Web Engineering
Lect 11 (JavaScript)

Instructor: Faheem Shaukat

Meeting Hours

Wednesday and Thursday

12PM to 02PM

JavaScript

- JavaScript, often abbreviated as JS.
- It is a programming language that conforms to the ECMAScript specification.
- JavaScript is high-level, often just-in-time compiled and multi-paradigm.
- It has curly-bracket syntax, dynamic typing, prototype-based object-orientation and first-class functions.

What is JavaScript?

What can you do with it?

Where does JavaScript code run?

JavaScript vs ECMAScript?

stackoverflow

Most Popular Technologies

Programming, Scripting, and Markup Languages

All Respondents | Professional Developers

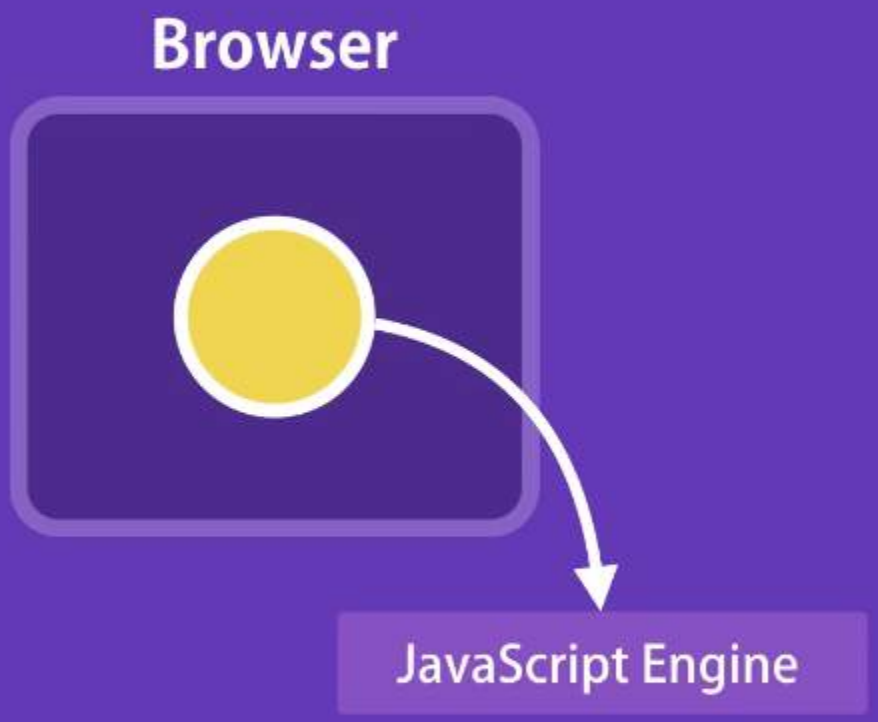
JAVASCRIPT TODAY

Web / Mobile Apps

Real-time Networking Apps

Command-line Tools

Games



FireFox: SpiderMonkey

Chrome: v8

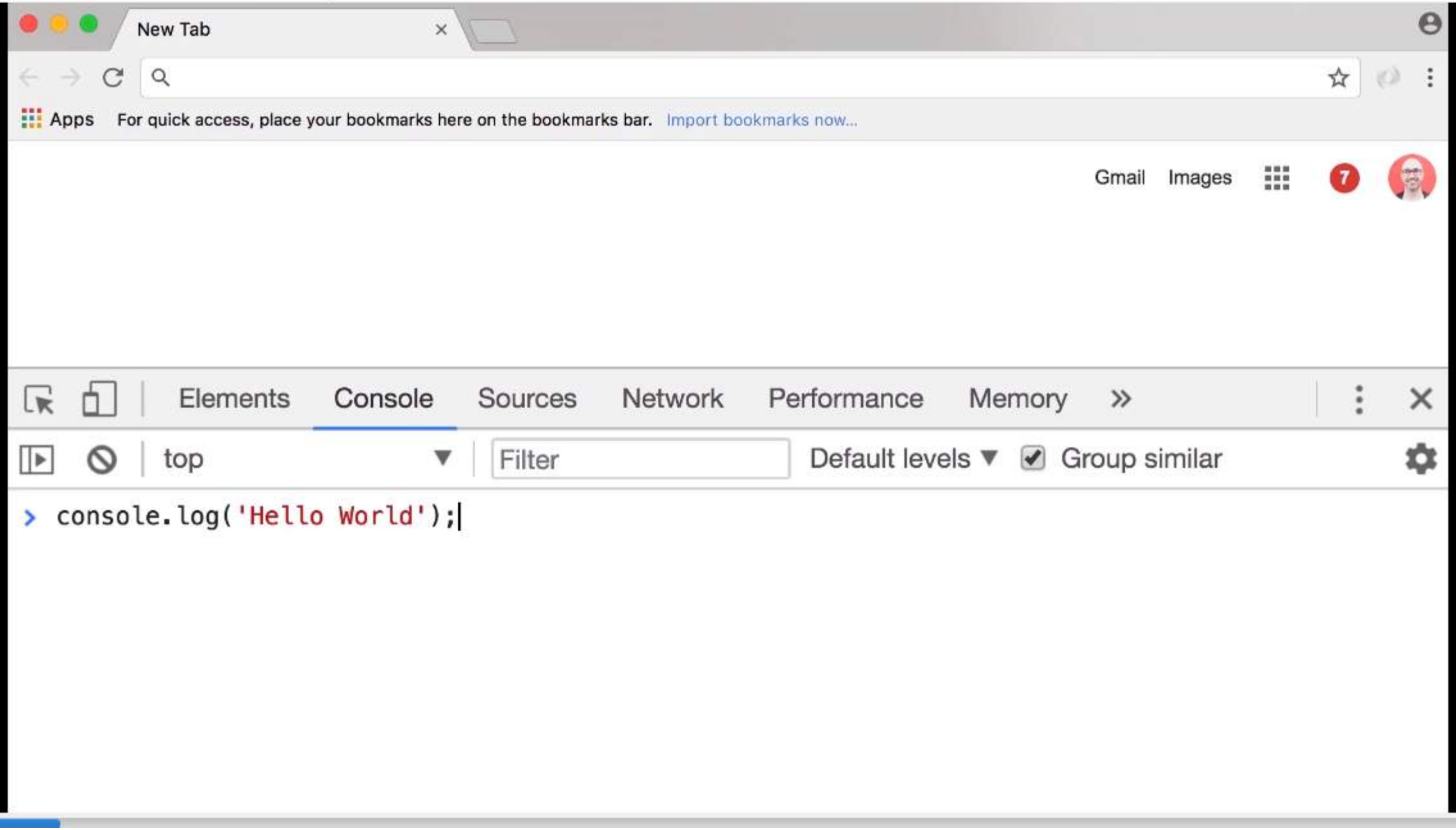
ECMAScript

Specification

JavaScript

Programming Language


v1
ES2015/ES6
1997
2015
2016
…


New Tab
Apps
For quick access, place your bookmarks here on the bookmarks bar.
Import bookmarks now...
Gmail
Images
7
Elements
Console
Sources
Network
Performance
Memory
top
Filter
Default levels
Group similar
> console.log('Hello World');


index.html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
</head>
<body>
<h1>Hello World</h1>
<script>
console.log('Hello World');
</script>
</body>
</html>
0 0
Port : 5500
Ln 12, Col 32 (27 selected)
Spaces: 2
UTF-8
LF
HTML

In this lecture, we will discuss…

Defining Variables, Functions and Scope

Variables

```
var message = "hi";
```

- Variable definition should always start with ‘var’
- No types are declared
  - JS is dynamically typed language
  - Same variable can hold different types during the life of the execution

Functions

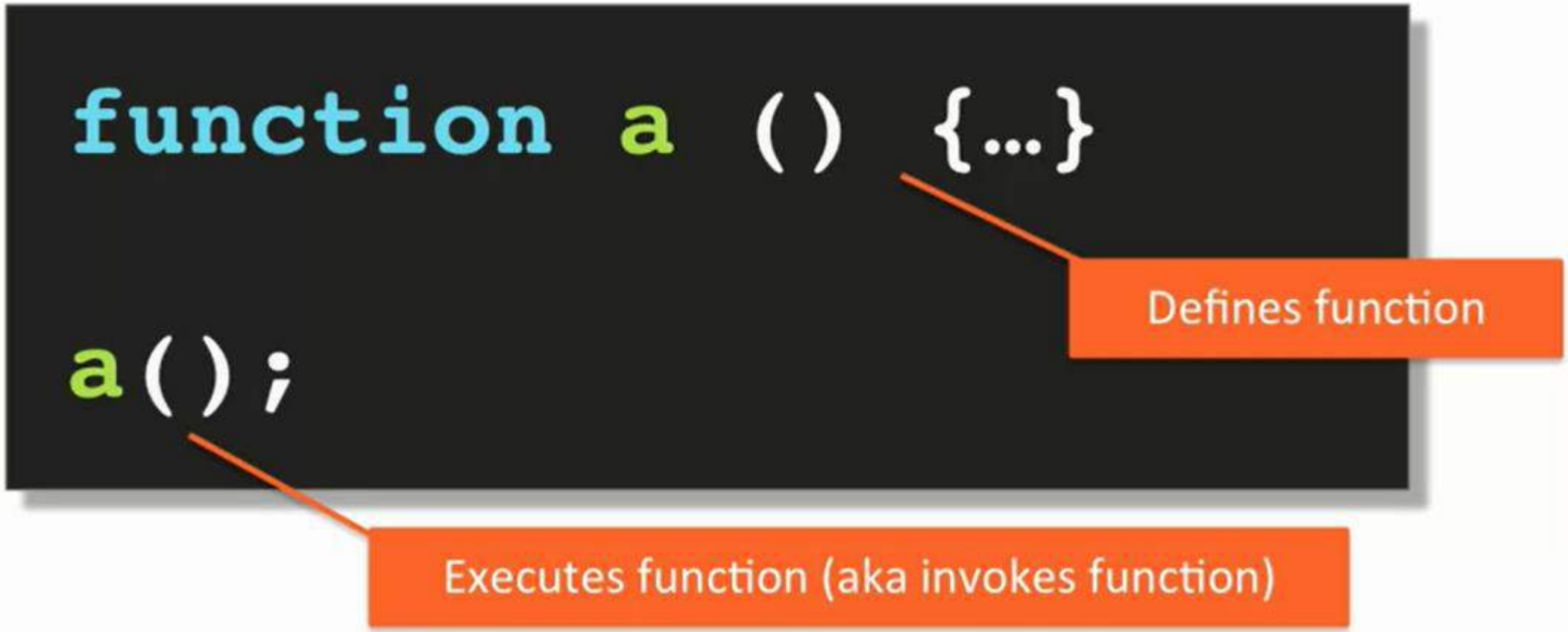
```
function a () {…}
```

```
var a = function () {…}
```

Value of function assigned,
NOT the returned result!

No name defined

Functions


function a () {…}
a();
Defines function
Executes function (aka invokes function)

Functions

Arguments defined without ‘var’

```
function compare (x, y) {
   return x > y;
}
```

Functions

```
function compare (x, y) {…}
var a = compare(4, 5);
compare(4, "a");
compare();
```

ALL LEGAL

Scope

Global

- Variables and functions defined here are available everywhere

Function
aka lexical

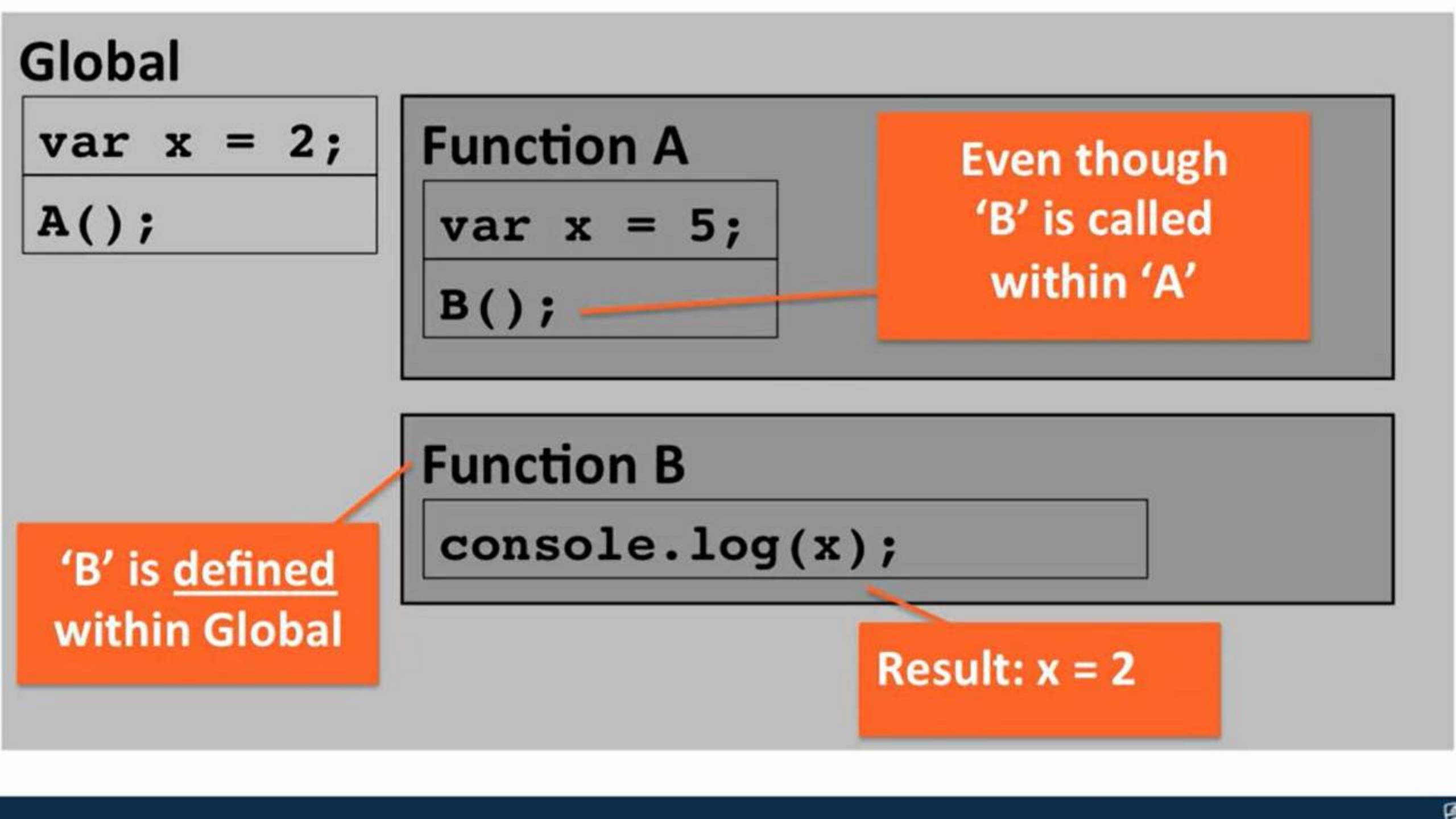
- Variables and functions defined her
  are available only within this function

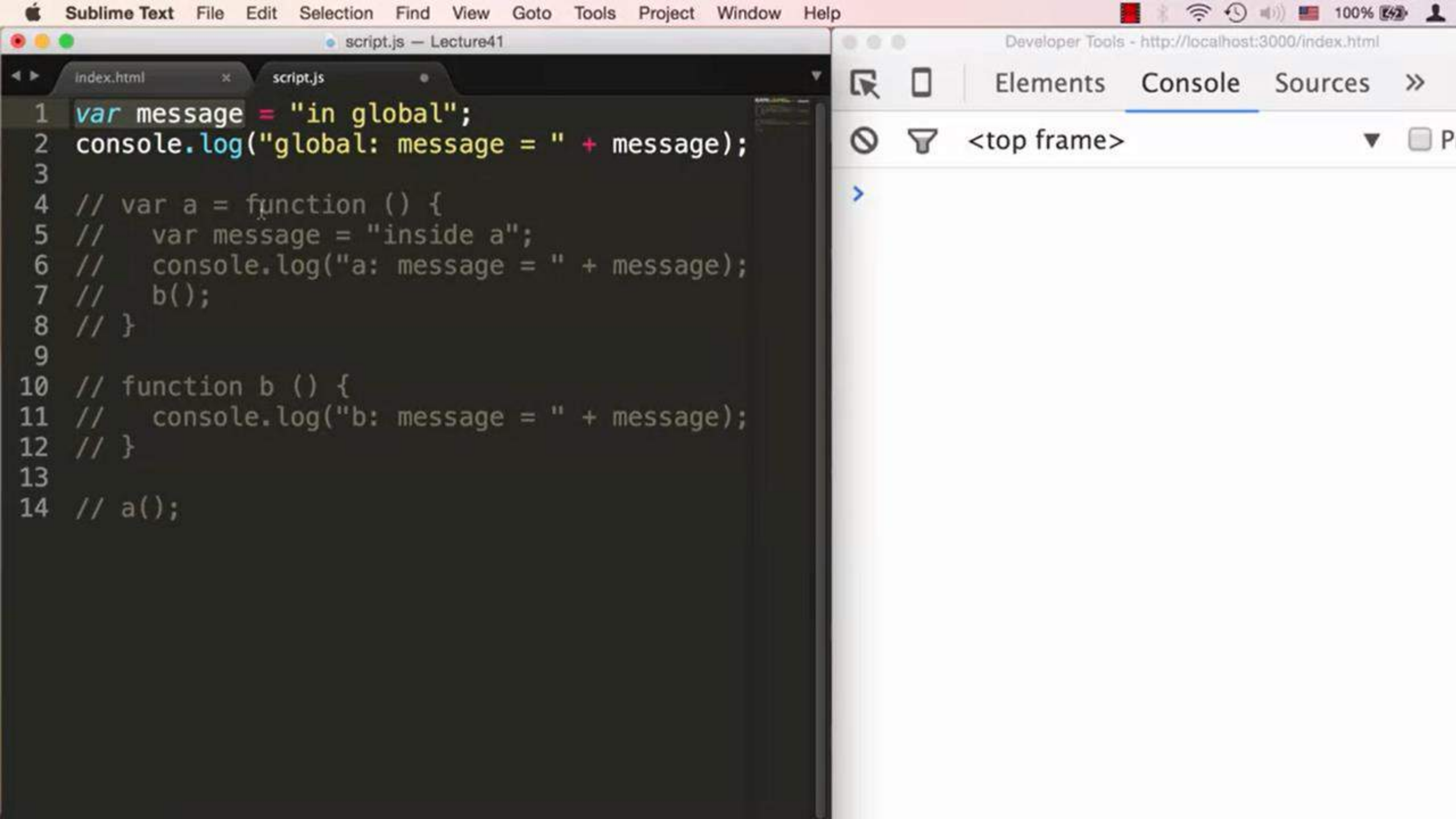
Scope Chain

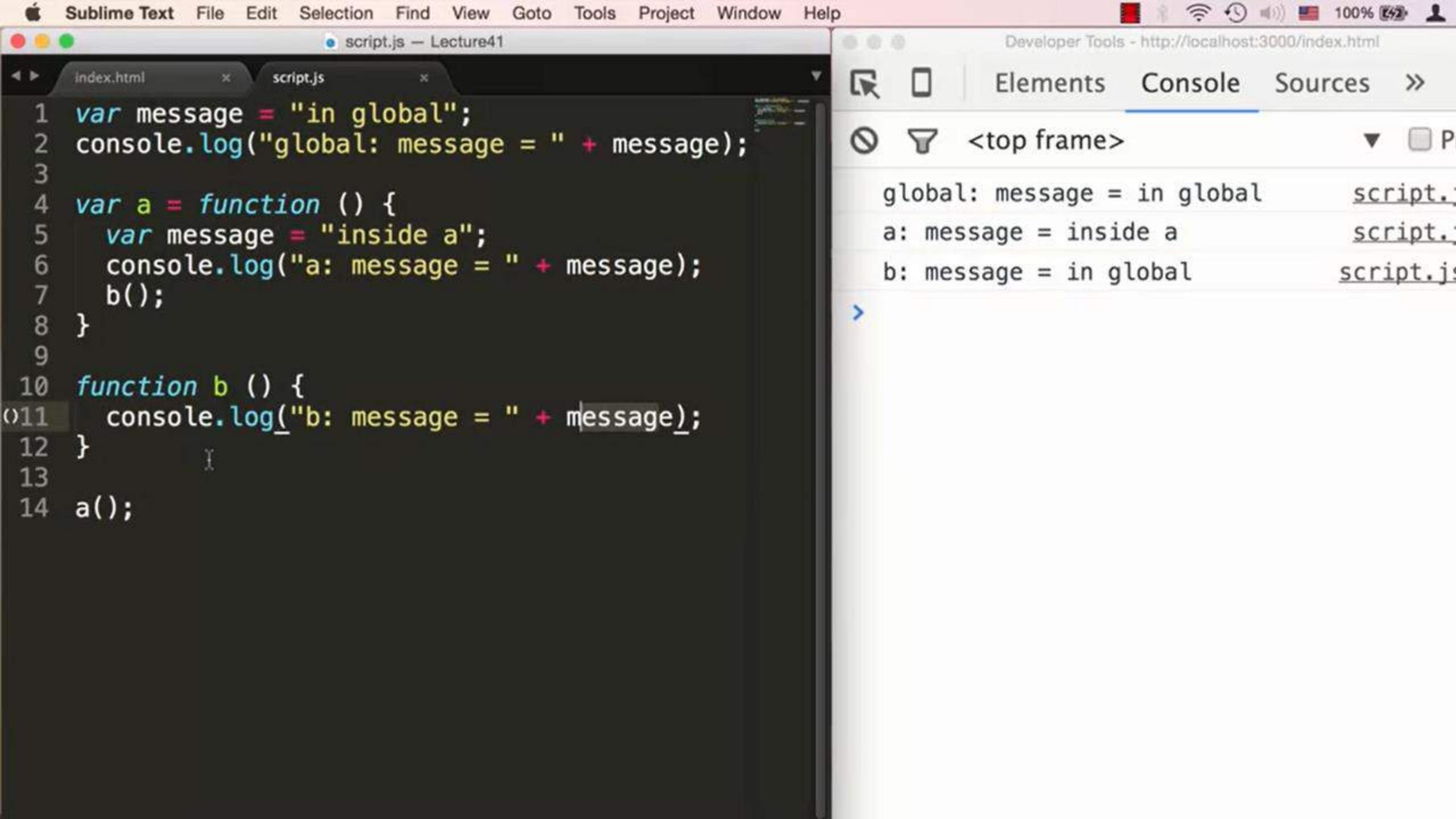
- Everything is executed in an *Execution Context*
- Function invocation creates a new *Execution Context*
- Each *Execution Context* has:
  - Its own *Variable Environment*
  - Special '`this`' object
  - Reference to its *Outer Environment*
- Global scope does not have an *Outer Environment* as it's the most outer there is

