# **CS 1101-01: Discussion Forum Unit 6**

Godknows Egi

Bachelor of Science in Computer Science, Uopeople

CS 1101-01 - AY2024-T3: Lists

Bianca Gilyot

March 13th , 2024

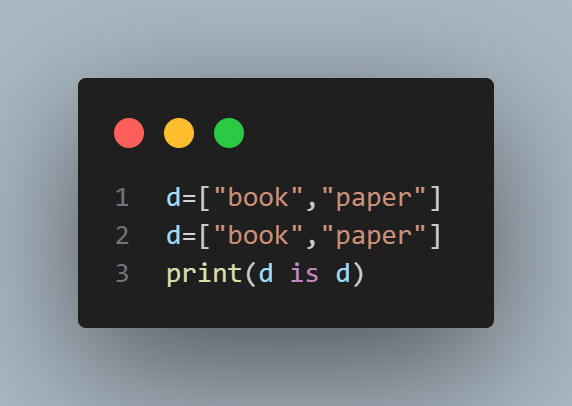
**Discussion Assignment**

* Use the terms **"equivalent"** and **"identical"** to distinguish between objects and values. Illustrate the difference further using your own examples with Python lists and the **“is”** operator.
* Using your own Python list examples, explain how objects, references, and aliasing relate to one another.
* Finally, create your own example of a function that modifies a list passed in as an argument. Hence, describe what your function does in terms of arguments, parameters, objects, and references.

**Solutions**

1. Objects serves as a container that retain/hold values while values can be contained inside an object. The ‘is’ operator is used to show if two objects similar or not , The return response for this is always a bool i:e True or False.

But in lists objects, if different variables refer to the same objects they are not similar but equivalent as stated on Downey, 2015, *List (chapter 10.10),* If the ‘is’ operator is used in this case, the python interpreter returns False.

