# **The Differences Between let and var**

A var variable can be redeclared *and* updated.

A let variable be be updated but *not* redeclared.

An example of trying to redeclare a let variable:// In editor:<script>  
let points = 50;  
let points = 60;  
</script>// In the console I get an error:  
Uncaught SyntaxError: Identifier 'points' has already been declared

However, I can update it:

let points = 50;  
points = 60;// In console:  
points// Returns:  
60

If I declare a let variable at the global scope, then redeclare it within a block (curly brackets) as in the example below, I will not get an error in the console, but it will not actually redeclare let.

**// This 'let' is scoped to the window (globally):**  
**let winner = false;**if(points > 40) {  
// This 'let' is scoped to the block (between the curly brackets):  
 let winner = true;  
}

In the above example if we change bothletvariables to var, then call winner in the console it returns true because it is *not* inside a function. (Remember: var is function scoped.) The var variable is being redeclared within its scope, which is the window in this case.

# **The Differences Between let and const**

const variables cannot be updated. let variables are *made* to be updated.

// If I define the const variable:  
const key = 'xyz123';// Then try to redeclare it:  
key = 'xyz1234'// I get the following error:  
Uncaught TypeError: Assignment to constant variable.

There is an interesting caveat to this, though. If I create a const variable that is an object, the attributes of that object can be updated.