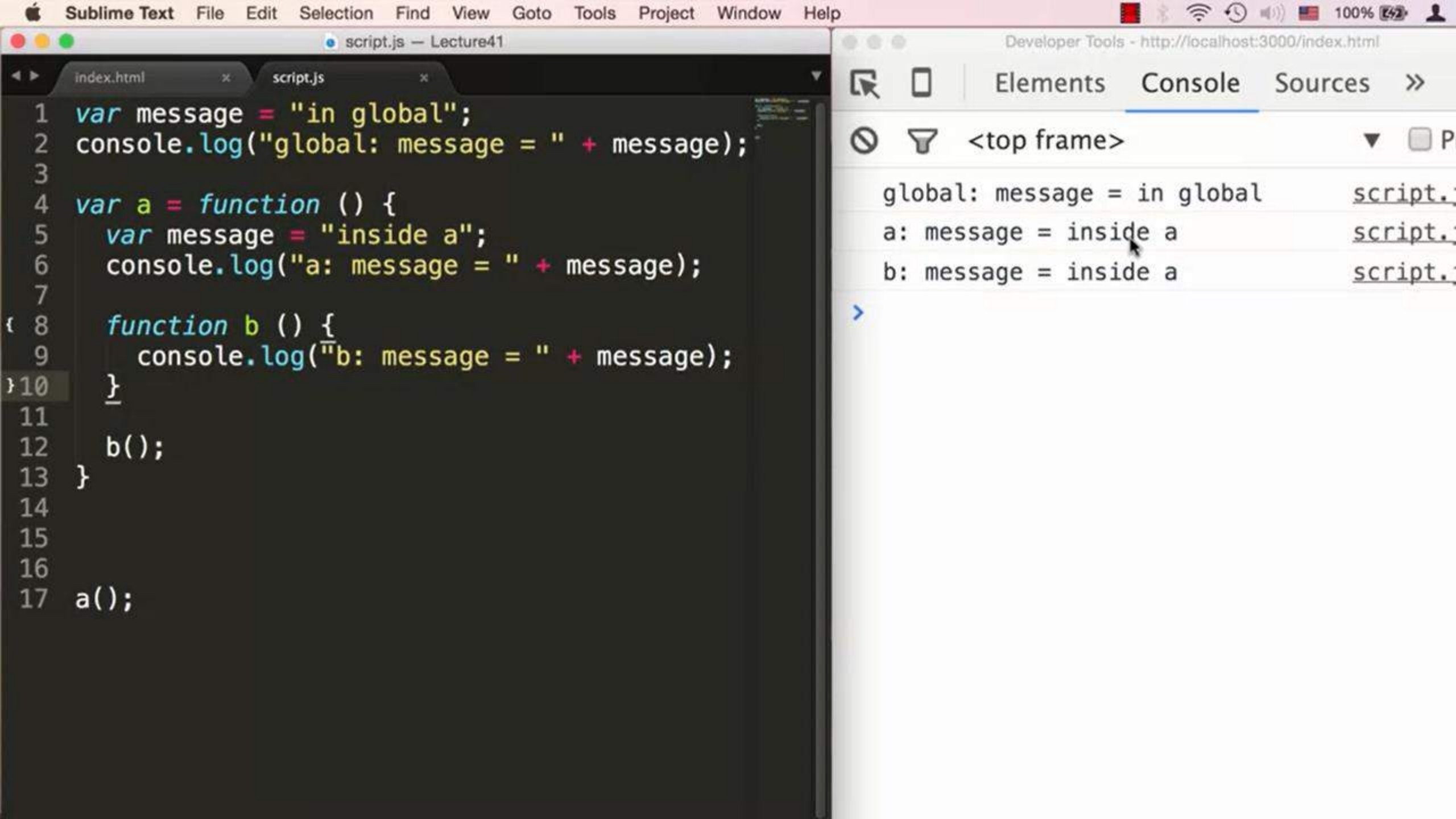
Scope Chain

Referenced (not defined) variable will be searched for in its current scope first. If not found, the Outer Reference will be searched.
If not found, the Outer Reference's Outer Reference will be searched, etc.
This will keep going until the Global scope.
If not found in Global scope, the variable is `undefined`.


Global
var x = 2;
A();
Function A
var x = 5;
B();
Even though 'B' is called within 'A'
Function B
console.log(x);
'B' is defined within Global
Result: x = 2


Sublime Text File Edit Selection Find View Goto Tools Project Window Help
100%
script.js — Lecture41
index.html
script.js
var message = "in global";
console.log("global: message = " + message);
// var a = function () {
//   var message = "inside a";
//   console.log("a: message = " + message);
//   b();
// }
// function b () {
//   console.log("b: message = " + message);
// }
// a();
Developer Tools - http://localhost:3000/index.html
Elements
Console
Sources
<top frame>


Sublime Text File Edit Selection Find View Goto Tools Project Window Help
100%
script.js — Lecture41
index.html
script.js
var message = "in global";
console.log("global: message = " + message);
var a = function () {
var message = "inside a";
console.log("a: message = " + message);
b();
}
function b () {
console.log("b: message = " + message);
}
a();
Developer Tools - http://localhost:3000/index.html
Elements
Console
Sources
<top frame>
global: message = in global
a: message = inside a
b: message = in global


Sublime Text File Edit Selection Find View Goto Tools Project Window Help
100%
script.js — Lecture41
index.html
script.js
var message = "in global";
console.log("global: message = " + message);
var a = function () {
var message = "inside a";
console.log("a: message = " + message);
function b () {
console.log("b: message = " + message);
}
b();
}
a();
Developer Tools - http://localhost:3000/index.html
Elements
Console
Sources
<top frame>
global: message = in global
a: message = inside a
b: message = inside a
script.
script.
script.

Summary

- ✧ Defining variables – dynamically typed
- ✧ Defining functions – creates its own scope (lexical)
- ✧ JS code runs within an Execution Context
- ✧ Scope chain is used to retrieve variables from Outer Variable Environments

In this lecture, we will discuss…

Javascript Types

Types

A type is a particular data structure.

- ✧ Each language defines some built-in types
- ✧ Built-in types can be used to build other data structures
- ✧ JS has 7 built-in types: 6 primitive and 1 Object type

Object Type

Object is a collection of name/value pairs

Object Type

Person Object

```
firstName: "Yaakov",
lastName:  "Chaikin",
social: {
    linkedin : "yaakovchaikin",
    twitter: "yaakovchaikin",
    facebook: "CourseraWebDev"
}
```

name

value

Primitive Types

Primitive type represents a *single, immutable* value

- ✧ **Single value, i.e., *not* an object**
- ✧ **Immutable means once it's set, it can't be changed**
  - **Value becomes read-only**
  - **You can create another value based on an existing one**

Primitive Type: Boolean

Boolean can only have 2 values: `true` or `false`

Primitive Type: Undefined

Undefined signifies that no value has ever been set

- Can only have one value: `undefined`
- You *can* set a variable to `undefined`, but you *should NEVER do it*
  - Its meaning is that it's never been defined, so defining it to `undefined` is counter to its core meaning

Primitive Type: Null

Null signifies lack of value

- As opposed to `undefined`, which is lack of definition
- Can only have one value: `null`
- It's ok to explicitly set a variable to `null`

Primitive Type: Number

Number is the only numeric type in Javascript

- ✧ **Always represented under the hood as double-precision 64-bit floating point**
- ✧ **JS does *not* have an integer type**
  - **Integers are a subset of doubles instead of a separate data type**

Primitive Type: String

String is sequence of characters used to represent text

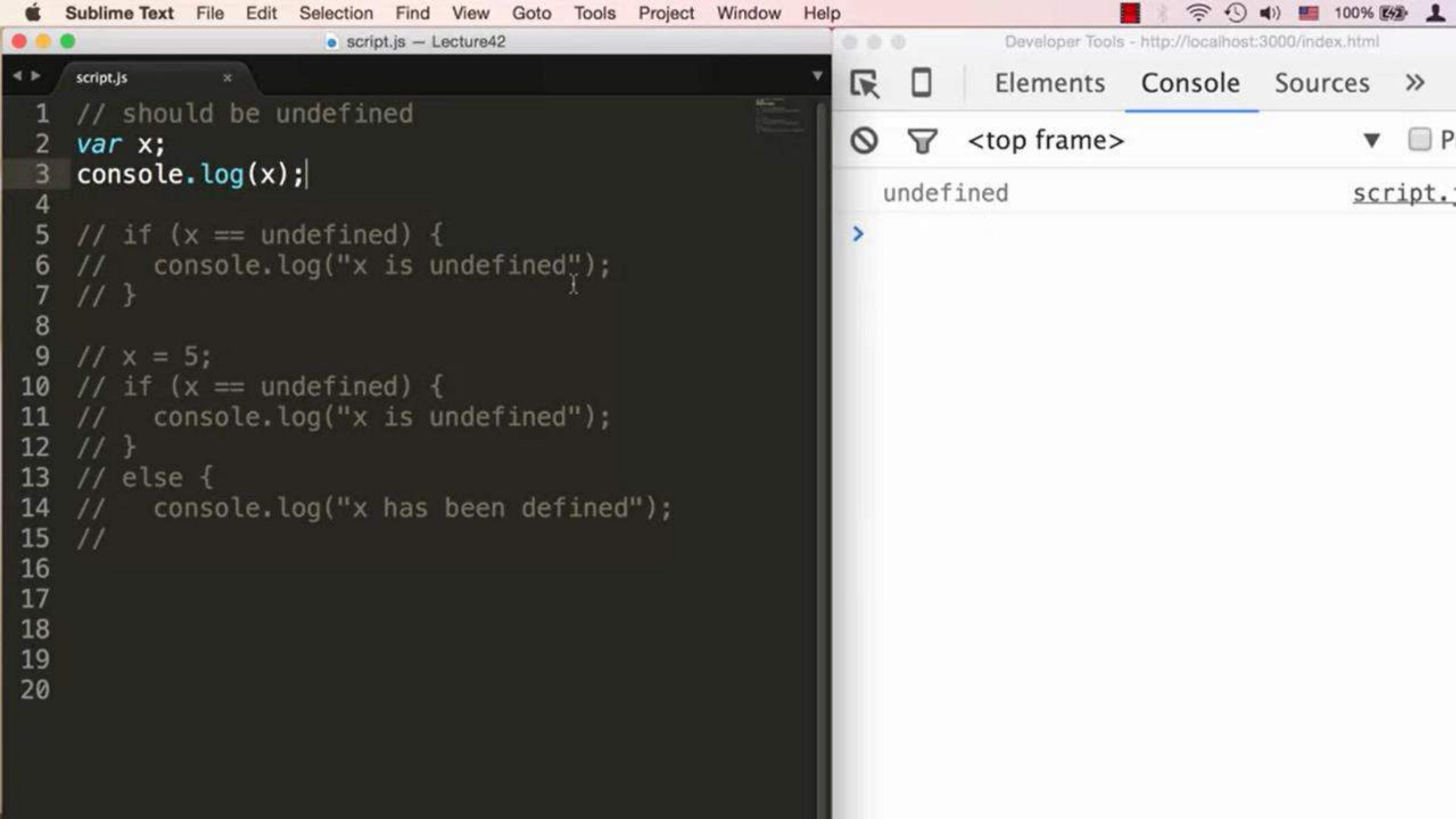
- ✧ **Use either single or double quotes, i.e., ‘text’ or “text”**

