” with an extra variable called “time”.
The function should return the distance (horizontal velocity * time), only if the input “time” is smaller than “airtime”, else just the result of “throw_distance” without the time parameters.
Try to reuse the sub functions above instead of rewriting logic 😊

f. Do a minimal effort documentation of the two functions “throw_distance” and “throw_possition” as autogenerated by you editor, or format as we showed in class.

5. Go back to ex5_list exercise 3 and make a “quantile” function which takes two inputs, “series” and “alpha” (which defaults to 0.05).
   and returns two values, the numerical lower and upper confidence interval of the numbers in the series.
   And do a minimal effort documentation of the function autogenerated by you editor, or format as we showed in class.