JavaScript Variable Scope Notes

Overview

- Understand the difference between global and local variables in JavaScript based on where they are declared and their accessibility.

Key Concepts

- **Global Variables**:
  - Declared in main code (outside functions) with `var`.
  - Accessible everywhere unless shadowed by a local variable with the same name.
- **Local Variables**:
  - Declared inside a function with `var` or as parameters (implicitly declared).
  - Only accessible within that function.
- **Differences**:
  - Global: Declared in main code, usable everywhere unless shadowed.
  - Local: Declared in function, usable only in that function.
- **Shadowing**:
  - Local variable with same name as global takes precedence inside function.
  - Example: Global `theSum = 1000`, local `theSum = 4` inside function; function uses local.
- **Returning Values**:
  - Local variables value can be returned to a global variable.

    ```
    var theSum = addNumbers(); // Global theSum gets
       local value
    function addNumbers() {
      var theSum = 2 + 2; // Local
      return theSum; // Returns 4
    }
    ```
- **Best Practice**:
  - Avoid global variables in functions to prevent confusion/errors.
  - Pass data explicitly via arguments/parameters.
- **Pitfall**:
  - Using a variable in a function without `var` makes it global. Always use `var` for local variables.

Example

```
var theSum; // Global
function addNumbers() {
  var theSum = 2 + 2; // Local, shadows global
  alert(theSum); // Shows 4
}
addNumbers();
alert(theSum); // Undefined or breaks, global theSum
   unassigned
```