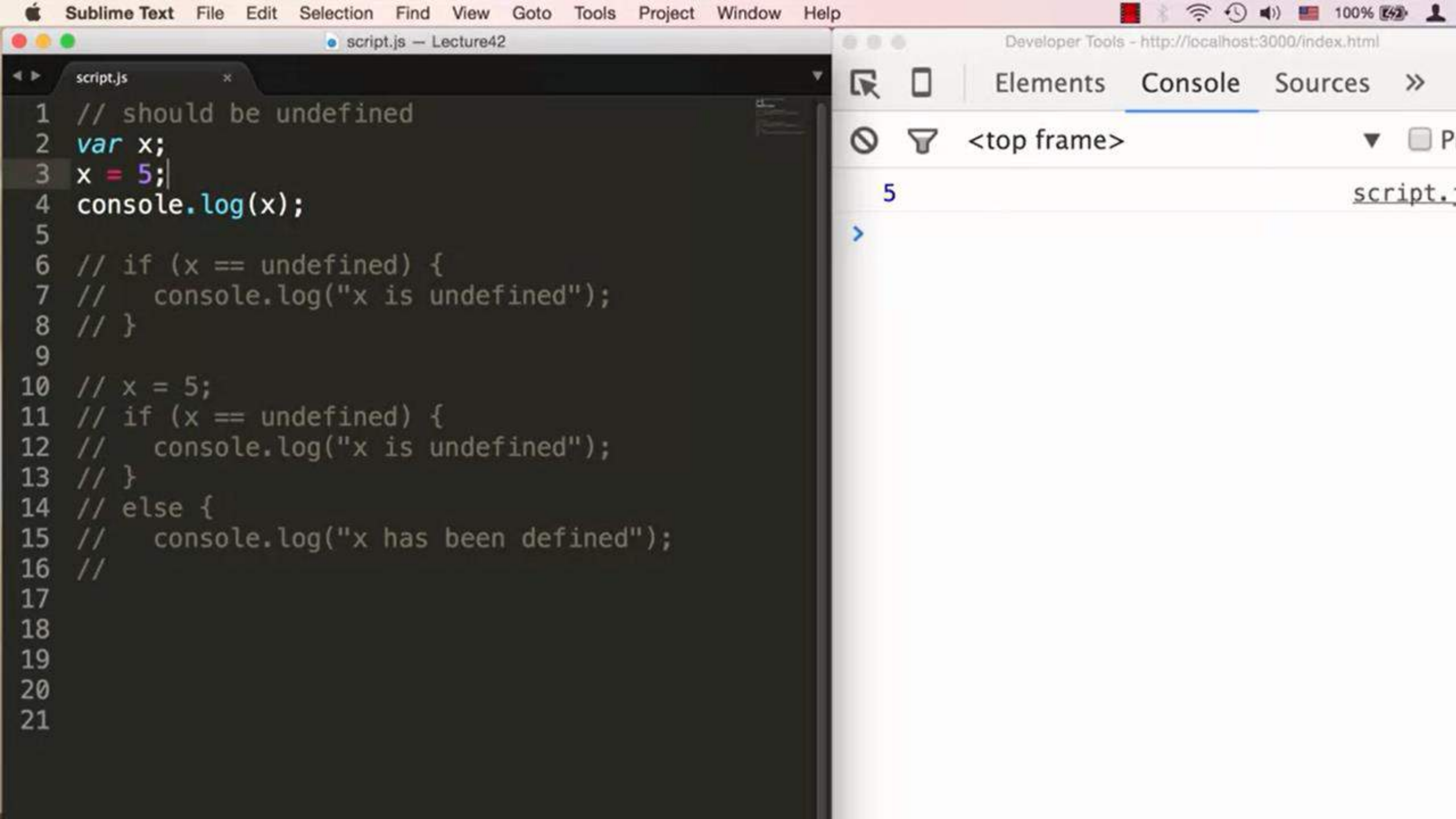
Primitive Type: Symbol

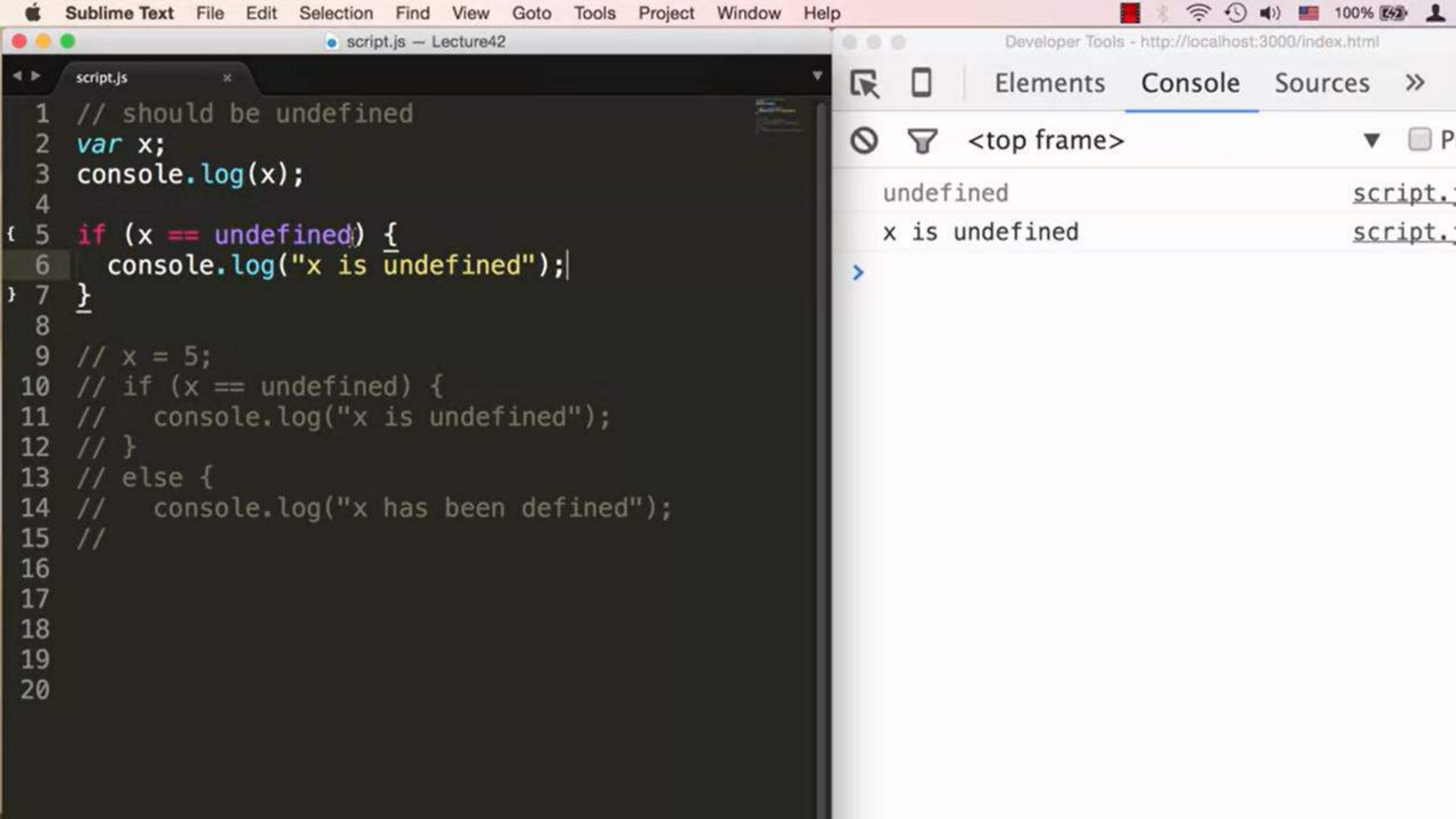
Symbol is new to ES6

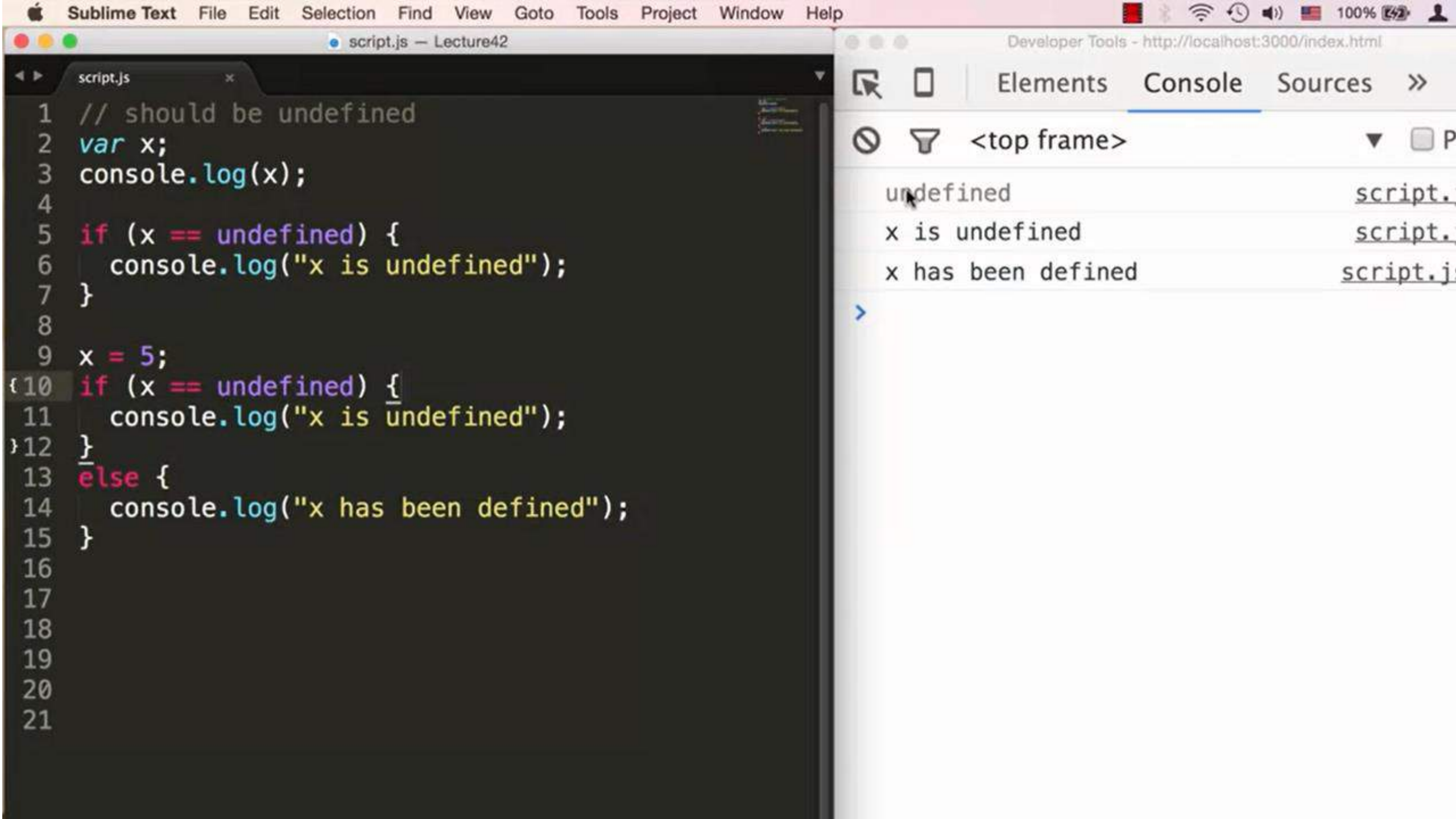
Not covered in this class

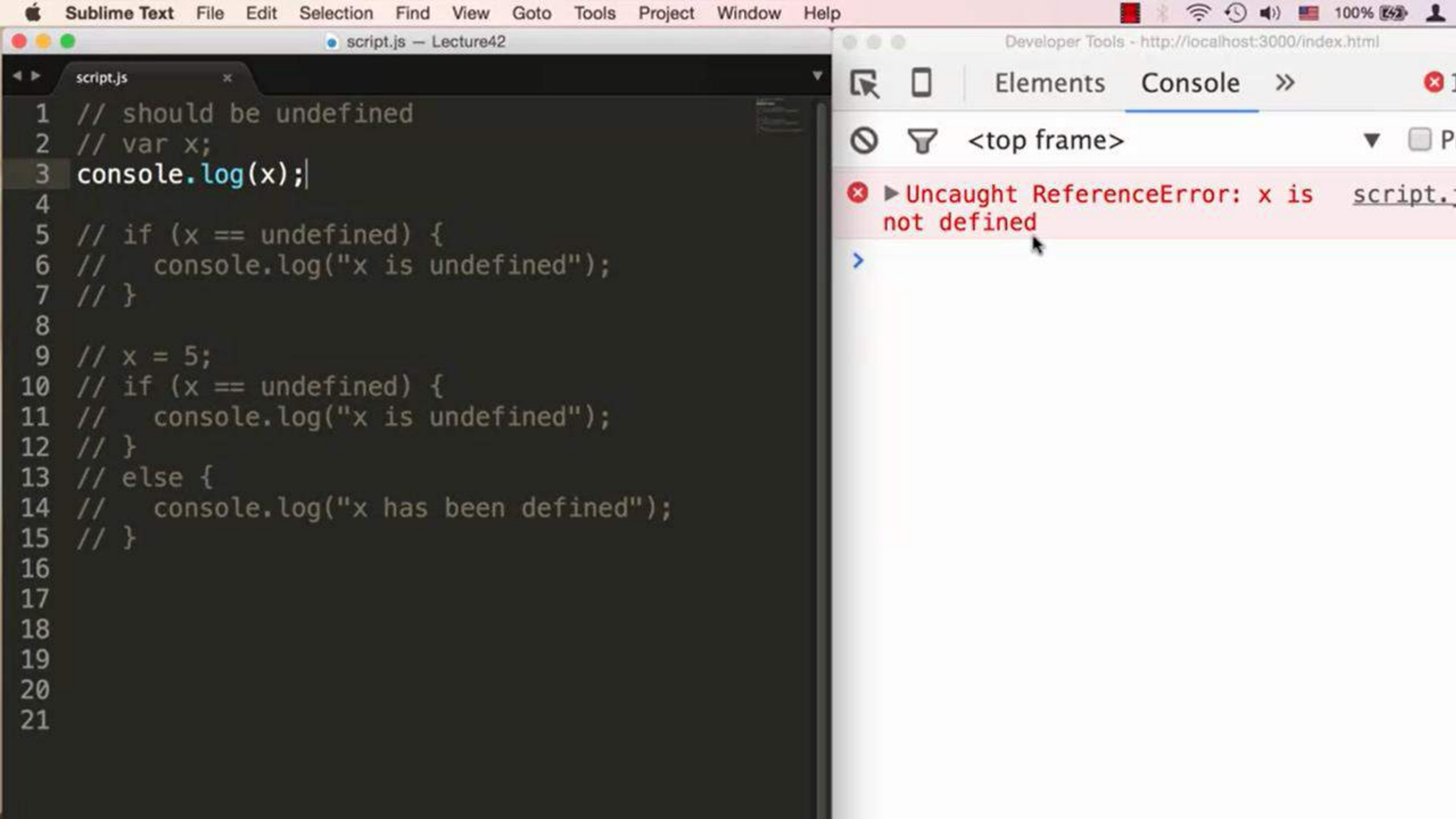
- ES6 (released 2015) isn't widely supported or used yet


Sublime Text File Edit Selection Find View Goto Tools Project Window Help
100%
script.js — Lecture42
script.js
1 // should be undefined
2 var x;
3 console.log(x);
4
5 // if (x == undefined) {
6 //   console.log("x is undefined");
7 // }
8
9 // x = 5;
10 // if (x == undefined) {
11 //   console.log("x is undefined");
12 // }
13 // else {
14 //   console.log("x has been defined");
15 //
16
17
18
19
20
Developer Tools - http://localhost:3000/index.html
Elements
Console
Sources
<top frame>
undefined
script.
>


Sublime Text File Edit Selection Find View Goto Tools Project Window Help
100%
script.js — Lecture42
script.js
// should be undefined
var x;
x = 5;
console.log(x);
// if (x == undefined) {
//   console.log("x is undefined");
// }
// x = 5;
// if (x == undefined) {
//   console.log("x is undefined");
// }
// else {
//   console.log("x has been defined");
//
Developer Tools - http://localhost:3000/index.html
Elements
Console
Sources
<top frame>
5
script.
>

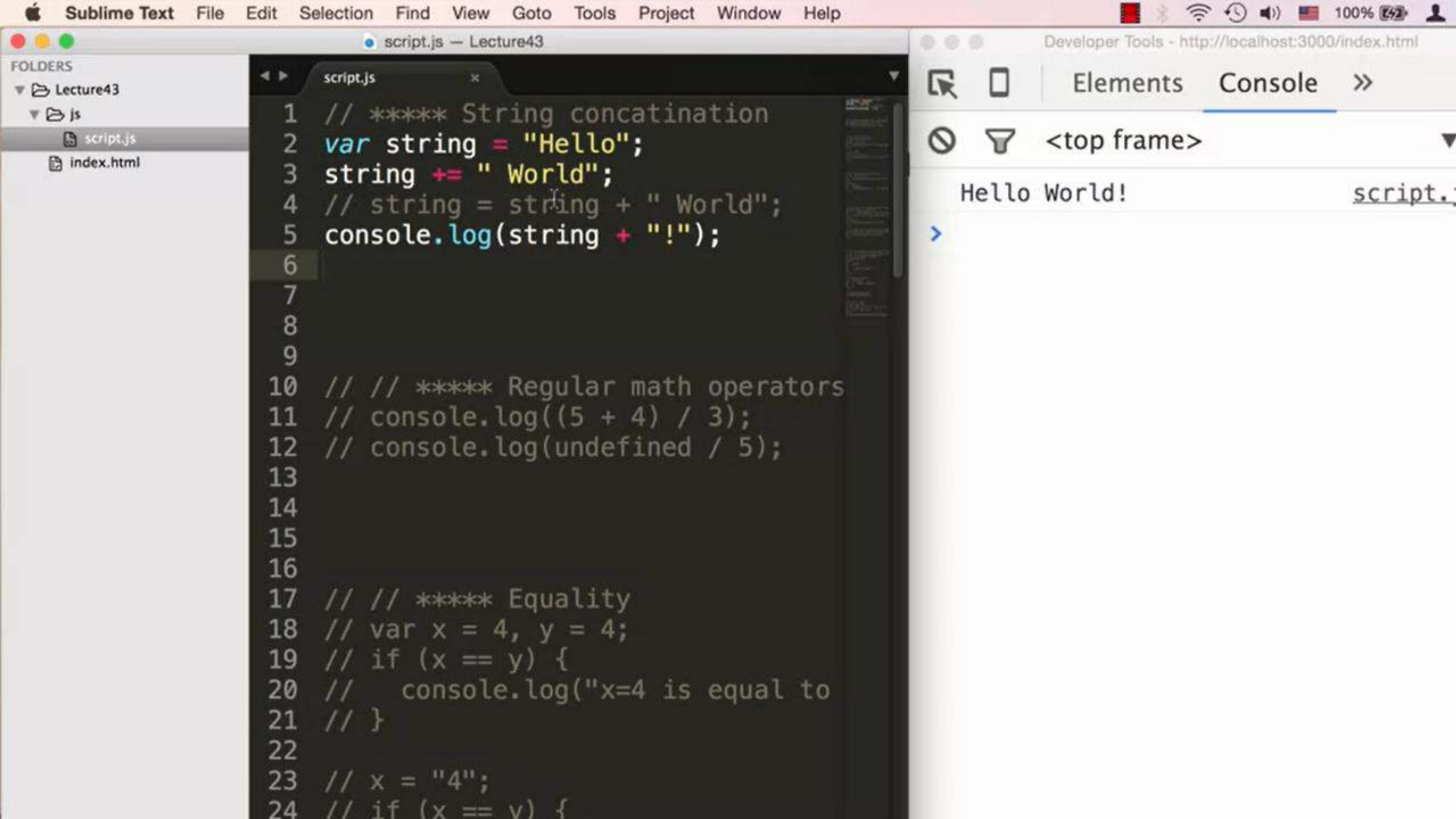

Sublime Text File Edit Selection Find View Goto Tools Project Window Help
100%
script.js — Lecture42
script.js
1 // should be undefined
2 var x;
3 console.log(x);
4
5 if (x == undefined) {
6 console.log("x is undefined");
7 }
8
9 // x = 5;
10 // if (x == undefined) {
11 // console.log("x is undefined");
12 // }
13 // else {
14 // console.log("x has been defined");
15 //
16
17
18
19
20
Developer Tools - http://localhost:3000/index.html
Elements
Console
Sources
<top frame>
undefined
script.
x is undefined
script.
>

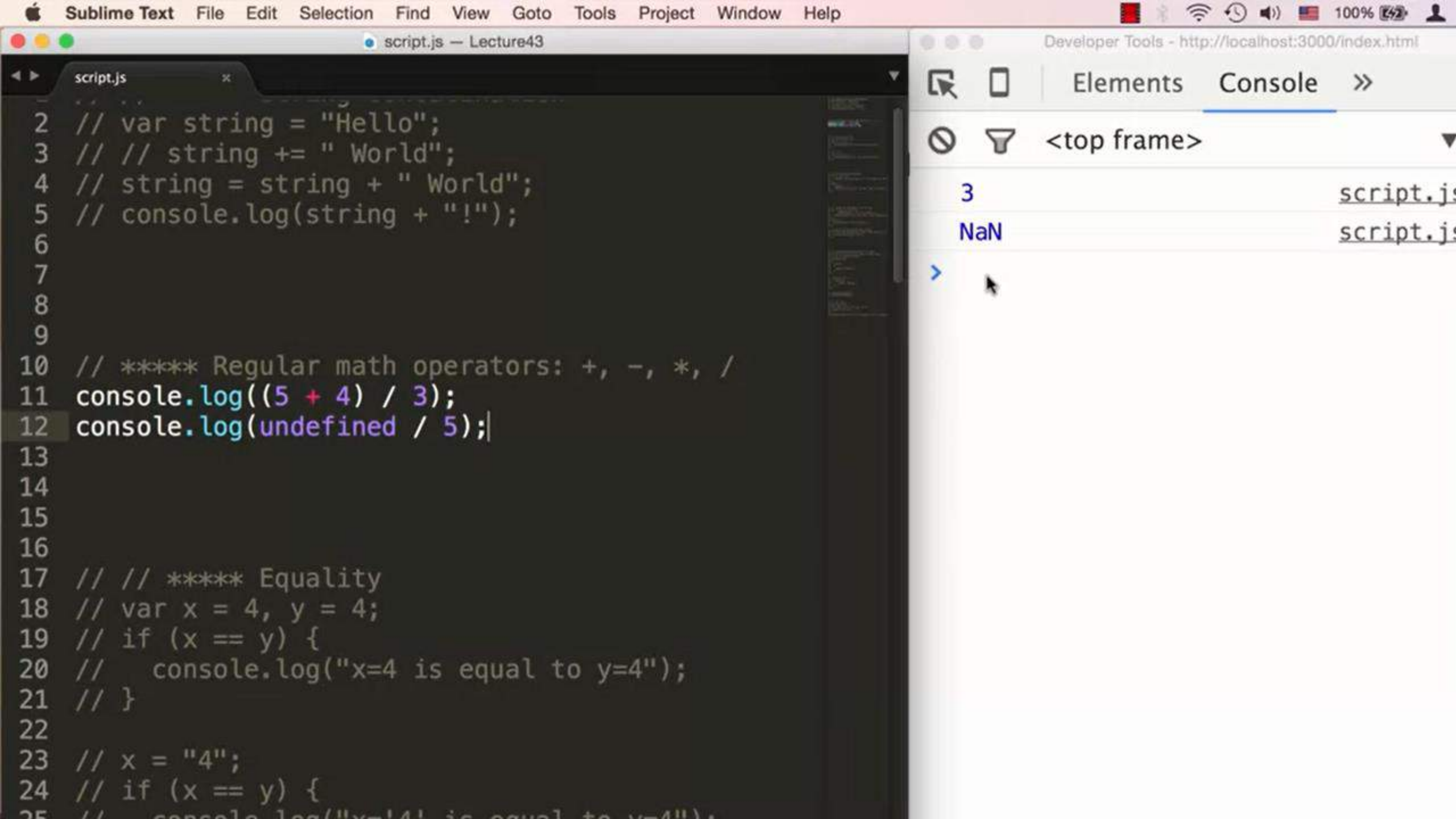

Sublime Text File Edit Selection Find View Goto Tools Project Window Help
100%
script.js — Lecture42
script.js
1 // should be undefined
2 var x;
3 console.log(x);
4
5 if (x == undefined) {
6 console.log("x is undefined");
7 }
8
9 x = 5;
10 if (x == undefined) {
11 console.log("x is undefined");
12 }
13 else {
14 console.log("x has been defined");
15 }
16
17
18
19
20
21
Developer Tools - http://localhost:3000/index.html
Elements Console Sources
<top frame>
undefined script.
x is undefined script.
x has been defined script.j
>

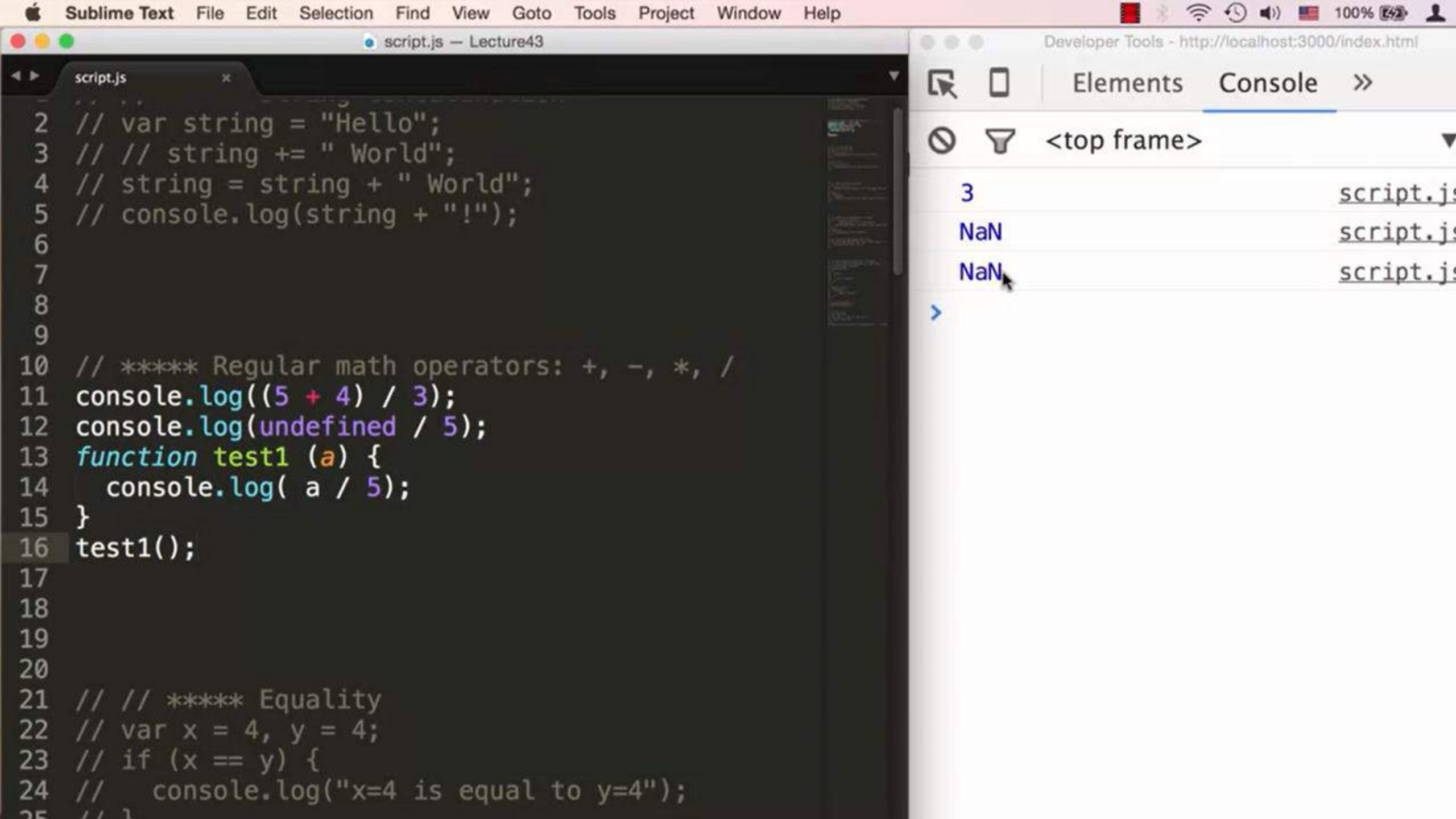

Sublime Text File Edit Selection Find View Goto Tools Project Window Help
100%
script.js — Lecture42
script.js
1 // should be undefined
2 // var x;
3 console.log(x);
4
5 // if (x == undefined) {
6 //   console.log("x is undefined");
7 // }
8
9 // x = 5;
10 // if (x == undefined) {
11 //   console.log("x is undefined");
12 // }
13 // else {
14 //   console.log("x has been defined");
15 // }
16
17
18
19
20
21
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
Uncaught ReferenceError: x is not defined
script.
>

