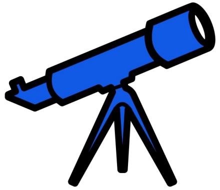
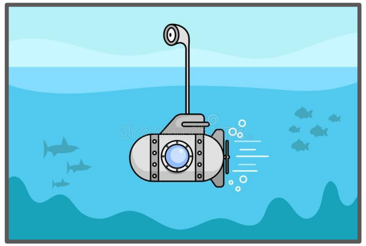
JavaScript **scope**: What is it?

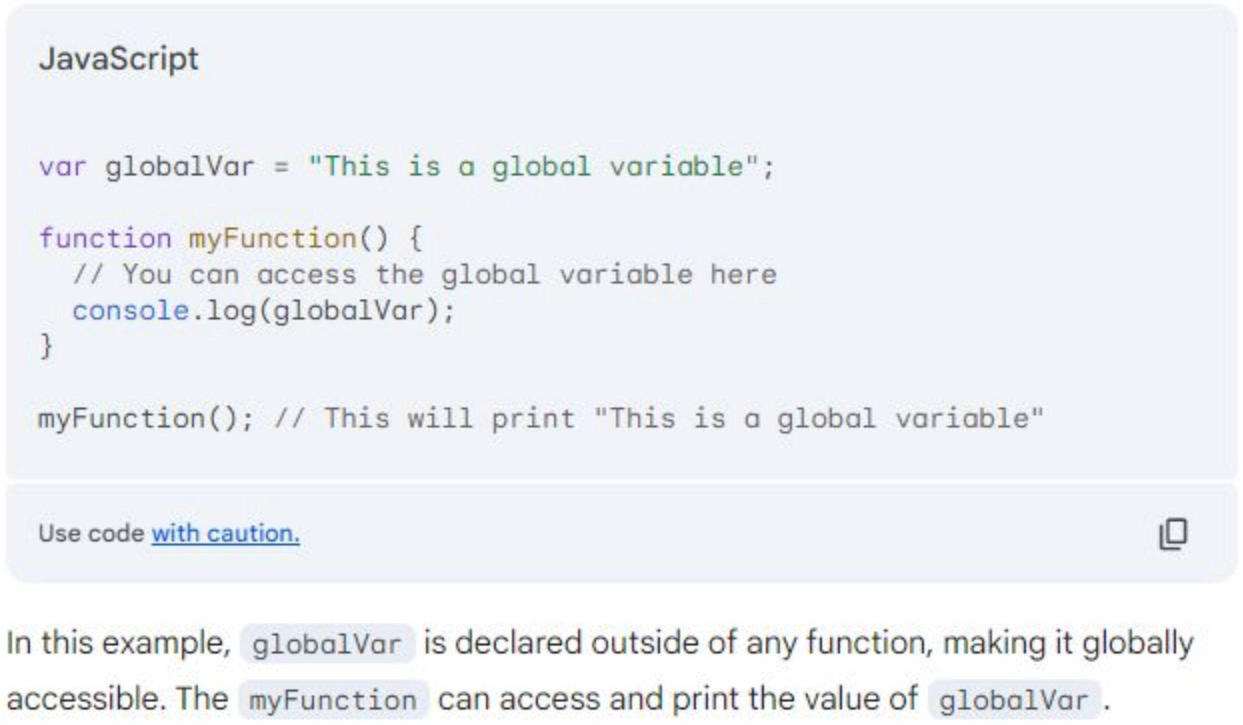



telescope	periscope	stethoscope
tele-	**peri-**	**stetho-**
at a distance (from Greek)	*around (from Greek)*	*chest (from Greek)*
*What do you think **scope** means or has to do with?*		

Scope: Official Definition

What part of your script can use your variable?

global scope	**function** scope	**block** scope
Any part of your script can access/use the variable	Variable is only accessible/visible **within a function**	Variable is only accessible/visible within a code block **{ }**
Avoid using variables that have global scope		Applies to variables *declared* (created) using **let** or **const**
		A code block begins with **{** and ends with **}**


JavaScript

```javascript
var globalVar = "This is a global variable";

function myFunction() {
  // You can access the global variable here
  console.log(globalVar);
}

myFunction(); // This will print "This is a global variable"
```

Use code with caution.