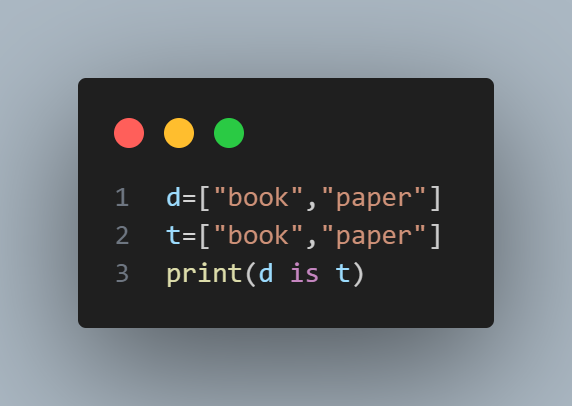


*Fig 1 (identical object)*

**

Fig 1 Output *(identical object)*

This returned **true** because these objects are similar and equivalent

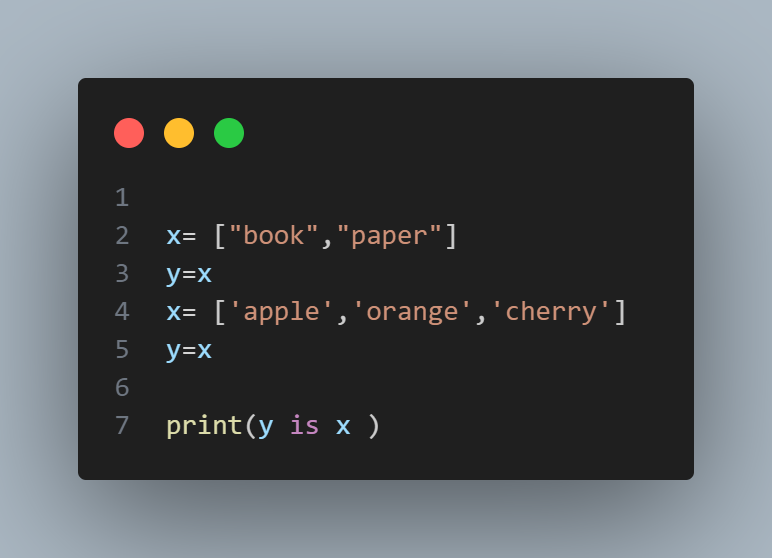


*Fig 2 (equivalent but not identical object)*

**

*Fig 2 Output (equivalent but not identical object)*

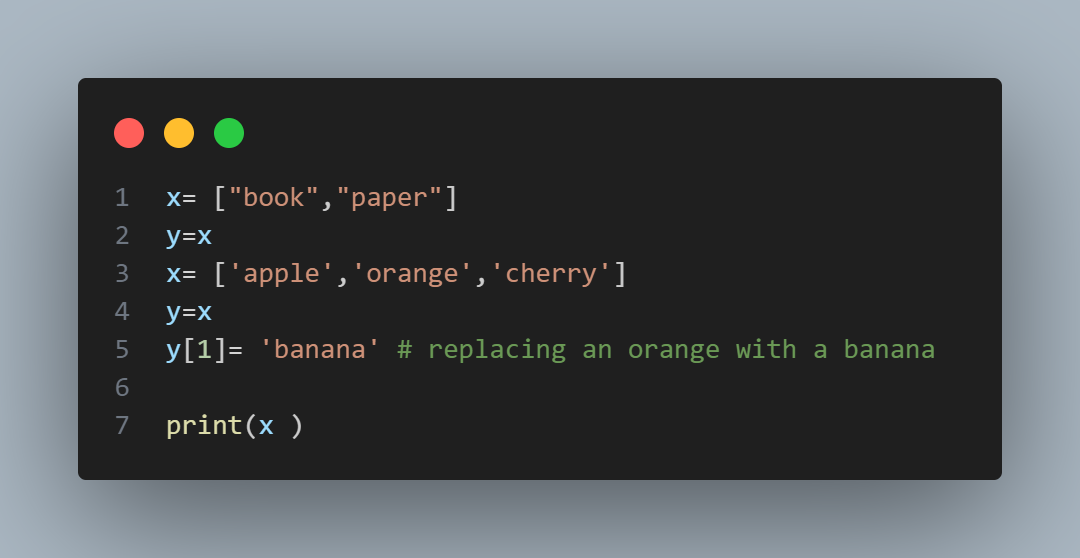
1. Aliasing is when an object has more than one reference. While reference refers the assignment of a variable to an object.



*Fig 3 (equivalent but not identical object)*

In the above example, the object of  **x** has been aliased. Changes made to either variable have an effect on the other as stated (Downey, 2015), *List (chapter 10.12)*.

Here is an example.



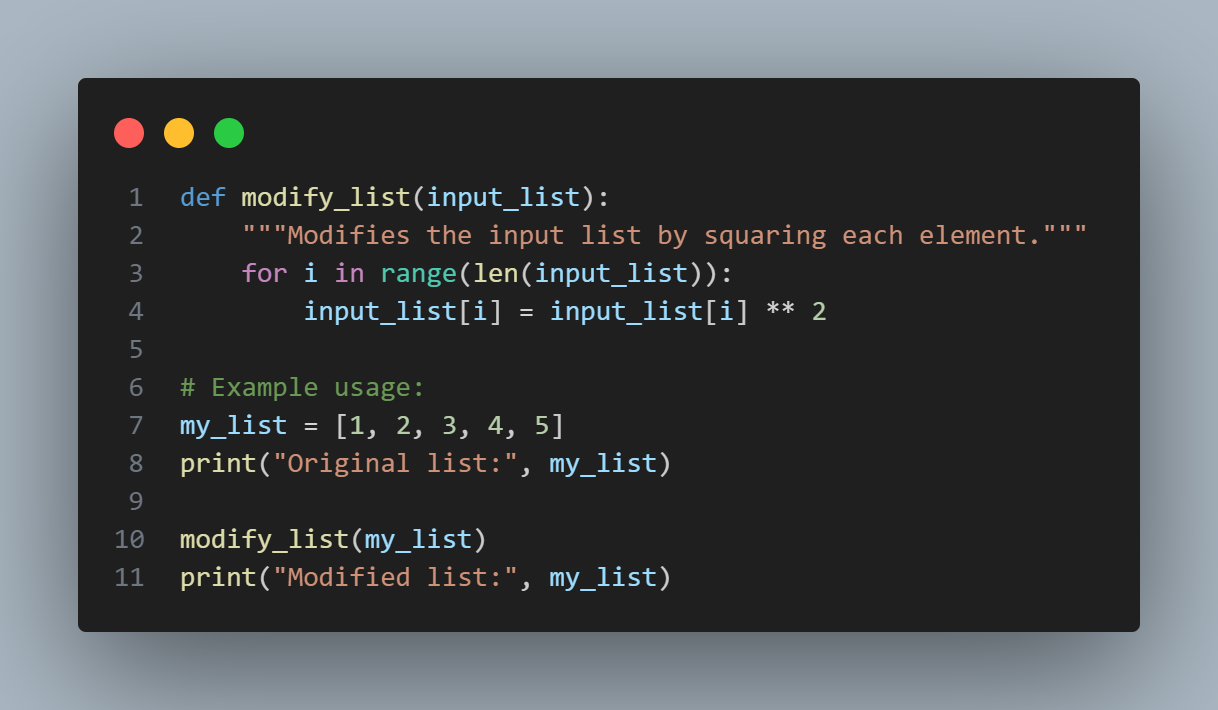
*Fig 3 (Aliased variables with changes that affect both )*