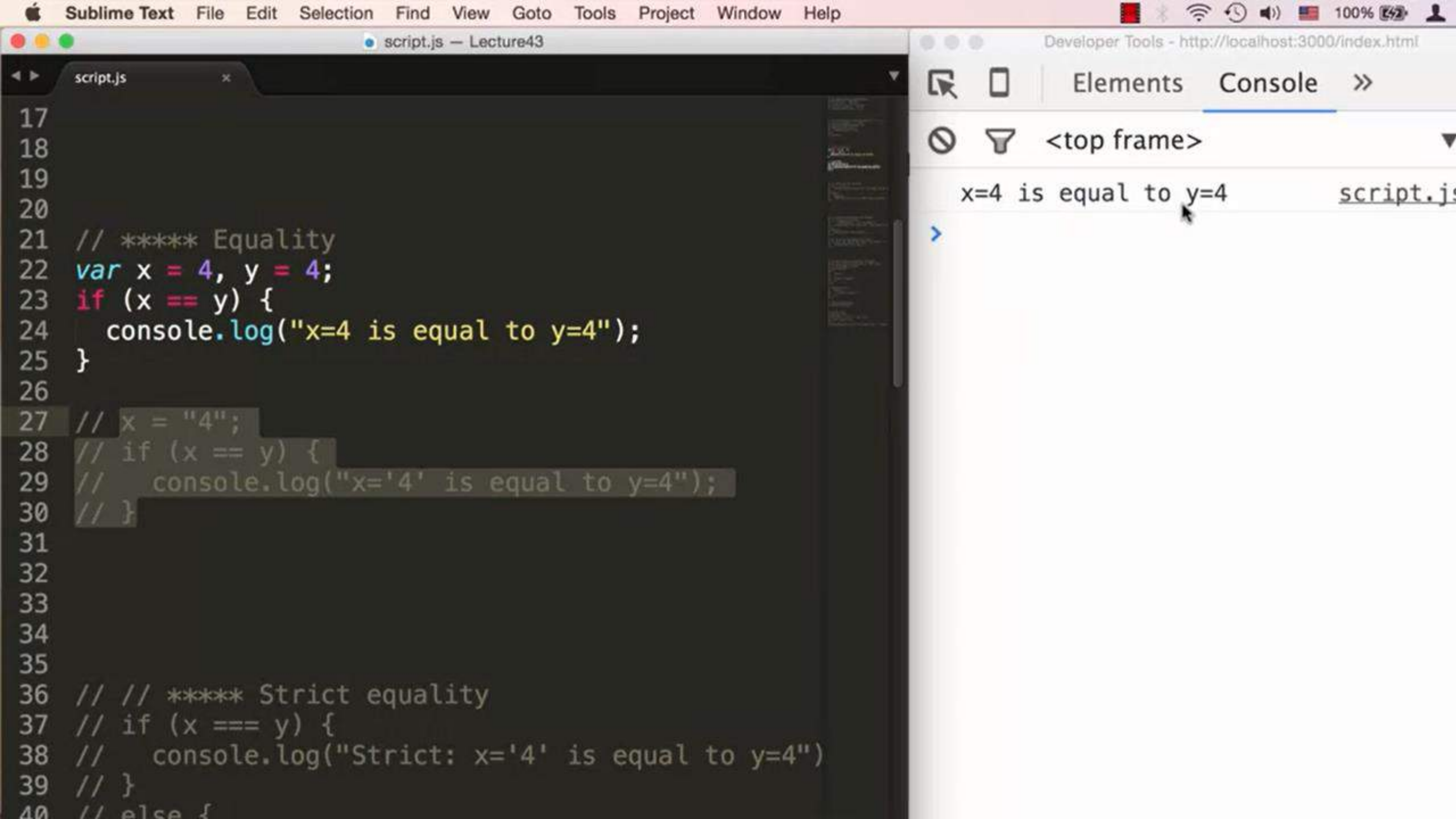
Summary

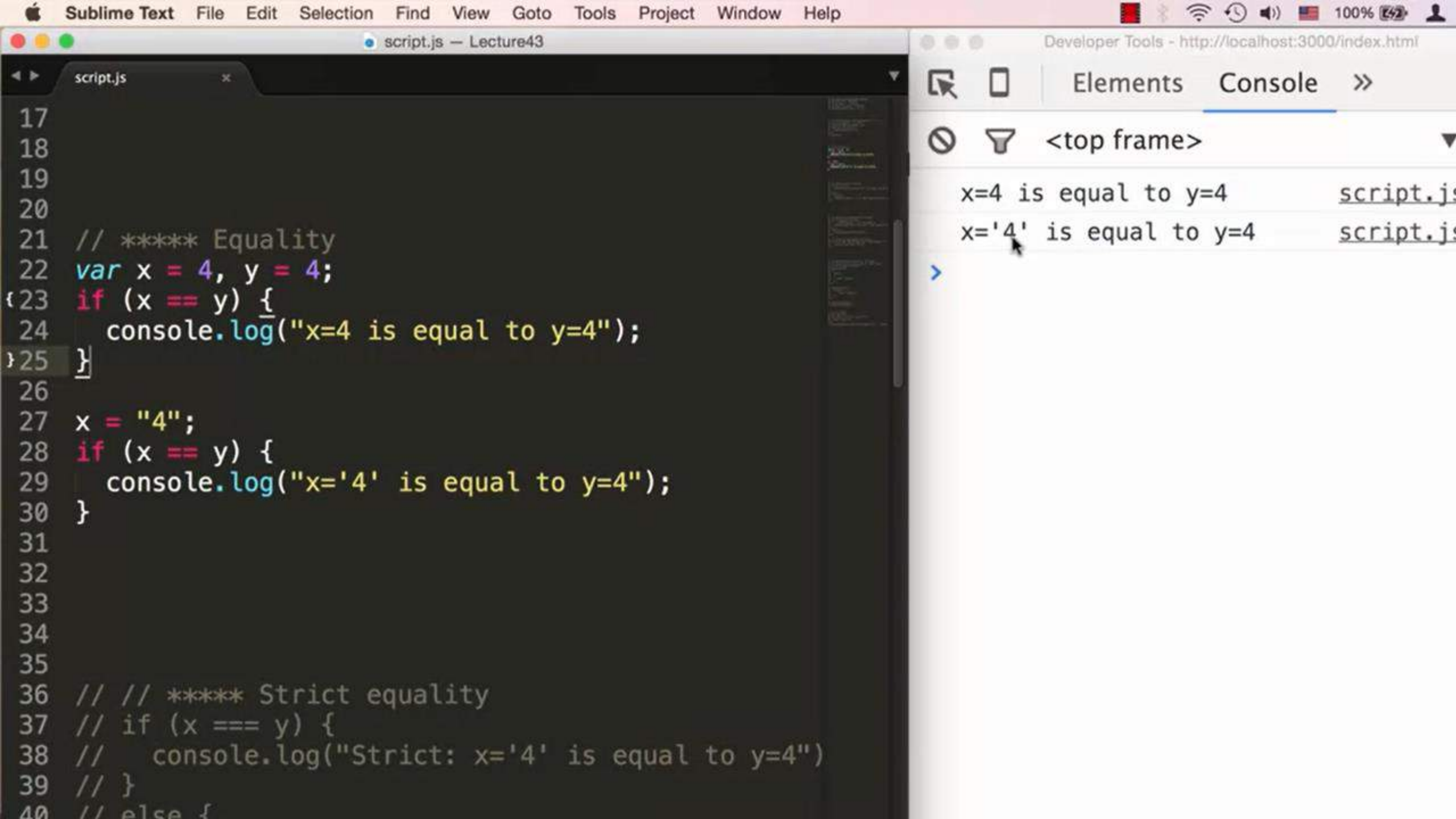
- ✧ Javascript defines 7 built-in types
  - Object and 6 Primitives
- ✧ Object type is a collection of name/value pairs
- ✧ Primitive type can contain a single, immutable value
- ✧ Undefined means variable memory has been allocated but no value has ever been explicitly set yet

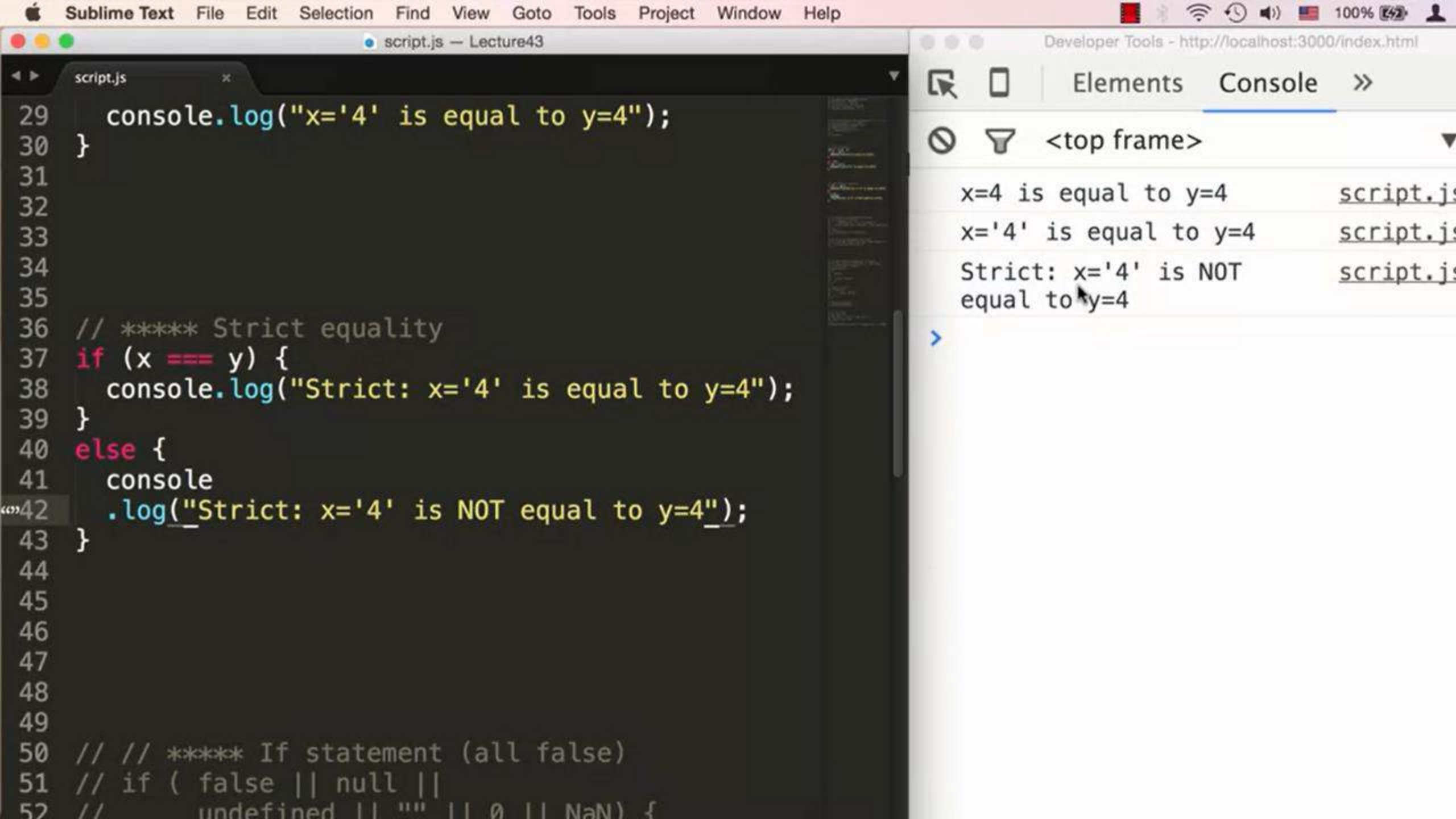

Sublime Text File Edit Selection Find View Goto Tools Project Window Help
script.js — Lecture43
FOLDERS
Lecture43
js
script.js
index.html
script.js
// ***** String concatination
var string = "Hello";
string += " World";
// string = string + " World";
console.log(string + "!");
// // ***** Regular math operators
// console.log((5 + 4) / 3);
// console.log(undefined / 5);
// // ***** Equality
// var x = 4, y = 4;
// if (x == y) {
// console.log("x=4 is equal to
// }
// x = "4";
// if (x == y) {
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
Hello World!
script.

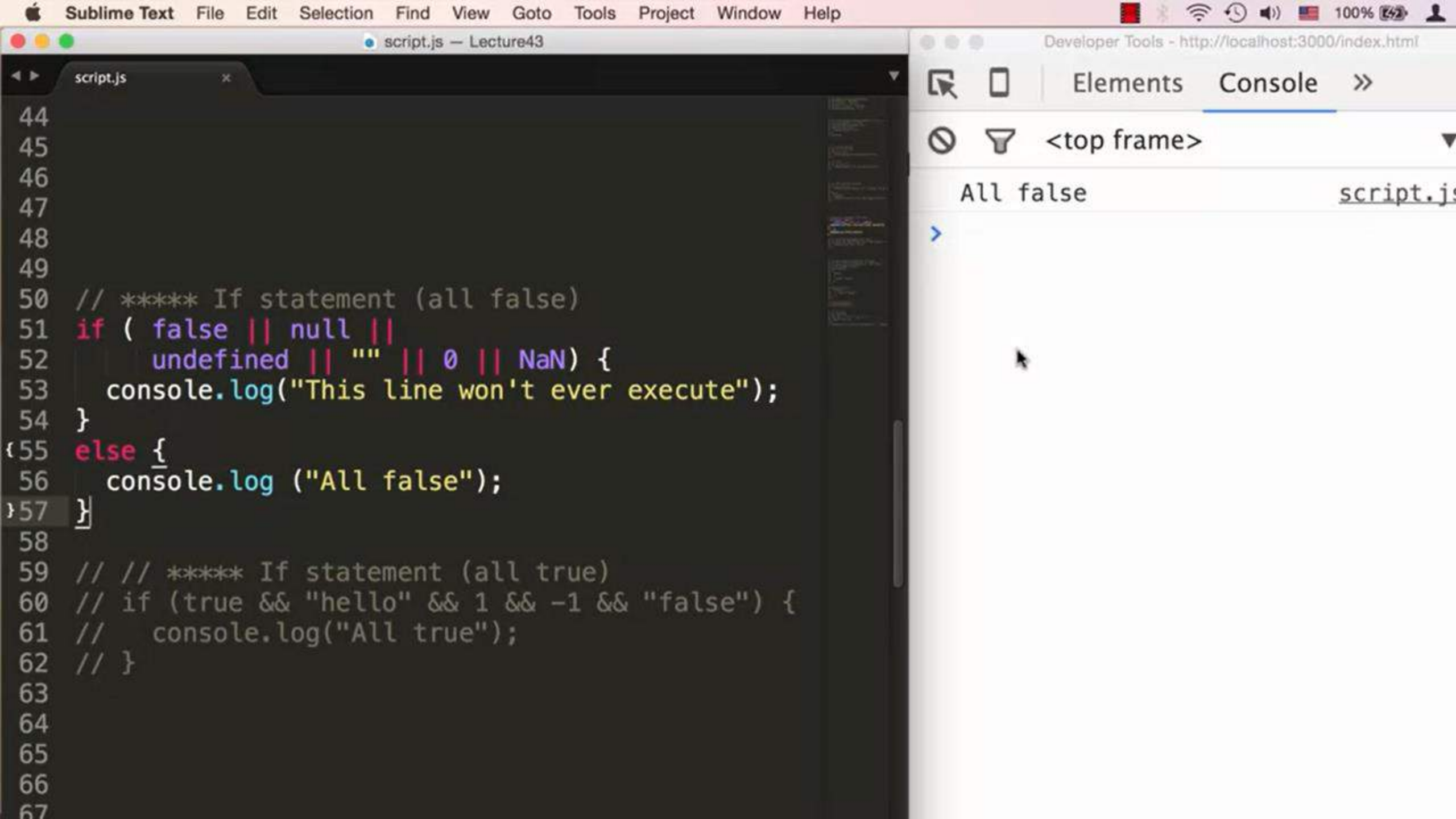

Sublime Text File Edit Selection Find View Goto Tools Project Window Help
100%
script.js — Lecture43
script.js
// var string = "Hello";
// // string += " World";
// string = string + " World";
// console.log(string + "!");
// ***** Regular math operators: +, -, *, /
console.log((5 + 4) / 3);
console.log(undefined / 5);
// // ***** Equality
// var x = 4, y = 4;
// if (x == y) {
//   console.log("x=4 is equal to y=4");
// }
// x = "4";
// if (x == y) {
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
3
script.js
NaN
script.js

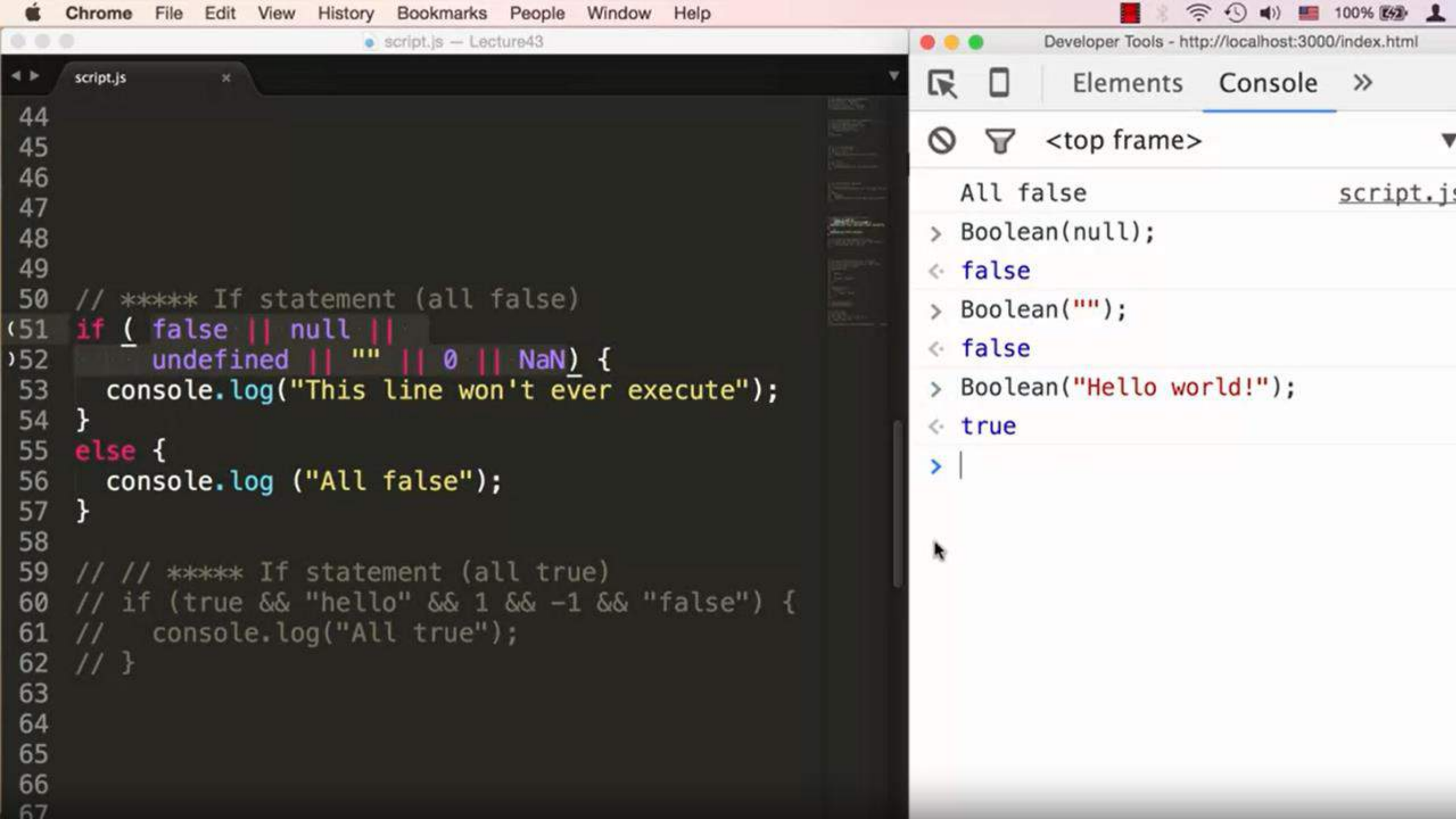

Sublime Text File Edit Selection Find View Goto Tools Project Window Help
script.js — Lecture43
script.js
// var string = "Hello";
// // string += " World";
// string = string + " World";
// console.log(string + "!");
// ***** Regular math operators: +, -, *, /
console.log((5 + 4) / 3);
console.log(undefined / 5);
function test1 (a) {
console.log(a / 5);
}
test1();
// // ***** Equality
// var x = 4, y = 4;
// if (x == y) {
// console.log("x=4 is equal to y=4");
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
3
script.js
NaN
script.js
NaN
script.js

