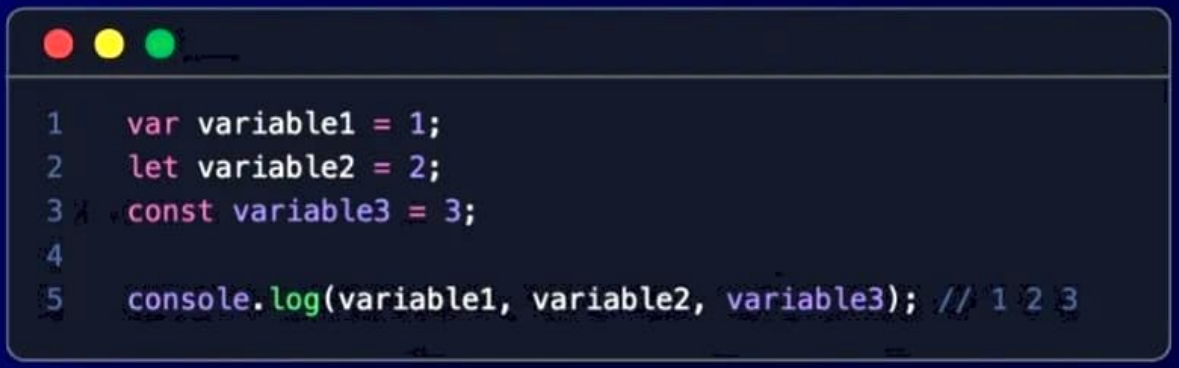
var - let - const

(scope, hoisting, everything you should know)



```
var variable1 = 1;
let variable2 = 2;
const variable3 = 3;

console.log(variable1, variable2, variable3); // 1 2 3
```

Scope is accessibility (visibility) of variables.

Hoisting is a JavaScript mechanism where variables and function declarations are moved to the top of their scope before code execution.

JS

	var	let	const
scope	global/function	global/block	global/block
re-declaring	+	-	-
updateing	+	+	-
hoisting	+	+	+
default initialization	+ (as undefined)	-	-

JS

SCOPE types

```
1   // global scope
2   var globalVar = "global var";
3   let globalLet = "global let";
4   const globalConst = "global const";
5
6   function fn() {
7     // function scope
8     var functionVar = "function var";
9     let functionLet = "function let";
10    const functionConst = "function const";
11  }
12
13  if (true) {
14    // block scope
15    let blockLet = "block let";
16    const blockConst = "block const";
17  }
```

JS

Updating and Re-declaring

```
1   var testVar = 1;
2   testVar = 2; // updating
3   var testVar = 3; // re-declaring
4
5   let testLet = 1;
6   testLet = 2; // updating
7   let testLet = 3; // re-declaring
8   // TypeError: Identifier 'testLet' has already been declared
9
10  const testConst = 1;
11  testConst = 2; // updating
12  // TypeError: Assignment to constant variable.
13  const testConst = 3; // re-declaring
14  // SyntaxError: Identifier 'testConst' has already been declared
```

JS

Hoisting

```
1   console.log(var1); // undefined
2   var var1 = "var";
3   console.log(let1);
4   // ReferenceError: Cannot access 'let1' before initialization
5   let let1 = "let";
6   console.log(const1);
7   // ReferenceError: Cannot access 'const1' before initialization
8   const const1 = "const";
```

JS