Manuel Fernando Planas Ron

**¿Qué es mutabilidad en el contexto de Python?**

En Python, la mutabilidad se refiere a la capacidad de cambiar un objeto después de haberlo creado. Los objetos mutables son aquellos que se pueden modificar en su lugar, mientras que los objetos inmutables no. Los ejemplos de objetos mutables en Python incluyen listas, conjuntos y diccionarios, mientras que los ejemplos de objetos inmutables incluyen tuplas, cadenas y números. Cuando se modifica un objeto mutable, cualquier referencia a ese objeto reflejará el cambio, mientras que con un objeto inmutable, se debe crear un nuevo objeto para reflejar cualquier cambio. Comprender la mutabilidad es importante en Python, ya que puede afectar la forma en que se almacenan, acceden y modifican los datos en un programa.

**¿Cuándo una variable es mutable o inmutable?**

In Python, mutability refers to an object's ability to be changed after creation. Mutable objects can be modified after creation, while immutable objects cannot. Examples of mutable objects in Python include lists, dictionaries, and sets, while examples of immutable objects include strings, tuples, and numbers.

Understanding mutability is important in Python as it can impact how data is stored, accessed, and modified in a program. For example, if you modify a mutable object, all references to that object will be affected. On the other hand, if you modify an immutable object, a new object will be created instead.

**¿Cuándo una variable es mutable o inmutable?**

En Python, la mutabilidad se refiere a la capacidad de un objeto para cambiar después de la creación. Los objetos mutables se pueden modificar después de la creación, mientras que los objetos inmutables no. Los ejemplos de objetos mutables en Python incluyen listas, diccionarios y conjuntos, mientras que los ejemplos de objetos inmutables incluyen cadenas, tuplas y números.

Comprender la mutabilidad es importante en Python, ya que puede afectar la forma en que se almacenan, acceden y modifican los datos en un programa. Por ejemplo, si modifica un objeto mutable, todas las referencias a ese objeto se verán afectadas. Por otro lado, si modifica un objeto inmutable, se creará un nuevo objeto en su lugar.