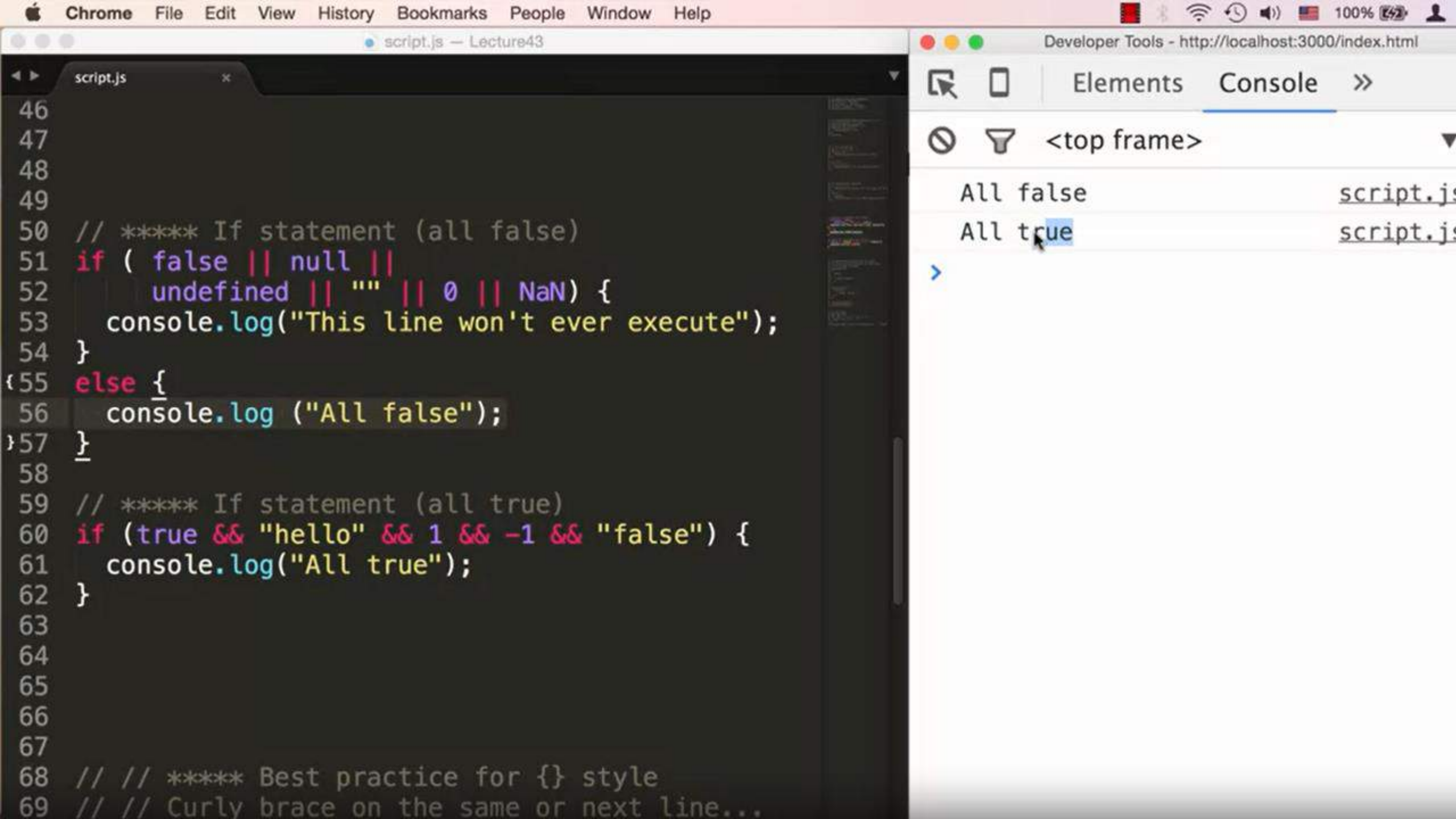

Sublime Text File Edit Selection Find View Goto Tools Project Window Help
script.js — Lecture43
script.js
// ***** Equality
var x = 4, y = 4;
if (x == y) {
console.log("x=4 is equal to y=4");
}
// x = "4";
// if (x == y) {
// console.log("x='4' is equal to y=4");
// }
// // ***** Strict equality
// if (x === y) {
// console.log("Strict: x='4' is equal to y=4")
// }
Developer Tools - http://localhost:3000/index.html
Elements Console
<top frame>
x=4 is equal to y=4
script.js

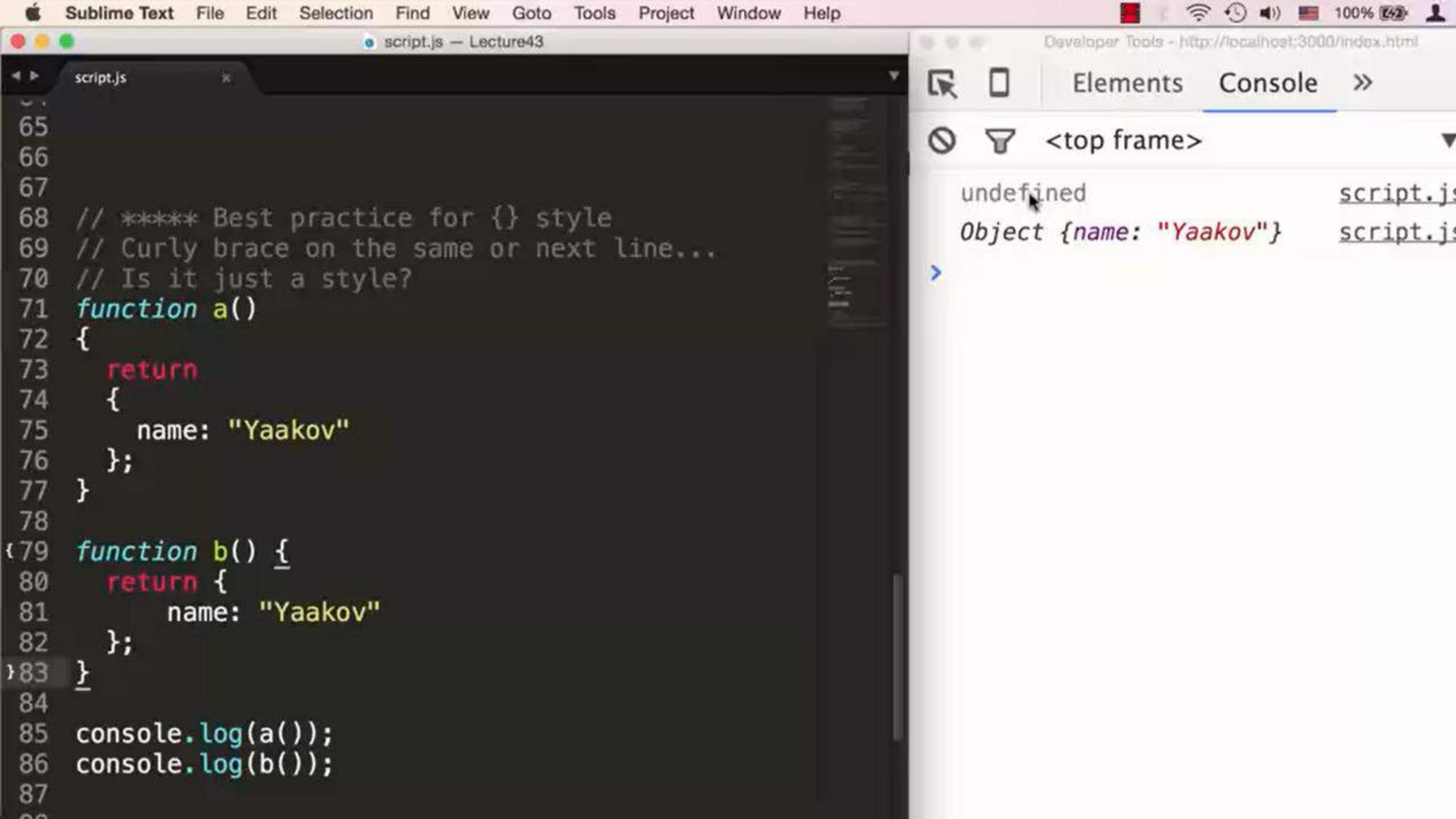

Sublime Text File Edit Selection Find View Goto Tools Project Window Help
script.js — Lecture43
script.js
17
18
19
20
21 // ***** Equality
22 var x = 4, y = 4;
23 if (x == y) {
24 console.log("x=4 is equal to y=4");
25 }
26
27 x = "4";
28 if (x == y) {
29 console.log("x='4' is equal to y=4");
30 }
31
32
33
34
35
36 // // ***** Strict equality
37 // if (x === y) {
38 // console.log("Strict: x='4' is equal to y=4")
39 // }
40 // else {
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
x=4 is equal to y=4
script.js
x='4' is equal to y=4
script.js

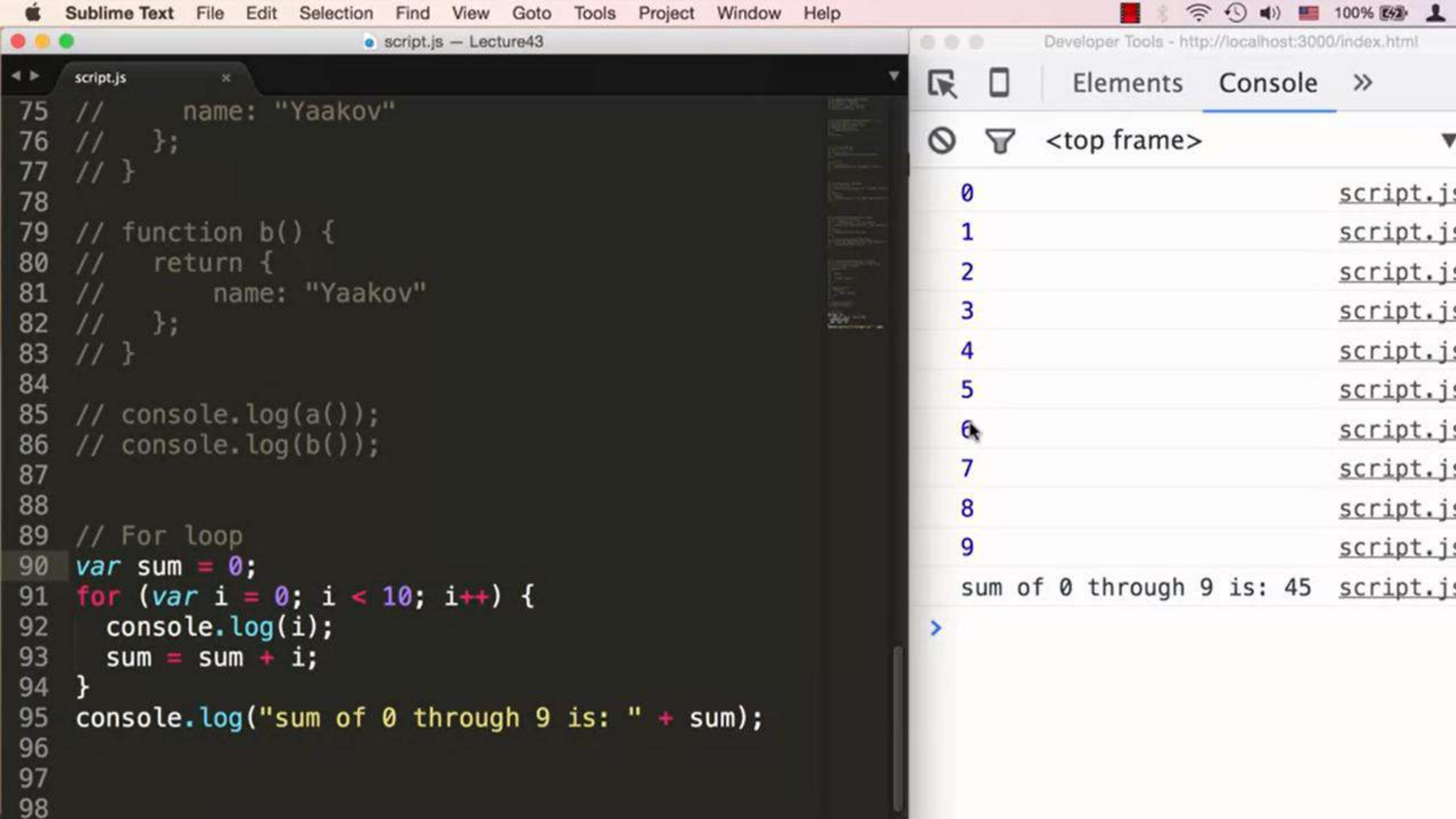

Sublime Text File Edit Selection Find View Goto Tools Project Window Help
script.js — Lecture43
script.js
29 console.log("x='4' is equal to y=4");
30 }
31
32
33
34
35
36 // ***** Strict equality
37 if (x === y) {
38 console.log("Strict: x='4' is equal to y=4");
39 }
40 else {
41 console
42 .log("Strict: x='4' is NOT equal to y=4");
43 }
44
45
46
47
48
49
50 // // ***** If statement (all false)
51 // if (false || null ||
52 // undefined || "" || 0 || NaN) {
Developer Tools - http://localhost:3000/index.html
Elements Console
<top frame>
x=4 is equal to y=4 script.js
x='4' is equal to y=4 script.js
Strict: x='4' is NOT equal to y=4 script.js


Sublime Text File Edit Selection Find View Goto Tools Project Window Help
script.js — Lecture43
script.js
44
45
46
47
48
49
50 // ***** If statement (all false)
51 if (false || null ||
52 undefined || "" || 0 || NaN) {
53 console.log("This line won't ever execute");
54 }
55 else {
56 console.log ("All false");
57 }
58
59 // // ***** If statement (all true)
60 // if (true && "hello" && 1 && -1 && "false") {
61 // console.log("All true");
62 // }
63
64
65
66
67
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
All false
script.js
>


Chrome File Edit View History Bookmarks People Window Help
100%
script.js — Lecture43
script.js
44
45
46
47
48
49
50 // ***** If statement (all false)
51 if (false || null ||
52 undefined || "" || 0 || NaN) {
53 console.log("This line won't ever execute");
54 }
55 else {
56 console.log ("All false");
57 }
58
59 // // ***** If statement (all true)
60 // if (true && "hello" && 1 && -1 && "false") {
61 // console.log("All true");
62 // }
63
64
65
66
67
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
All false
script.js
> Boolean(null);
< false
> Boolean("");
< false
> Boolean("Hello world!");
< true
>

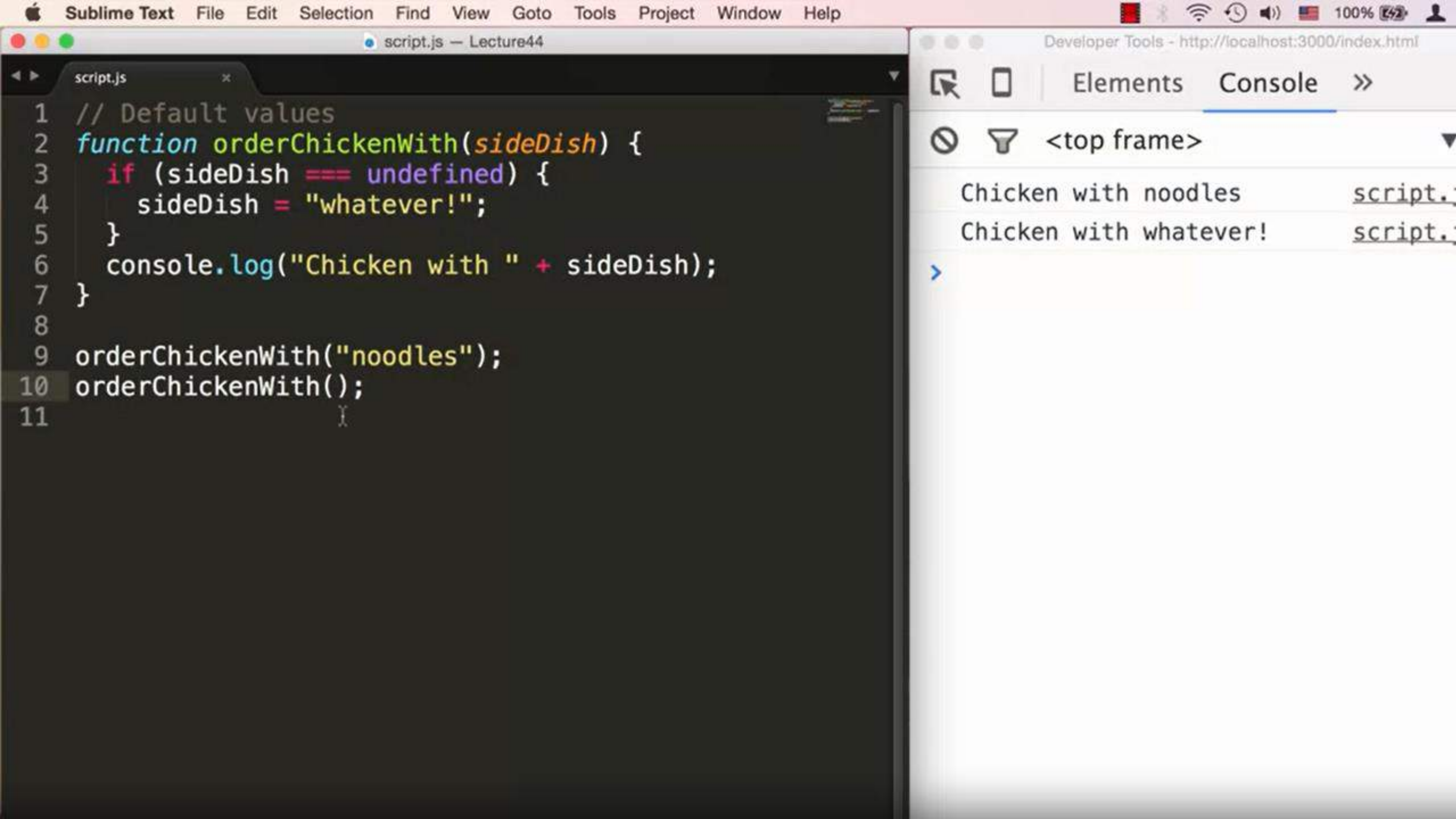

Chrome File Edit View History Bookmarks People Window Help
script.js — Lecture43
script.js
46
47
48
49
50 // ***** If statement (all false)
51 if (false || null ||
52 undefined || "" || 0 || NaN) {
53 console.log("This line won't ever execute");
54 }
55 else {
56 console.log ("All false");
57 }
58
59 // ***** If statement (all true)
60 if (true && "hello" && 1 && -1 && "false") {
61 console.log("All true");
62 }
63
64
65
66
67
68 // // ***** Best practice for {} style
69 // // Curly brace on the same or next line...
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
All false
script.js
All true
script.js

