perplexity

so that means new memory isnt created

When you reassign a variable's value in JavaScript—even when changing its type (for example, from Number to String or to BigInt)—the JavaScript engine may allocate new memory for the new value, but this is handled internally, and the process is abstracted from you as a programmer. For primitive reassignments of the same type, the value is typically just overwritten in the same memory slot, with no new memory allocated and no memory wastage[1] [2] [3].

However, if you assign a variable to a different type (e.g., a number to a string), JavaScript dynamically manages the required storage. The old value is eligible for garbage collection, and enough memory is made available for the new type. The specifics of how and when memory is actually allocated or reused depend on the JavaScript engine implementation[2] [4]. For you, as a developer, this means you do not directly control or need to worry about the details—JavaScript's automatic memory management takes care of it[1] [3].

So, if you only change a variable's value within the same primitive type (number to number, string to string, etc.), new memory isn't created; it's simply overwritten. When you change the type, the engine reallocates as necessary, but this transition is fully managed behind the scenes and is not considered memory "wastage" in practical terms[1] [3].

1. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Memory_management
2. https://okosunmary.hashnode.dev/primitive-values-and-reference-values-as-regards-memory-allocation
3. https://www.geeksforgeeks.org/javascript/memory-management-in-javascript/
4. https://vinoo.hashnode.dev/primitives-vs-objects-in-javascript