

Name: Tristan Bragg

Date: 11/13/19

Description: Explaining the difference between mutating a value and rebinding a variable

The main difference I got out of between the difference between mutating a value and rebinding a variable was that you can mutate a value and still have the variables set to each other. If you were to mutate a value as "num_1 = 10" while "num_2 == num_1" you would just be changing what num_2's value is. That's because num_2 == (equivalent of) num_1. Changing num_1's value means num_2's value is also changed. That's where rebinding a variable comes into a difference. If you were to swap "num_1 == num_2" while "num_2 = 77", num_1's variable/value would be swapped to 77. Then when you call on both num_1 and num_2 it would return 77. You can also establish rebinding a variable with "is". You can use the is operator to test whether they are bound to the same variable. Easiest and cleanest way for me to explain it is as such:

num_1==num_2 ←--- setting num_1 to be bound to num_2's variable/value.

num_2==num_1 ←--- this is also rebinding the variable because now it is num_2 which is the equivalent to num_1

num_2 = 117 ←--- this is mutating the value. Now if you were to call on num_1, it would return 117.

The reason why this could trick someone up is by missing a "=" when possibly rebinding a valuable and because it all looks the same at first glance. This can trick you up if you are checking to see if they are bound to the same object by using "is". That's because if you aren't paying attention and applying your code properly, when you check with "is" it can come back across "False". That is how it confused me at first at least.

By the way, sorry for the poor effort on the homework this week. I got into a car accident while moving cities and at the same time had to study for CS 225 exam and was not able to balance everything well.