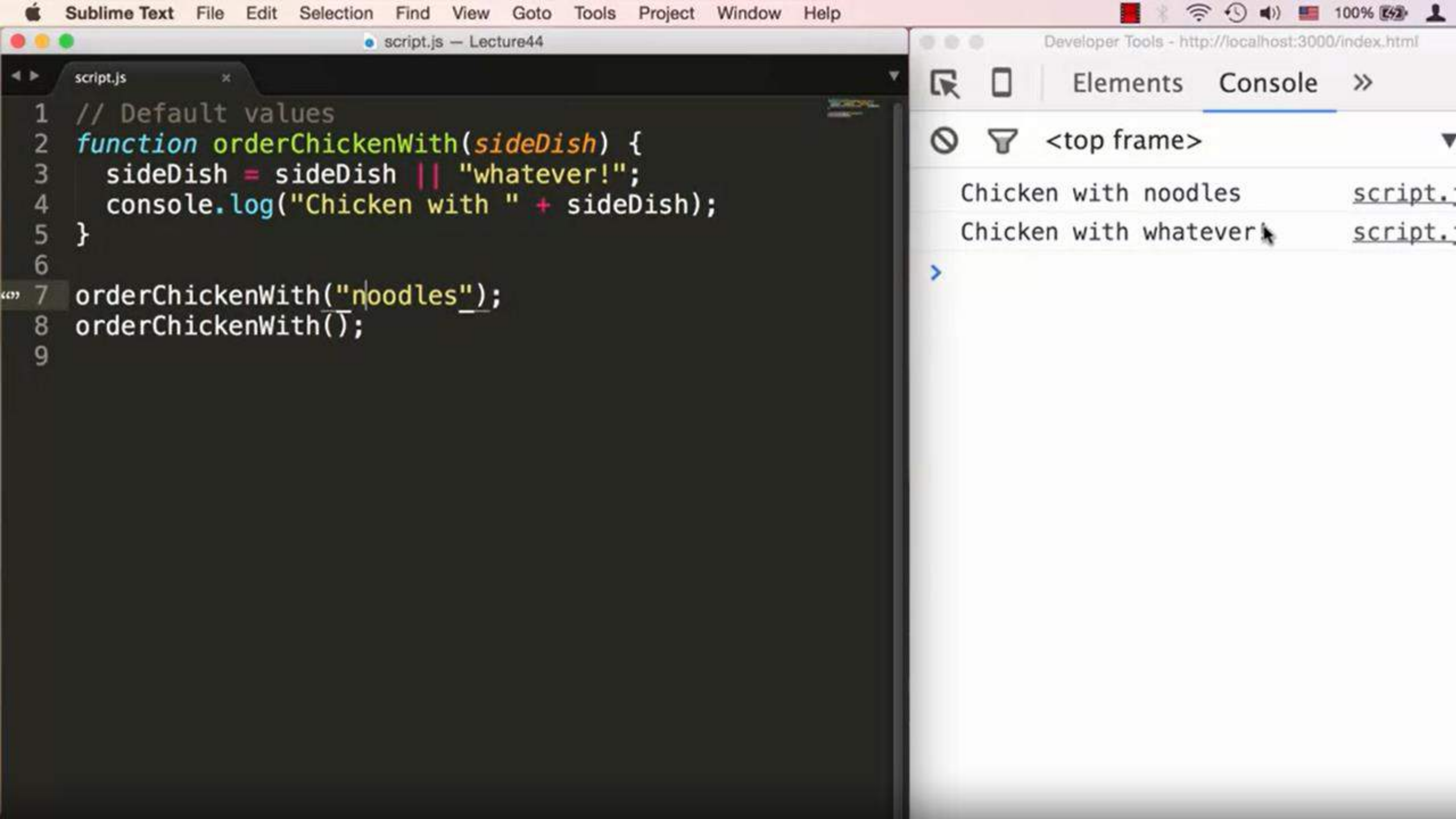

Sublime Text File Edit Selection Find View Goto Tools Project Window Help
script.js — Lecture43
script.js
65
66
67
68 // ***** Best practice for {} style
69 // Curly brace on the same or next line...
70 // Is it just a style?
71 function a()
72 {
73 return
74 {
75 name: "Yaakov"
76 };
77 }
78
79 function b() {
80 return {
81 name: "Yaakov"
82 };
83 }
84
85 console.log(a());
86 console.log(b());
87
Developer Tools - http://localhost:3000/index.html
Elements Console
<top frame>
undefined script.js
Object {name: "Yaakov"} script.js

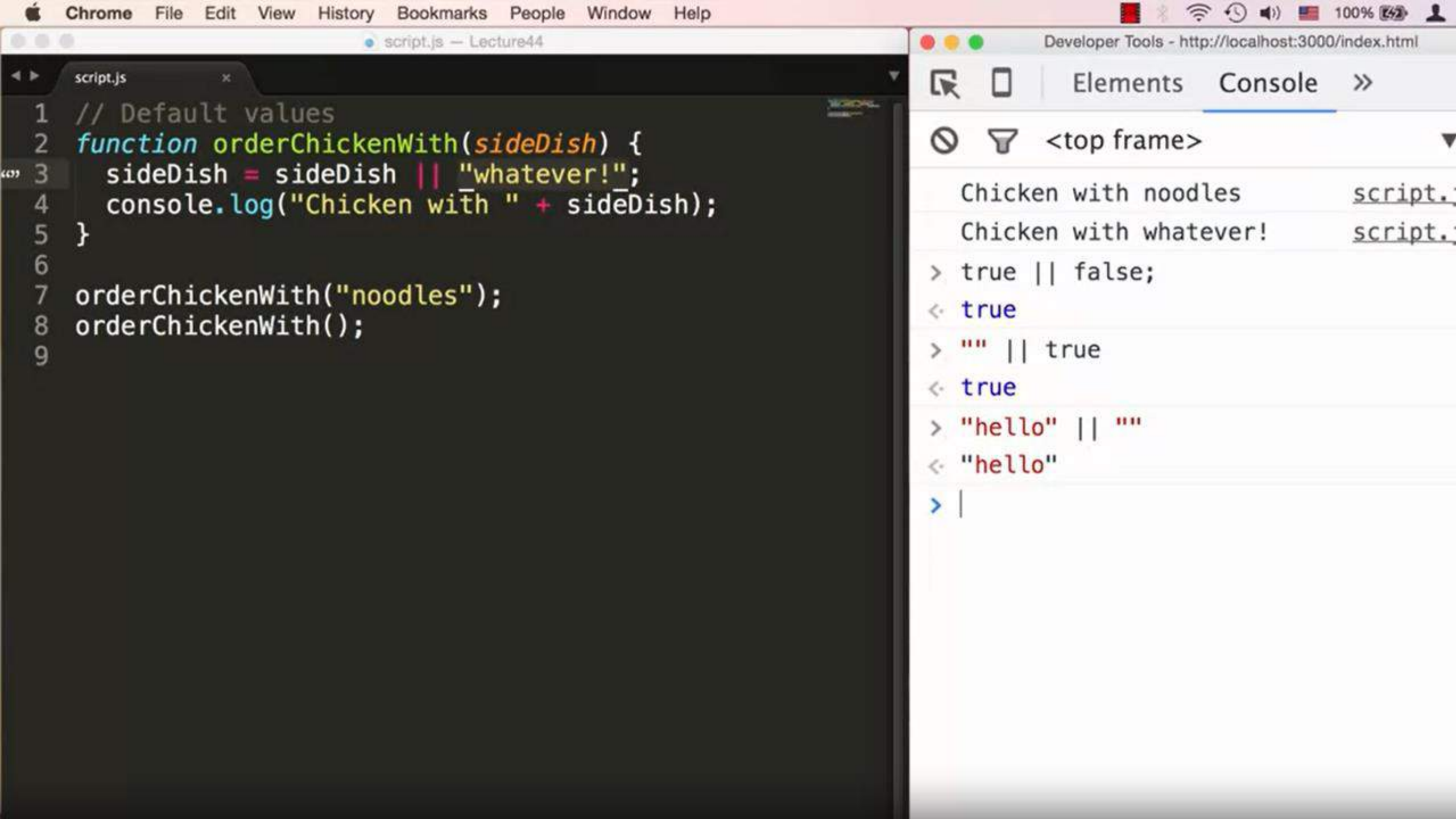

Sublime Text File Edit Selection Find View Goto Tools Project Window Help
script.js — Lecture43
script.js
// name: "Yaakov"
// };
// }
// function b() {
// return {
// name: "Yaakov"
// };
// }
// console.log(a());
// console.log(b());
// For loop
var sum = 0;
for (var i = 0; i < 10; i++) {
console.log(i);
sum = sum + i;
}
console.log("sum of 0 through 9 is: " + sum);
Developer Tools - http://localhost:3000/index.html
Elements Console
<top frame>
0 script.js
1 script.js
2 script.js
3 script.js
4 script.js
5 script.js
6 script.js
7 script.js
8 script.js
9 script.js
sum of 0 through 9 is: 45 script.js

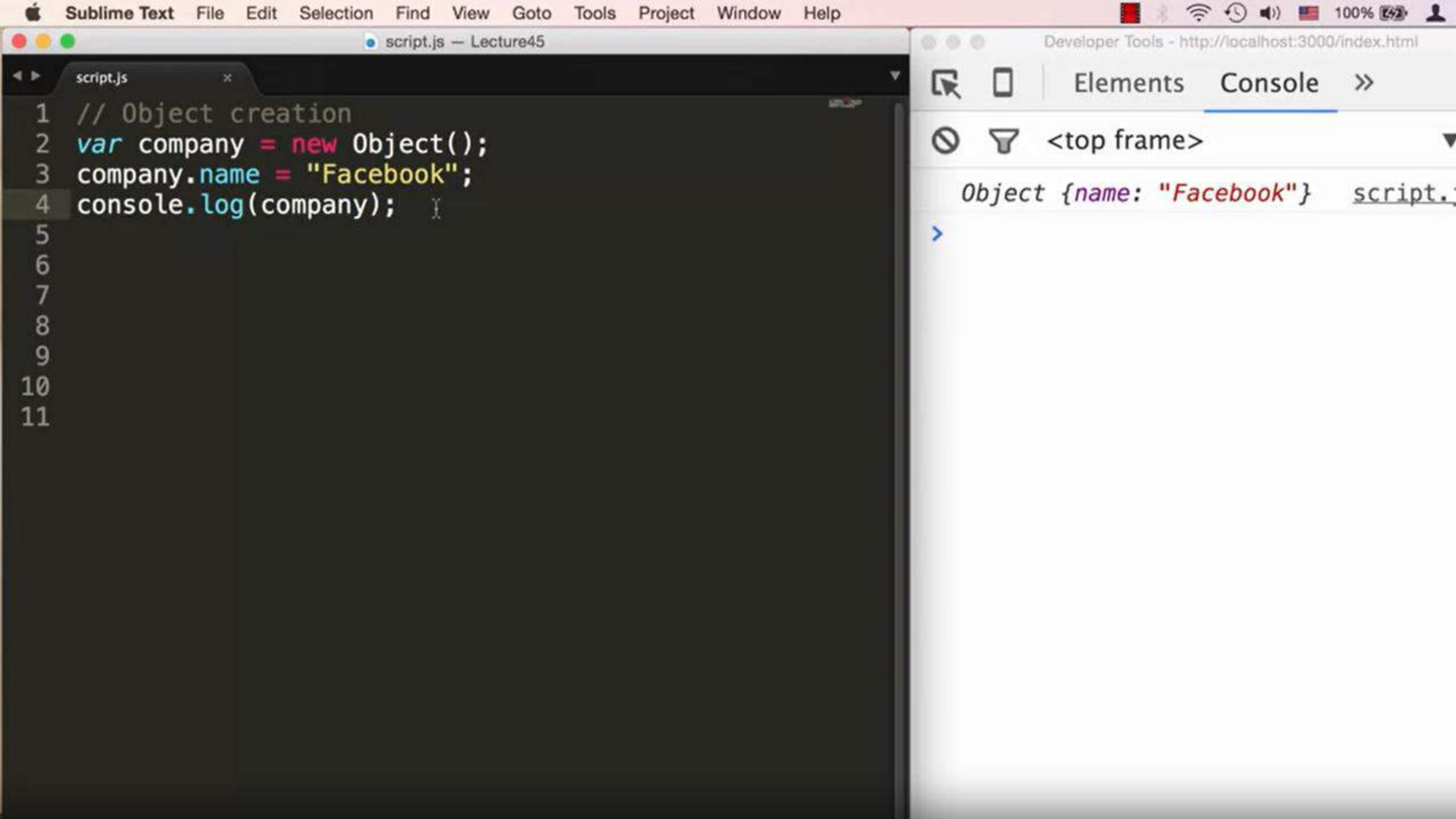
Summary

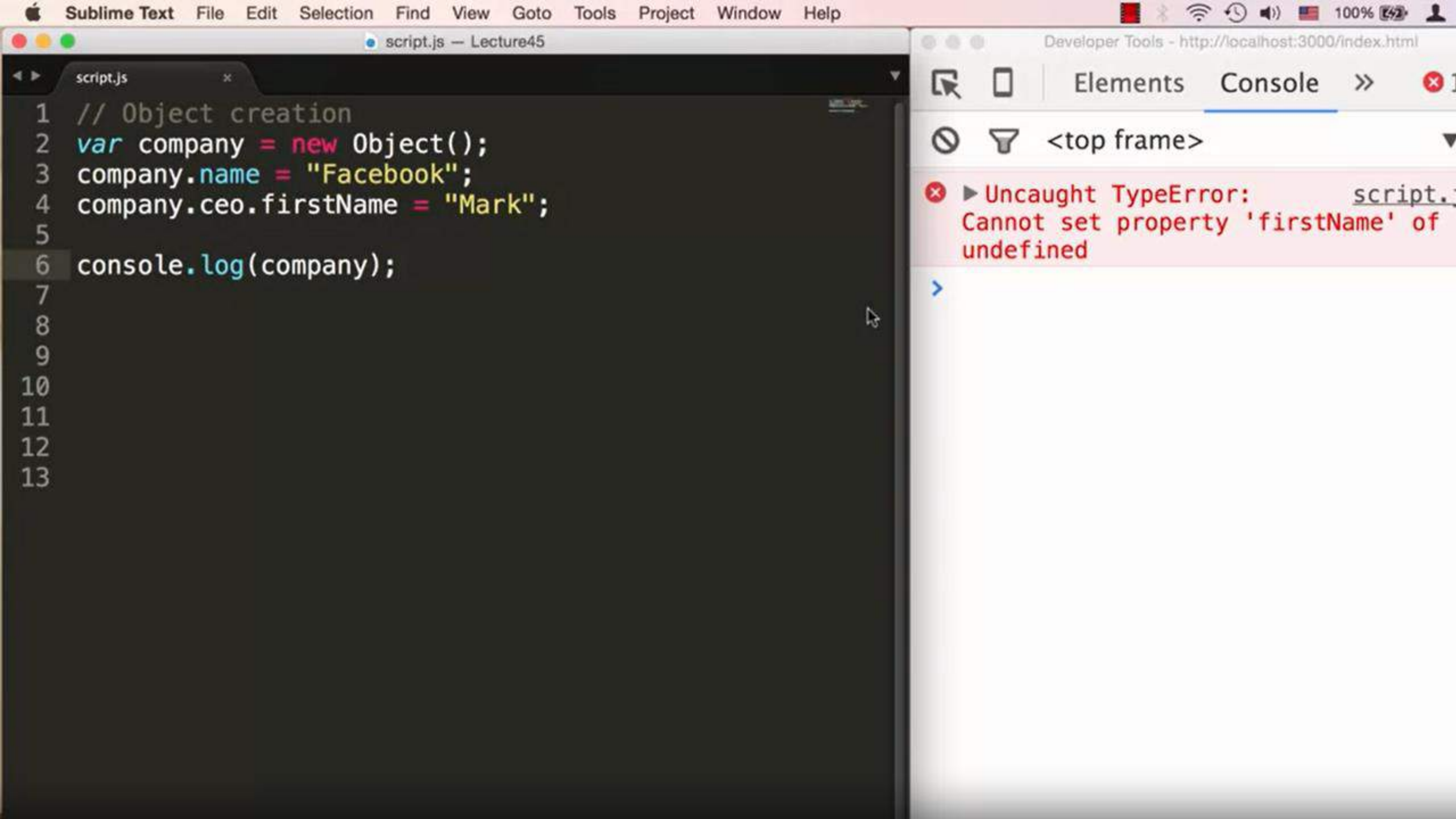
- String concatenation
- Math operations
- Type coercion
- Regular (==) and Strict (===) equality
- What is true and what is false in Javascript
- Opening curly brace placement (NOT just style)
- Always place semicolons at the end of statements
  - Some disagree: http://mislav.net/2010/05/semicolons/
- Basic for loop syntax

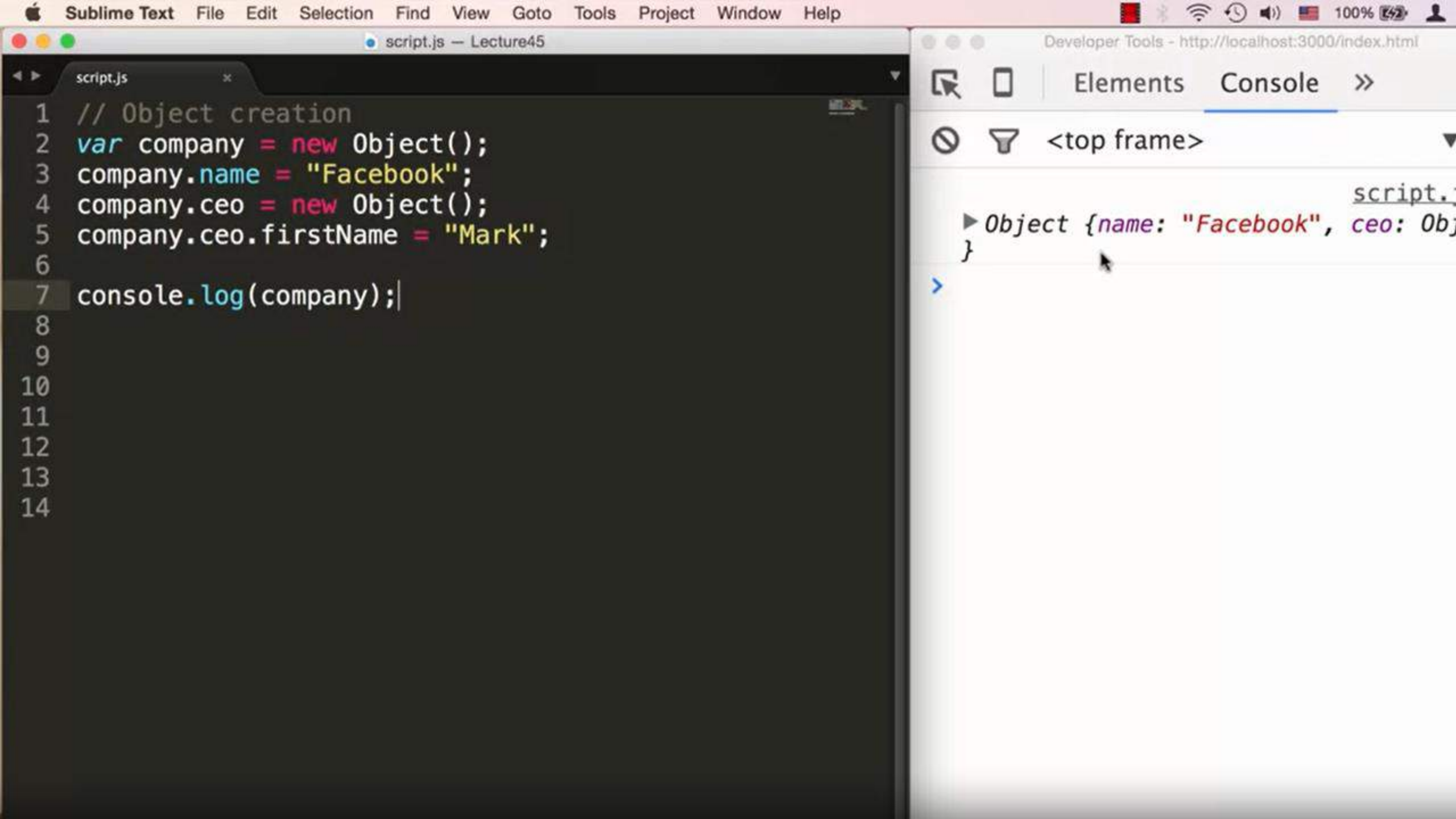

Sublime Text File Edit Selection Find View Goto Tools Project Window Help
100%
script.js — Lecture44
script.js
1 // Default values
2 function orderChickenWith(sideDish) {
3 if (sideDish === undefined) {
4 sideDish = "whatever!";
5 }
6 console.log("Chicken with " + sideDish);
7 }
8
9 orderChickenWith("noodles");
10 orderChickenWith();
11
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
Chicken with noodles
script.
Chicken with whatever!
script.
>