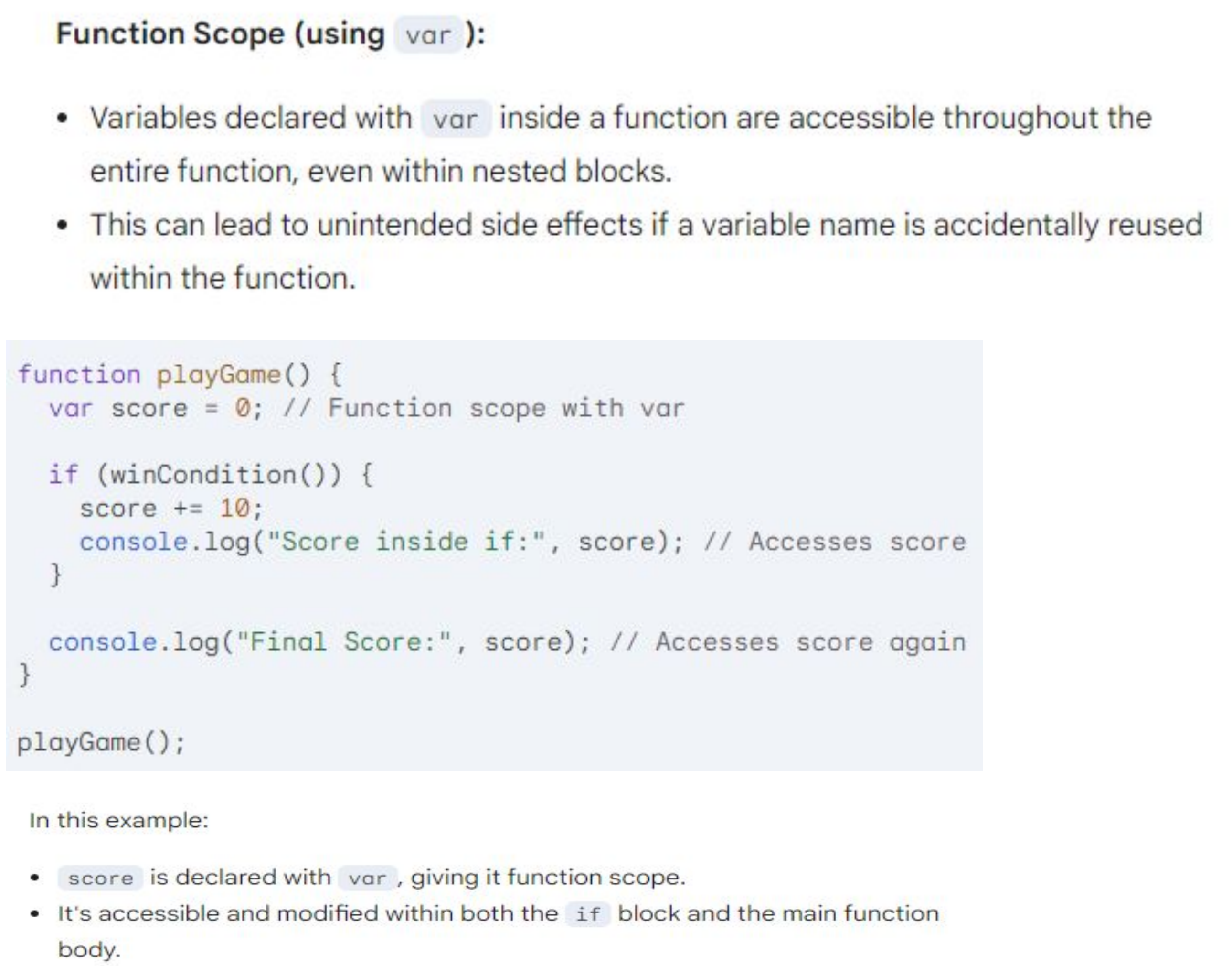
In this example, `globalVar` is declared outside of any function, making it globally accessible. The `myFunction` can access and print the value of `globalVar`.

function: A named chunk of code that performs a specific task.

Modern JavaScript does not use the keyword **var.**

Use **let** or **const** instead.

Variable with GLOBAL scope ☹



Function Scope (using `var`):

- Variables declared with `var` inside a function are accessible throughout the entire function, even within nested blocks.
- This can lead to unintended side effects if a variable name is accidentally reused within the function.

```
function playGame() {
  var score = 0; // Function scope with var

  if (winCondition()) {
    score += 10;
    console.log("Score inside if:", score); // Accesses score
  }

  console.log("Final Score:", score); // Accesses score again
}

playGame();
```

In this example:

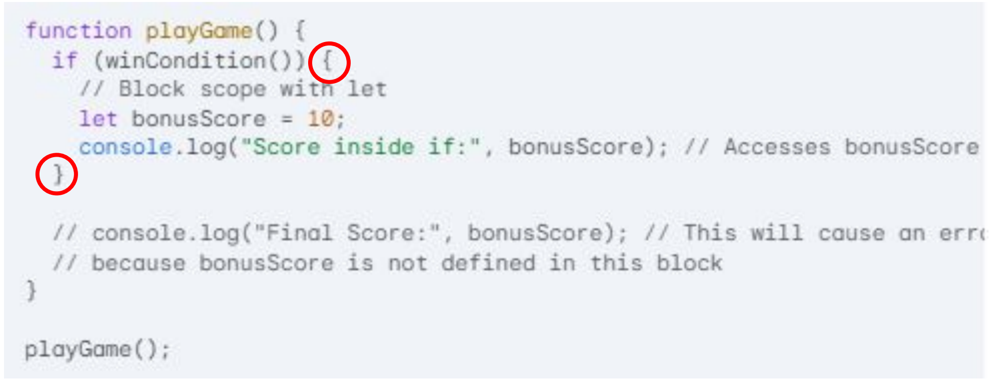
- `score` is declared with `var`, giving it function scope.
- It's accessible and modified within both the `if` block and the main function body.

function: A named chunk of code that performs a specific task.

Modern JavaScript does not use the keyword **var.**

Use **let** or **const** instead.

- Variables declared with `let` and `const` are only accessible within the specific code block where they are declared (e.g., `if` statement, loop, etc.).
- This provides more precise control over variable visibility and prevents accidental modifications from outside the block.



```
function playGame() {
  if (winCondition()) {
    // Block scope with let
    let bonusScore = 10;
    console.log("Score inside if:", bonusScore); // Accesses bonusScore
  }

  // console.log("Final Score:", bonusScore); // This will cause an err
  // because bonusScore is not defined in this block
}

playGame();
```

A code block begins with { and ends with }.

- `bonusScore` is declared with `let` within the `if` block, limiting its scope to that block.
- Trying to access `bonusScore` outside the `if` block will result in an error because it's not defined in that scope.

Block scope (using **let** and **const**)