# Author: William Casey Bellew
# GitHub username: psykibear
# Date: 2/20/23
# Description:
In your own words, explain the difference between mutating a value and rebinding a variable. Give your own example of code that might trip up someone who doesn't get the difference and explain why it might trip them up. This part (7c) should be submitted as a pdf here in Canvas.

In Python you can define variables within the code with initial values while also changing their values later.

When you "bind" a variable, you make the initial definition:

- Variable_1 = 7
  - This binds the value of 7 to the variable "Variable_1"

You can continue this thought process even if you are using multiple variables or using variables to define other variables:

- Variable_2 = 11
  - This binds the value of 11 to the variable "Variable_2"
- Variable_3 = Variable_1
  - This binds the value of "Variable_1" to the value of "Variable_1"

Rebinding a variable, you must first have a definition of what the variable is through binding followed by "redefining" or rebinding the variable.
Based on the values of variables defined above:

- Variable_1 = 7
- Variable_2 = 11
- Variable_3 = Variable_1 = 7

In lines of code, we can rebind a variable by changing its definition.

- Variable_1 = 7
- Variable_2 = 11
- Variable_3 = Variable_1 = 7
- Variable_1 = Variable_2

Stepping through the code, this step 4 changes the value of Variable_1 to the value of Variable_2, but where it may confuse someone, it does NOT change the value of Variable_3. You only rebind or "redefine" the variable moving forward from that line of code. No other variable value will change unless they are also rebound separately.

Bind/Rebinding Example:

Code:

```
X = 12
Y = 27
Z = 76

Z = Z + X

Print (X)
Print (Y)
Print (Z)
```

Output:

```
12
27
103
```

This code just defines the initial definitions of the variables followed by a rebound definition of the variable Z.

Mutating is very similar to the functions of binding and rebinding a variable, but it takes the initial value and “adds” to the value. This would then take the original value, usually in a list, and add another value to the list.
Here is an example:

- List_1 = [7,11]
    - This then binds/defines the variable value [7,11] with the variable List_1

Now we create a second variable that is equal to the first variable:

- List_2 = List_1
    - This then binds the variable value List_1 with the variable List_2
    - List_2 = [7,11]

Now we will manipulate the data with the append function:

- List_1.append(88)
    - List_1 = [7,11,88]
    - List_2 = [7,11,88]

This updates the value of the variables versus creating any new value. This is where the difference between the mutating versus rebinding comes into play. Mutating the variable through the append function updates the value for both variables, where rebinding the variables just changes the value of the variable being changed and not any subsequent “connected” variables. If you wanted to just change one of the two connected variables then you’d rebind the value versus mutating the variable.

Mutating Example:

Code:

```
X = [12,27]
Z = X

X.append[76]

Print (X)
Print (Z)
```

Output:

```
[12,27,76]
[12,27,76]
```

This code just defines the initial definitions of the variables followed by mutating the definitions of the variables using the append function.