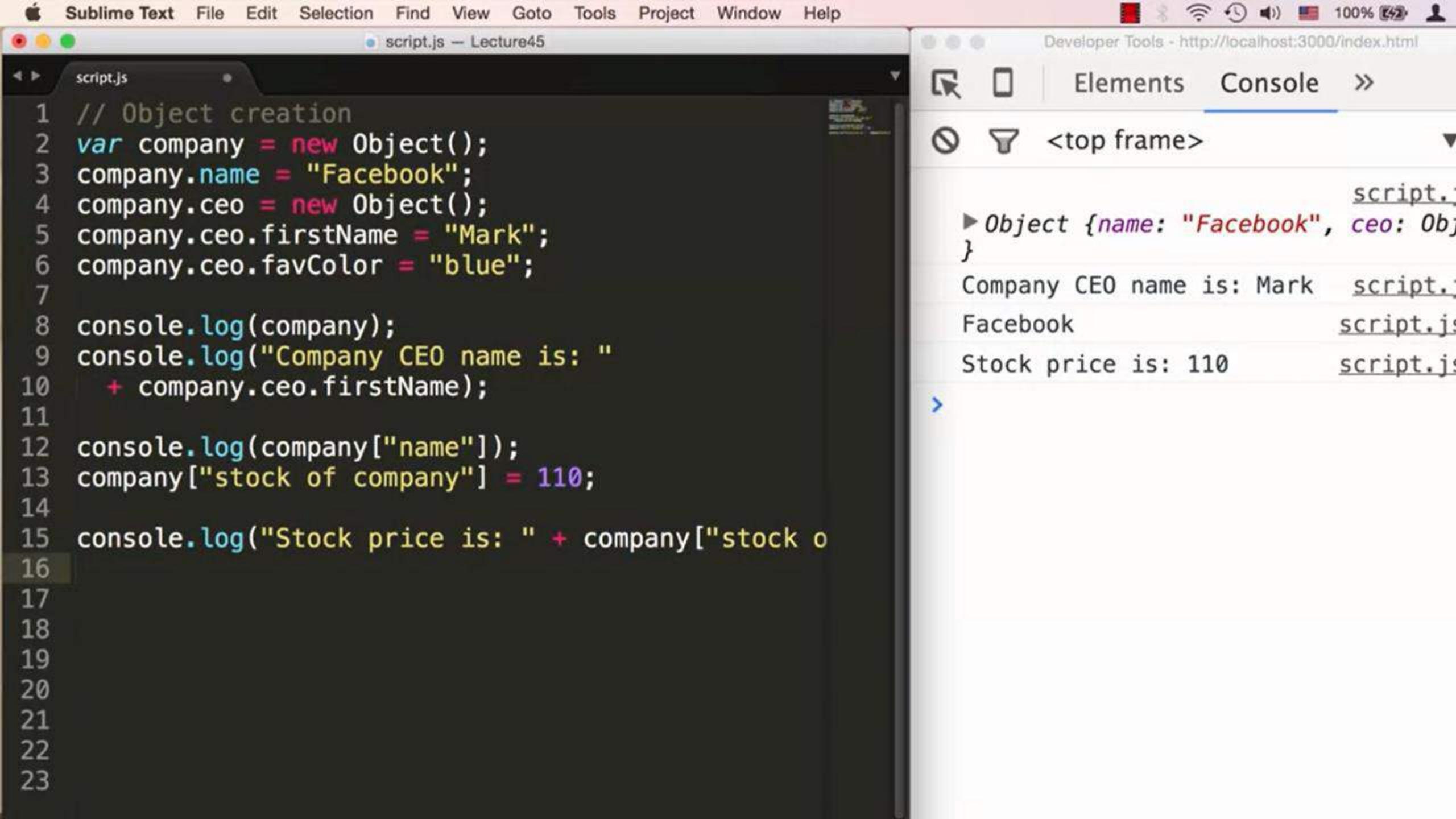

Sublime Text File Edit Selection Find View Goto Tools Project Window Help
100%
script.js — Lecture44
script.js
1 // Default values
2 function orderChickenWith(sideDish) {
3 sideDish = sideDish || "whatever!";
4 console.log("Chicken with " + sideDish);
5 }
6
7 orderChickenWith("noodles");
8 orderChickenWith();
9
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
Chicken with noodles
script.
Chicken with whatever!
script.

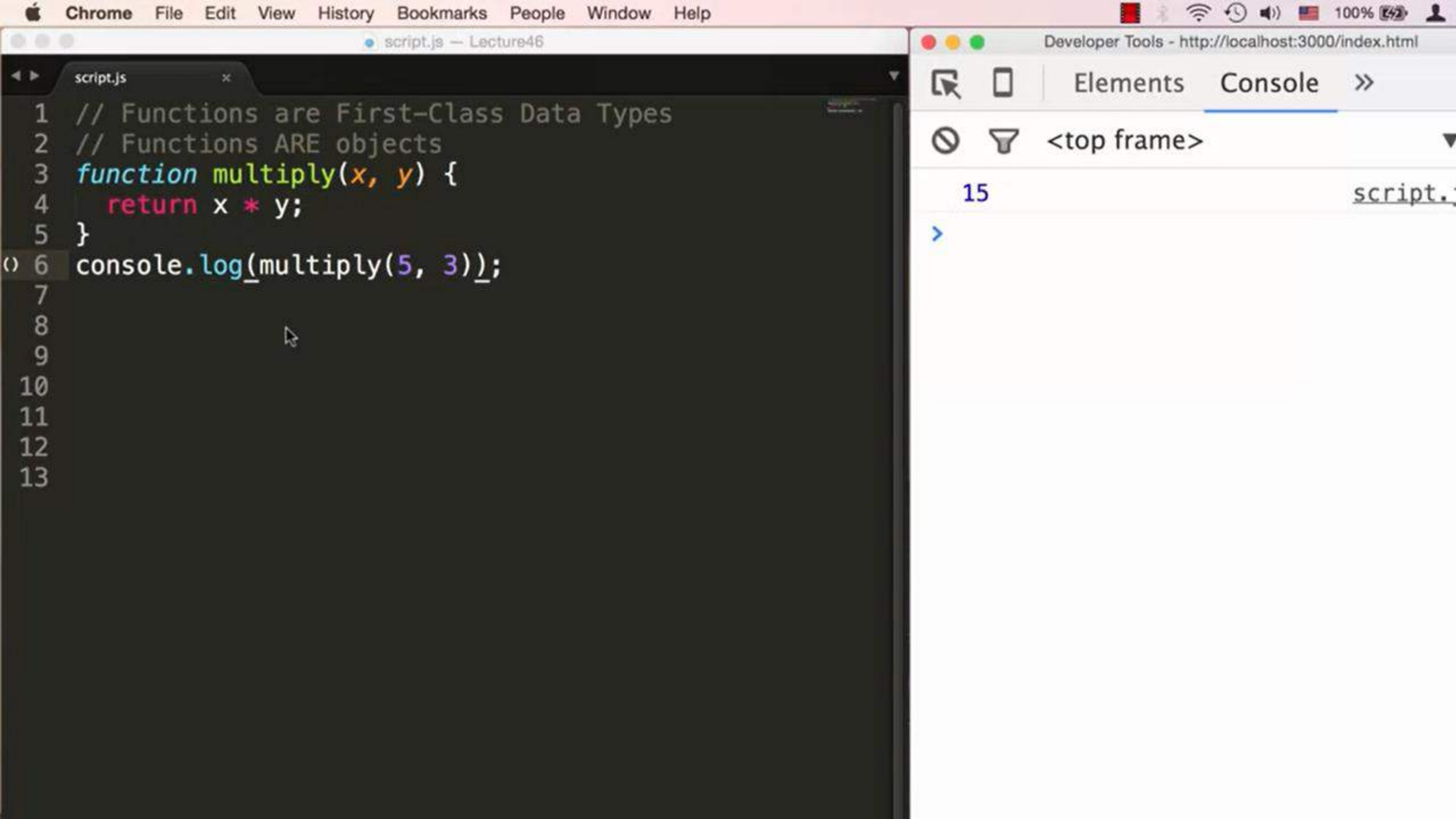

Chrome File Edit View History Bookmarks People Window Help
script.js — Lecture44
script.js
1 // Default values
2 function orderChickenWith(sideDish) {
3 sideDish = sideDish || "whatever!";
4 console.log("Chicken with " + sideDish);
5 }
6
7 orderChickenWith("noodles");
8 orderChickenWith();
9
Developer Tools - http://localhost:3000/index.html
Elements Console
<top frame>
Chicken with noodles script.
Chicken with whatever! script.
> true || false;
< true
> "" || true
< true
> "hello" || ""
< "hello"
>

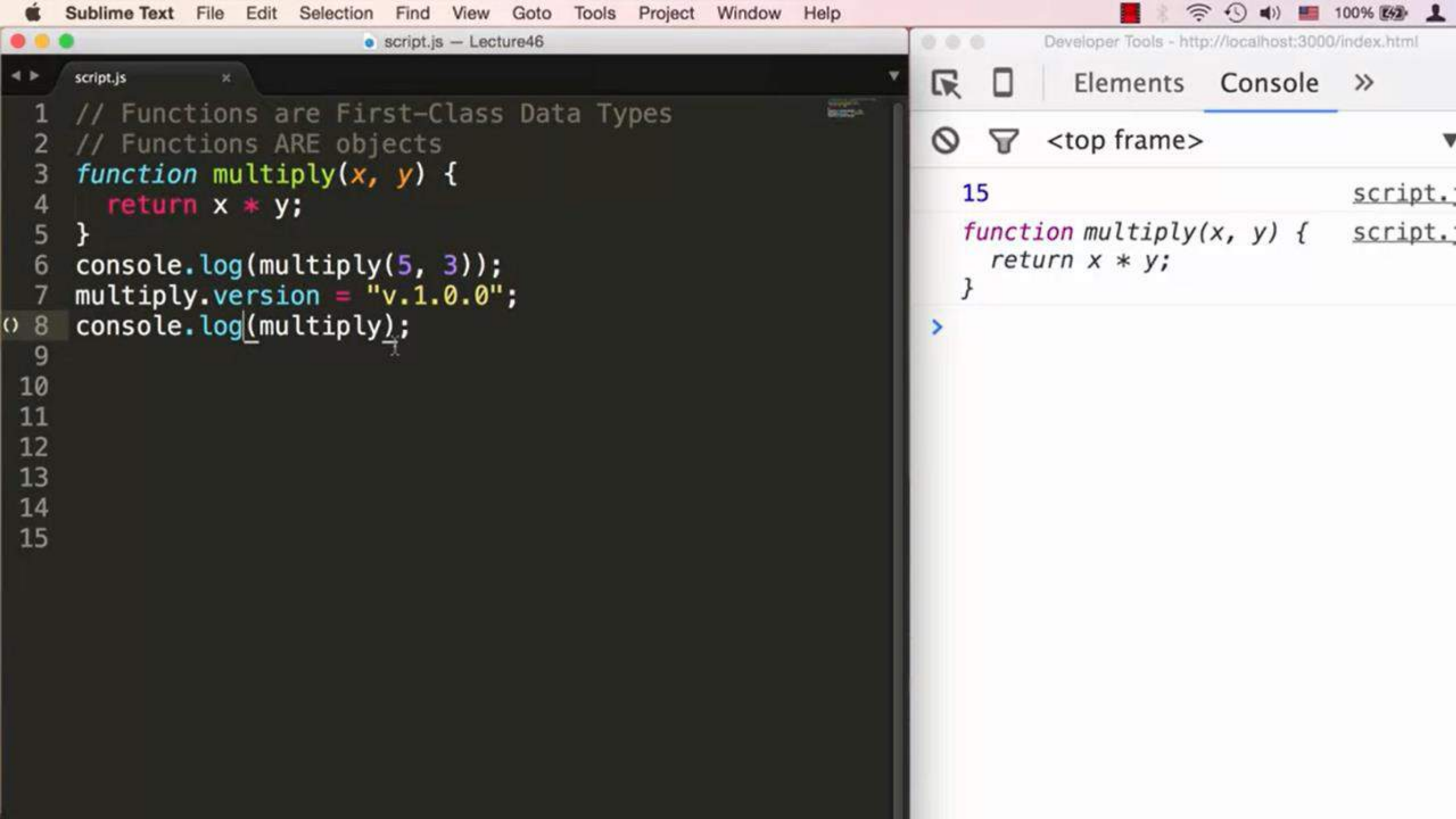

Sublime Text File Edit Selection Find View Goto Tools Project Window Help
100%
script.js — Lecture45
script.js
1 // Object creation
2 var company = new Object();
3 company.name = "Facebook";
4 console.log(company);
5
6
7
8
9
10
11
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
Object {name: "Facebook"}
script.
>

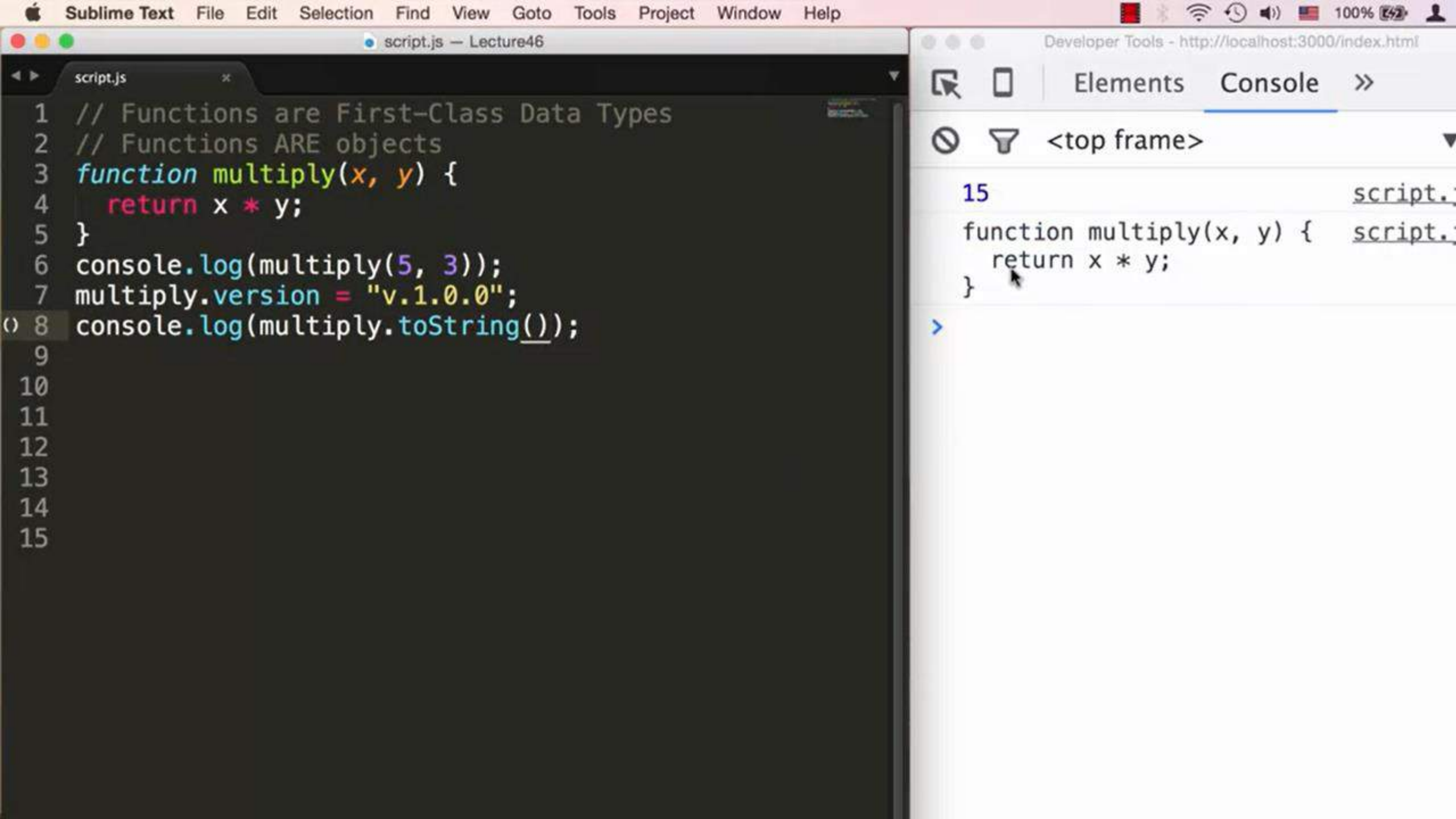

Sublime Text File Edit Selection Find View Goto Tools Project Window Help
100%
script.js — Lecture45
script.js
1 // Object creation
2 var company = new Object();
3 company.name = "Facebook";
4 company.ceo.firstName = "Mark";
5
6 console.log(company);
7
8
9
10
11
12
13
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
Uncaught TypeError: Cannot set property 'firstName' of undefined
script.
>

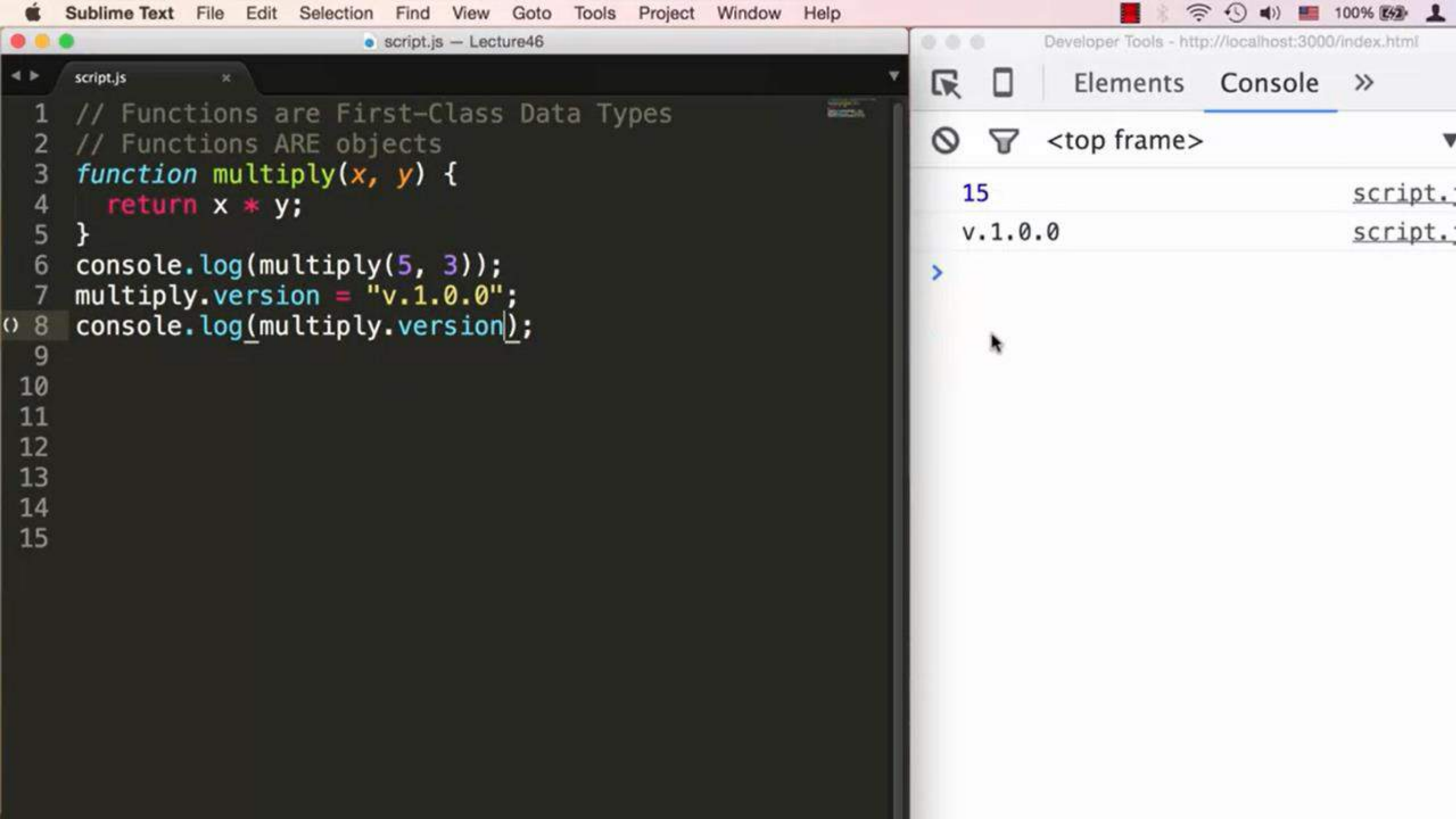

Sublime Text File Edit Selection Find View Goto Tools Project Window Help
100%
script.js — Lecture45
script.js
1 // Object creation
2 var company = new Object();
3 company.name = "Facebook";
4 company.ceo = new Object();
5 company.ceo.firstName = "Mark";
6
7 console.log(company);
8
9
10
11
12
13
14
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
script.
▶ Object {name: "Facebook", ceo: Obj
}
>


Sublime Text File Edit Selection Find View Goto Tools Project Window Help
script.js — Lecture45
script.js
// Object creation
var company = new Object();
company.name = "Facebook";
company.ceo = new Object();
company.ceo.firstName = "Mark";
company.ceo.favColor = "blue";
console.log(company);
console.log("Company CEO name is: "
+ company.ceo.firstName);
console.log(company["name"]);
company["stock of company"] = 110;
console.log("Stock price is: " + company["stock o
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
script.
▶Object {name: "Facebook", ceo: Obj
}
Company CEO name is: Mark
script.
Facebook
script.js
Stock price is: 110
script.js


Chrome File Edit View History Bookmarks People Window Help
100%
script.js — Lecture46
script.js
// Functions are First-Class Data Types
// Functions ARE objects
function multiply(x, y) {
return x * y;
}
console.log(multiply(5, 3));
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
15
script.


