

Variables Lab

In this lab we're going to work with variable scopes and hoisting.

Hoisting with var

1. Create a new file called variables.js. Include it in your web page.
2. Add this function to it:

```
export function showHoisting() {  
  var x = 1;  
  if (x === 2) {  
    var y = true;  
  }  
  return y;  
}
```

3. Call this function from the web page and console.log() its output. What is returned? Was that expected or not? Why did it happen? But wait! Wouldn't most languages have a compile time failure on the return statement? I mean, y is local to the block, right?

Working with let and const

4. Now change the vars to lets.
5. Rerun your test. What happened this time? Can you explain its behavior?
6. Go ahead and make the test run. Any way you like is fine.
7. Lastly, change the vars/lets to consts. Depending on how you chose to fix your test in the last step, it may or may not still run. Again think about what might have changed.
8. Try to reassign the y const. (Should not work).

Are consts constant or not?

9. Create a new const:
`const z = {foo: true, bar: true, baz: true};`
10. Your tests should run fine. Make sure.
11. Now on the next lines of code go:
`z.foo = false;`
`z.qux = true;`
12. z is a const, so this should fail, right? Wrong! Run your tests and they'll pass. Discuss with your teammate why a const is allowed to change.