

# JavaScript Scope

The scope is the current context of execution in which values and expressions (an expression is a valid unit of code that resolves to a value) are "visible" or can be accessible. If a variable or expression is not in the current scope, it will not be available for use. Scopes can also be layered in a hierarchy, so that child scopes have access to parent scopes, but not vice versa.

JavaScript has the following kinds of scopes:

- Global scope: The default scope for all code running in script mode.
- Module scope: The scope for code running in module mode.
- Function scope: The scope created with a function.
- Block scope: The scope created with a pair of curly braces `{ }` (a block). Only variables declared with **let** and **const** are block-scoped