



JavaScript Scope

More About Global Scope

Learning objective: By the end of this lesson, students will understand the concept of global scope in JavaScript, particularly the role of the `window` object in browsers.

The `window` object represents the global scope in our browsers. This means it is at the top of the scope chain, and its properties are available to every function we write.

It is generally bad practice for our programs to create variables in the global scope because this can overwrite data that JavaScript libraries, frameworks, or other routines use.

Creating many global variables is known as “polluting the global scope.” When we define a variable (or function) in the global scope, it becomes a property of the `window` object. We can see this in action by adding `var pollution = 'is bad'` to our code, then add a console log for `window.pollution` (don’t forget the dot). If we run the code, we will see `"is bad"` console logged.

 Interestingly, using `var`, `let`, or `const` in the global scope creates a global variable; however, those created using `let` and `const` do not create properties on the `window` object as `var` does.