**Context Management:**

* \_contexts: This class variable holds a list of dictionaries, where each dictionary represents a context (scope).
* \_open\_contexts: This variable tracks the number of open contexts.
* enter\_context: Creates a new context by adding a new dictionary to \_contexts and increments \_open\_contexts.
* exit\_context: Removes the last context dictionary (if more than one exists) and decrements \_open\_contexts. It raises an exception if you try to pop the global context.

**Symbol Definition and Lookup:**

* define: Defines a symbol in the current context (the last dictionary in \_contexts). It checks for existing contexts and raises errors if the name is already defined or no context exists.
* lookup: Searches for a symbol by name, starting from the current context and iterating backwards through parent contexts. It returns the symbol if found, otherwise None.
* \_\_getitem\_\_: Allows retrieving symbols by name (str) or by the Symbol object itself. It searches through contexts in reverse order and raises a KeyError if not found.
* \_\_setitem\_\_: Allows setting symbols by name (str) or by the Symbol object in the current context. It checks for existing definitions and raises a KeyError if a symbol with the same name already exists.

**Additional Methods:**

* get\_all\_symbols: Returns a flattened dictionary containing all symbols from all contexts.
* \_\_repr\_\_: Provides a string representation of the SymbolTable object, showing a machine-readable representation of a context so that there is no recursively printed child table inside its parent’s context table.
* \_\_str\_\_: Returns a table-like representation of the SymbolTable object, showing the contents of each context