SI206
Discussion 4

Testing tips

Remember to first enumerate what are the different scenarios you are testing. It will help you better test your programs!

It is often useful to FIRST write tests, THEN write the program. Tests will help you think of how your program will behave in edge cases.

Now, go to github and clone the repository for today’s discussion section.

count_a

We made a function, count_a, that counts the number of a’s in a string. You are going to test and see if it works.

(1) Talk to the people in your breakout room. What are the different test cases we need to test?

(1) Write tests for these test cases

(1) Oh no! The GSI’s and IA’s messed up when creating count_a. Fix it! (Hint: first see if you can spot where likely errors are by looking at what test cases fail).

Warehouse and Items

Item class contains information about an item (name, price and stock)

Warehouse class stores items.

Now you will implement (and test!) methods for the Warehouse class.

(We have created some items in setUp for you to use – feel free to use them if you wish!)

add_item in the Warehouse class

add_item:

add_item is a method for the Warehouse class that adds an item to the warehouse.

Write tests to determine if it will work.

Implement the method in the Warehouse class.

get_max_stock in the Warehouse class

get_max_stock:

get_max_stock is a method for the Warehouse class that finds and returns the item highest in stock

Write tests to determine if it will work.

Implement the method in the Warehouse class.

get_max_price in the Warehouse class

get_max_price:

get_max_price is a method for the Warehouse class that finds and returns the item with the highest price

Write tests to determine if it will work.

Implement the method in the Warehouse class.