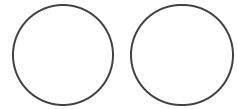


Python programming - mutable and immutable objects.



Maxime Guy BAKUNZI

Machine Learning Enthusiast | Python Developer | Aspiring AI Specialist



March 25, 2023

Firstly, we learned that every object in Python has a unique and constant identity throughout its lifetime. The 'id' function returns the identity of an object. Objects can be of two types: mutable and immutable. Mutable objects can be changed, whereas immutable objects cannot be altered after creation.

We delved into the differences between mutable and immutable objects. Immutable objects are hashable and can be used as keys in dictionaries, while mutable objects cannot. Changing an immutable object creates a new object while modifying a mutable object alters its existing value.

Next, we discussed the significance of mutable and immutable objects in Python programming. The distinction between mutable and immutable objects is critical for writing efficient code. The immutability of some objects ensures that they remain constant throughout the program, which can be crucial in specific scenarios.

We also explored how Python treats mutable and immutable objects differently. Python provides a different mechanism for copying mutable and immutable objects. Immutable objects are copied by reference, while mutable objects are copied by value. This difference leads to different memory allocation patterns, which can affect the performance of the code.

Finally, we looked at how Python passes arguments to functions and how this relates to mutable and immutable objects. Immutable objects are passed by value, while mutable objects are passed by reference. This implies that any changes made to mutable objects inside a function will reflect outside the function. In contrast, any changes made to immutable objects inside a function will not reflect outside the function.

In conclusion, this project helped us understand the importance of mutable and immutable objects in Python programming. It is essential to recognize the differences between mutable and immutable

objects to write efficient code. Understanding how Python handles mutable and immutable objects is crucial for effective programming.