**

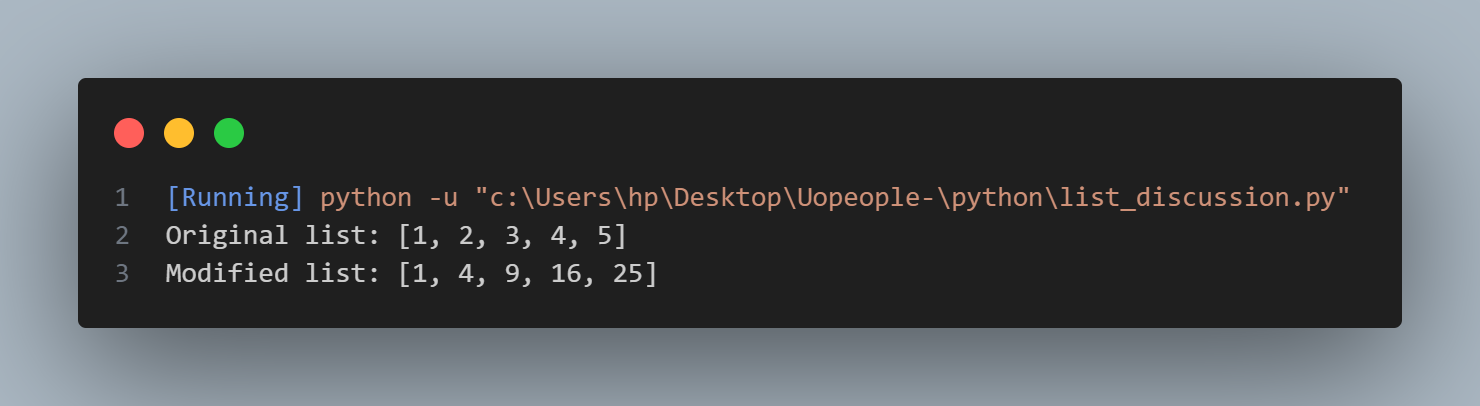
*Fig 3 Output (Aliased variables with changes that affect both )*

The above example shows that changes made to y have an effect on x

3.



*Fig 4 (function to modify list )*

**

*Fig 4 Output (function to modify list )*

The function **modify\_lis**t takes one parameter **input\_list**, which is the list to be modified. Inside the function, a for loop iterates over each element of the input\_list.

Each element is squared using the \*\* operator and then assigned back to the corresponding index in the list.The original list passed to the function is modified in place.

Now, let's discuss the concepts of arguments, parameters, objects, and references:

**Arguments**: In Python, an argument is the actual value that is passed to a function when it is called. In the example above, **my\_list** is the argument passed to the modify\_list function.

**Parameters:** Parameters are the variables in a function definition that receive the values of the arguments passed to the function. In the **modify\_lis**t function, **input\_list** is the parameter that receives the value of the argument **my\_lis**t.

**Objects:** In Python, lists are objects, meaning they are instances of the built-in list class. When you pass a list to a function, you are passing a reference to the same list object, not a copy of the list. This means that modifications made to the list inside the function will affect the original list outside the function.

**References**: When you pass an object (such as a list) to a function in Python, you are passing a reference to the object, not the object itself. This reference allows the function to access and modify the original object. In the example above, input\_list inside the function is a reference to the same list object as **my\_list** outside the function. Therefore, any modifications made to **input\_list** inside the function also affect **my\_list** outside the function.

**References**

Downey, A. (2015,). Think Python: How to think like a computer scientist ,*Chapter 10.10,10.12-Lists (pp 97-96)*

<https://greenteapress.com/thinkpython2/thinkpython2.pdf>