Sublime Text File Edit Selection Find View Goto Tools Project Window Help
100%
script.js — Lecture46
script.js
1 // Functions are First-Class Data Types
2 // Functions ARE objects
3 function multiply(x, y) {
4 return x * y;
5 }
6 console.log(multiply(5, 3));
7 multiply.version = "v.1.0.0";
8 console.log(multiply);
9
10
11
12
13
14
15
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
15
script.
function multiply(x, y) {
return x * y;
}
script.
>


Sublime Text File Edit Selection Find View Goto Tools Project Window Help
100%
script.js — Lecture46
script.js
1 // Functions are First-Class Data Types
2 // Functions ARE objects
3 function multiply(x, y) {
4 return x * y;
5 }
6 console.log(multiply(5, 3));
7 multiply.version = "v.1.0.0";
8 console.log(multiply.toString());
9
10
11
12
13
14
15
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
15
script.
function multiply(x, y) {
return x * y;
}
script.
>


Sublime Text File Edit Selection Find View Goto Tools Project Window Help
100%
script.js — Lecture46
script.js
1 // Functions are First-Class Data Types
2 // Functions ARE objects
3 function multiply(x, y) {
4 return x * y;
5 }
6 console.log(multiply(5, 3));
7 multiply.version = "v.1.0.0";
8 console.log(multiply.version);
9
10
11
12
13
14
15
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
15
script.
v.1.0.0
script.
>

In this lecture, we will discuss…

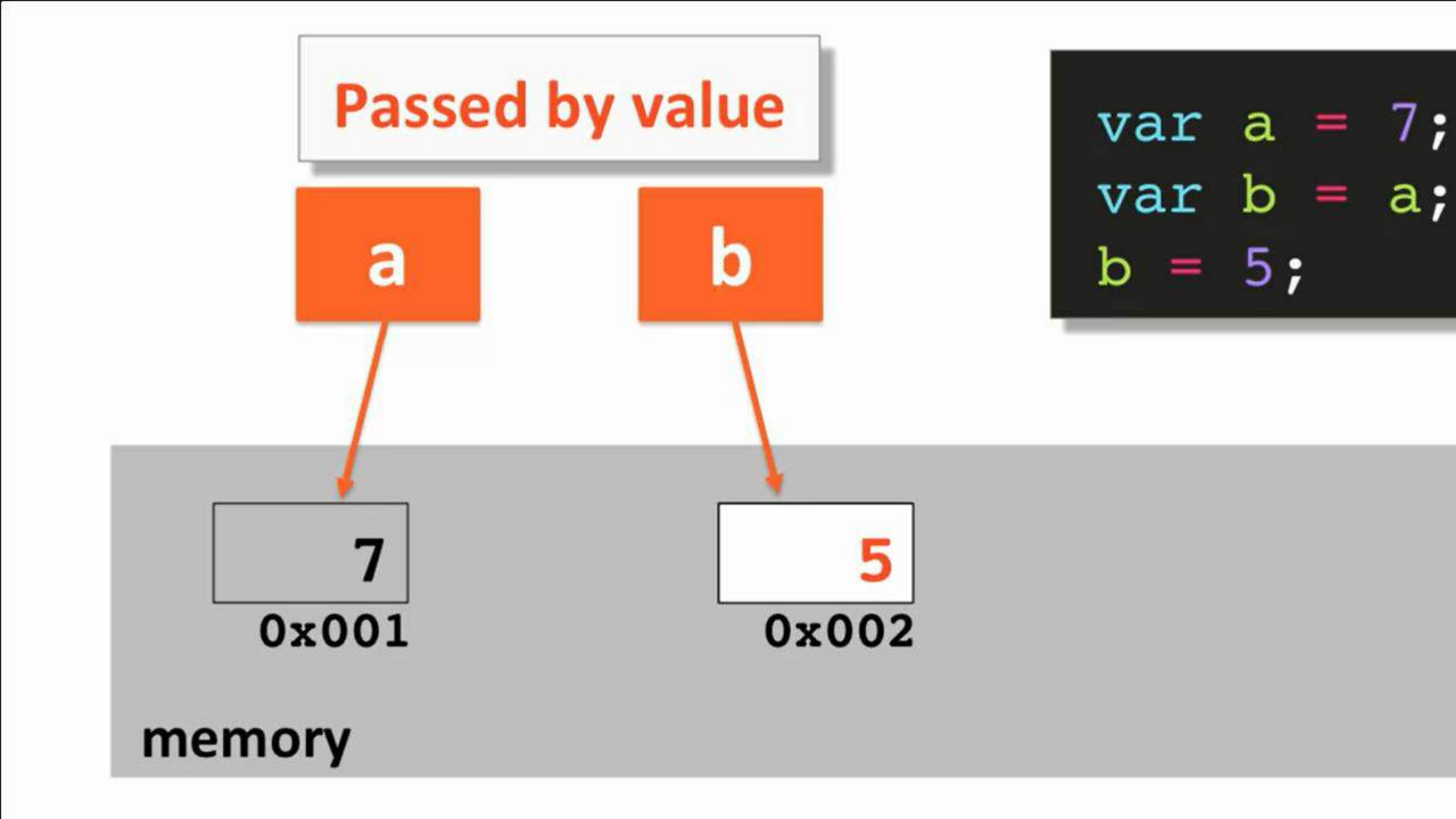
Passing Variables by Value vs. by Reference

Passing (or Copying) By Value

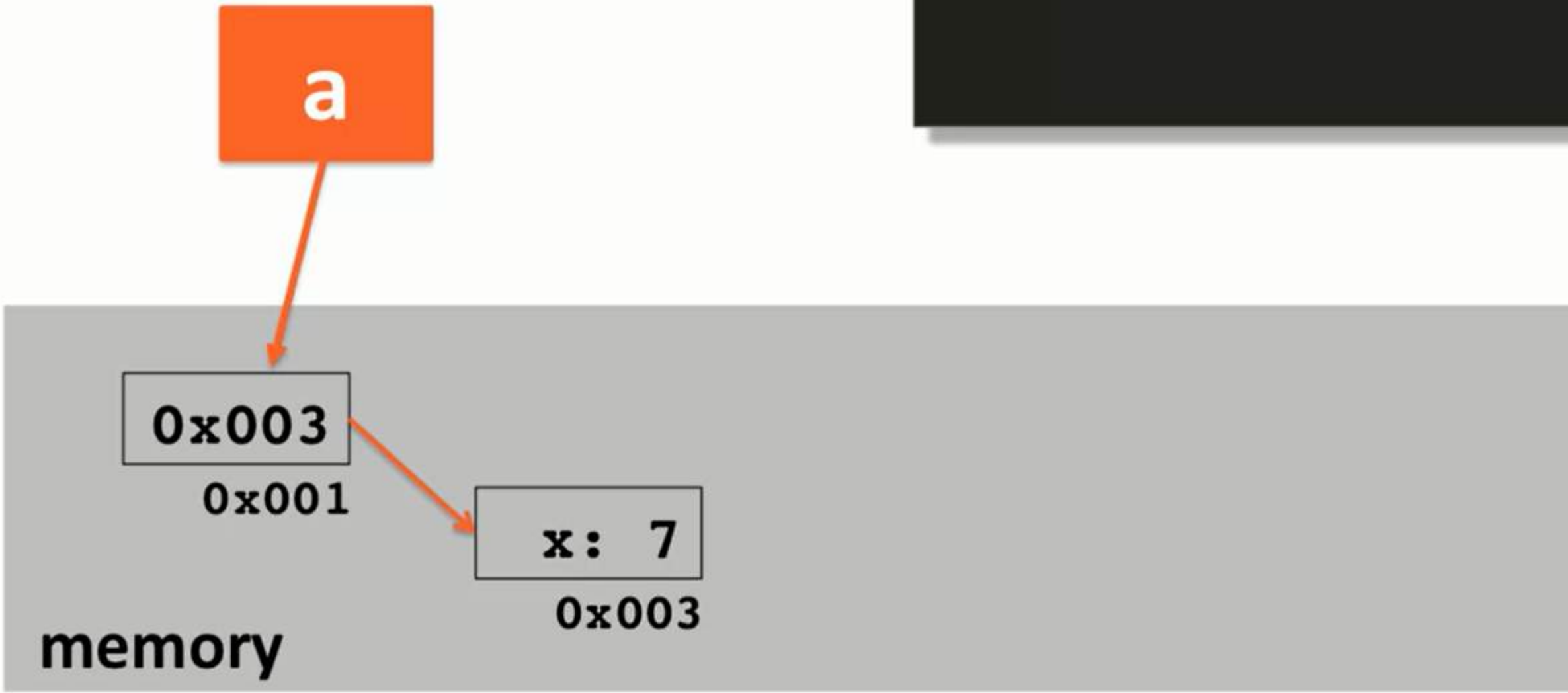
Given b=a, passing/copying by value means changing copied value in b does not affect the value stored in a and visa versa

In Javascript, primitives are passed by value, objects are passed by reference

- ✧ “Under the hood”, everything is actually passed by value

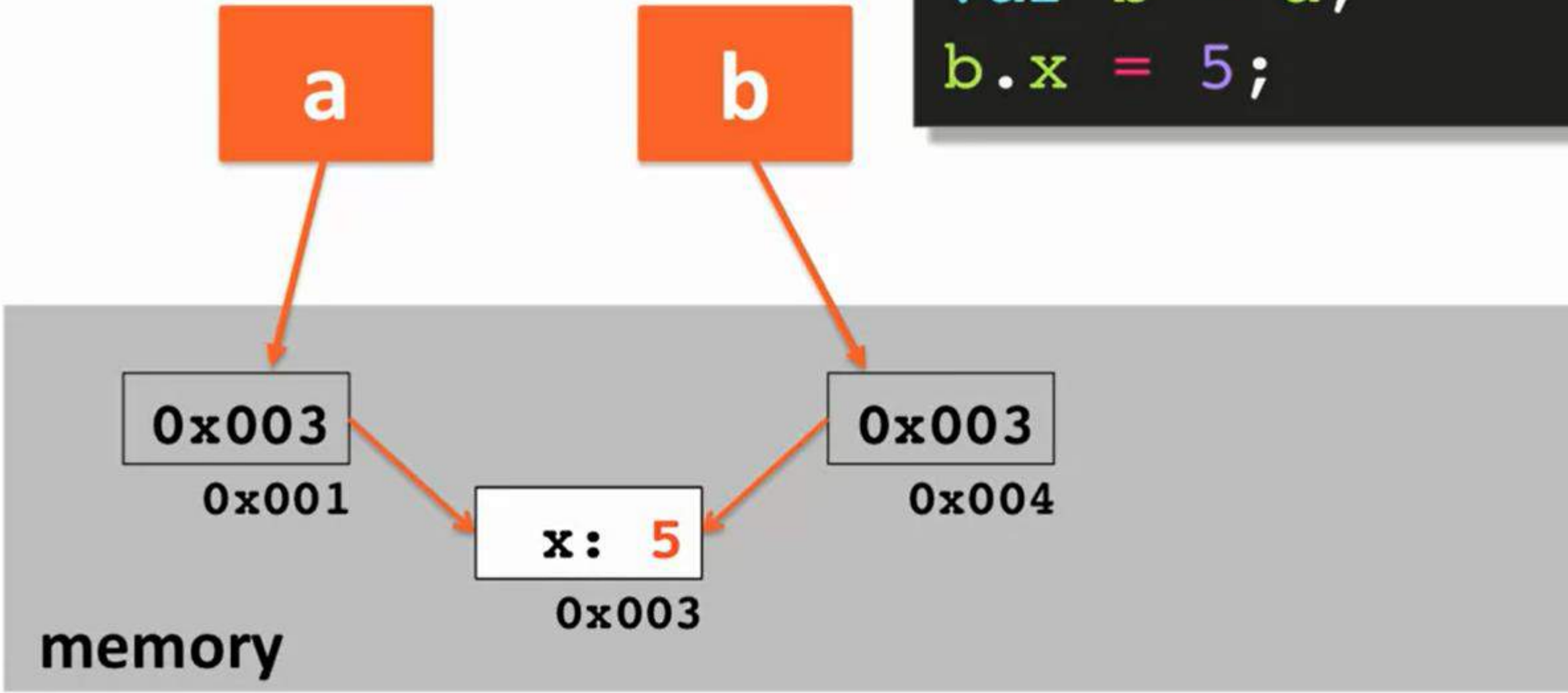

Passed by value
a
b
var a = 7;
var b = a;
b = 5;
7
0x001
5
0x002
memory

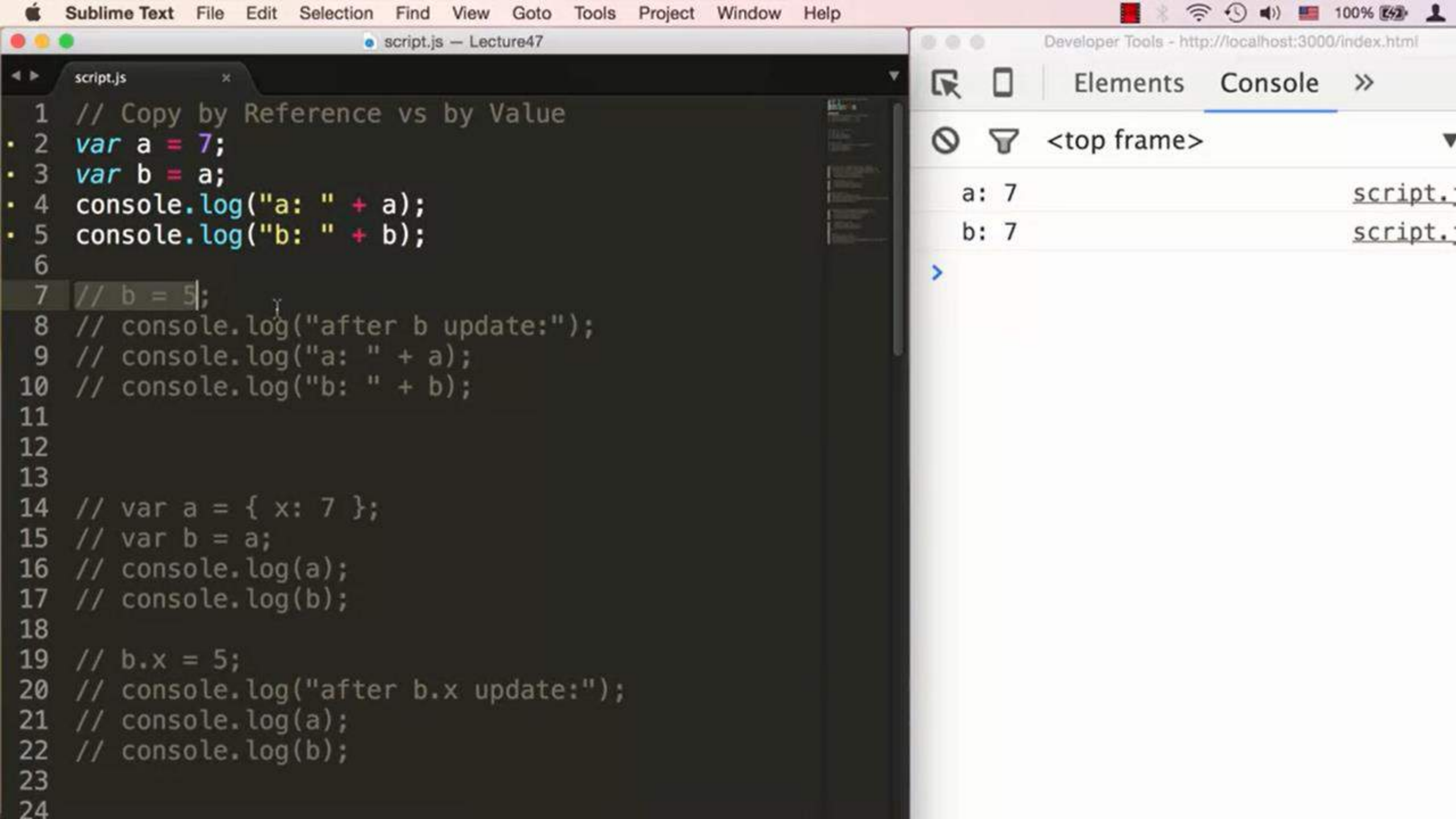
```
var a = {x: 7};
```

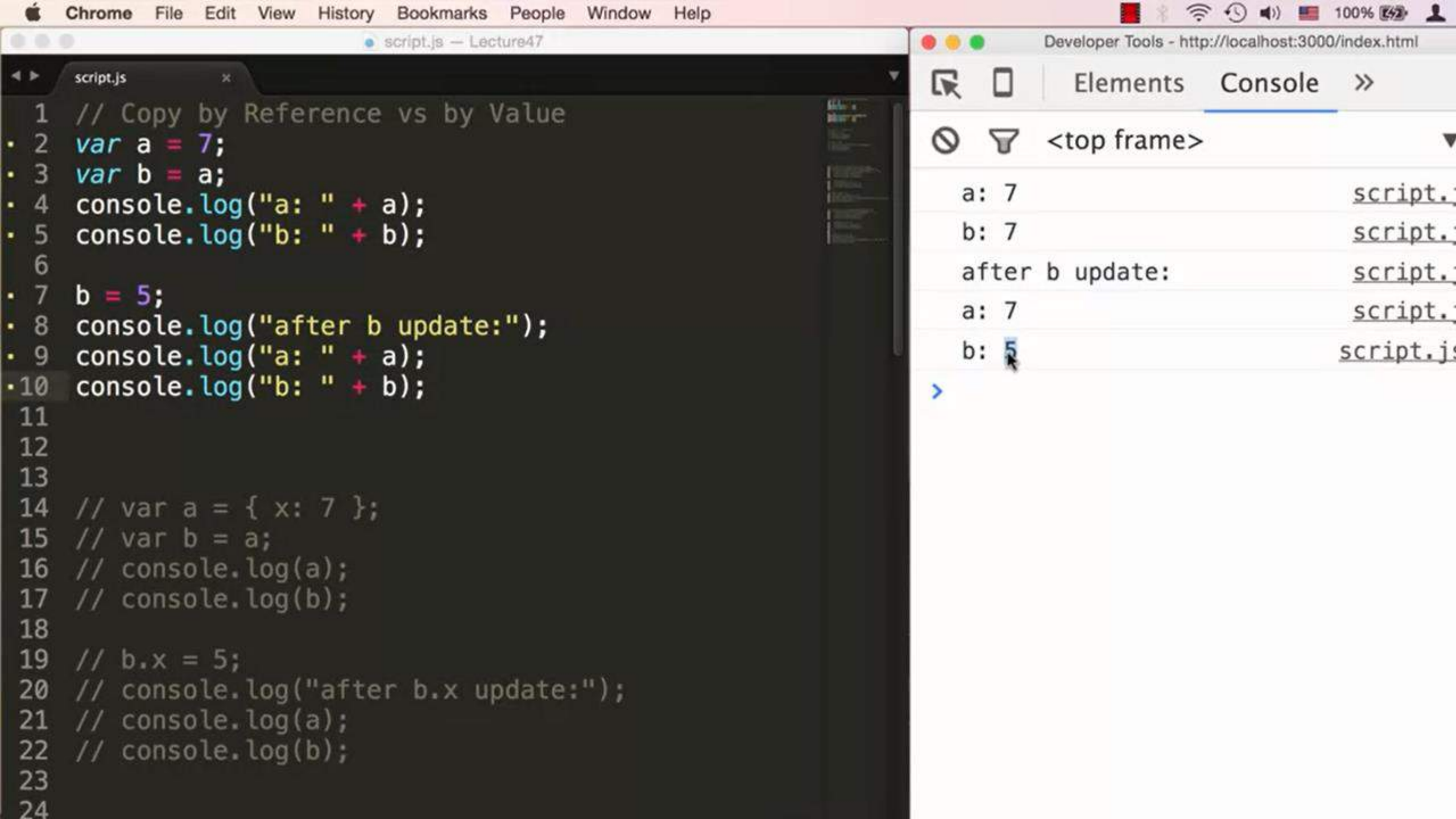


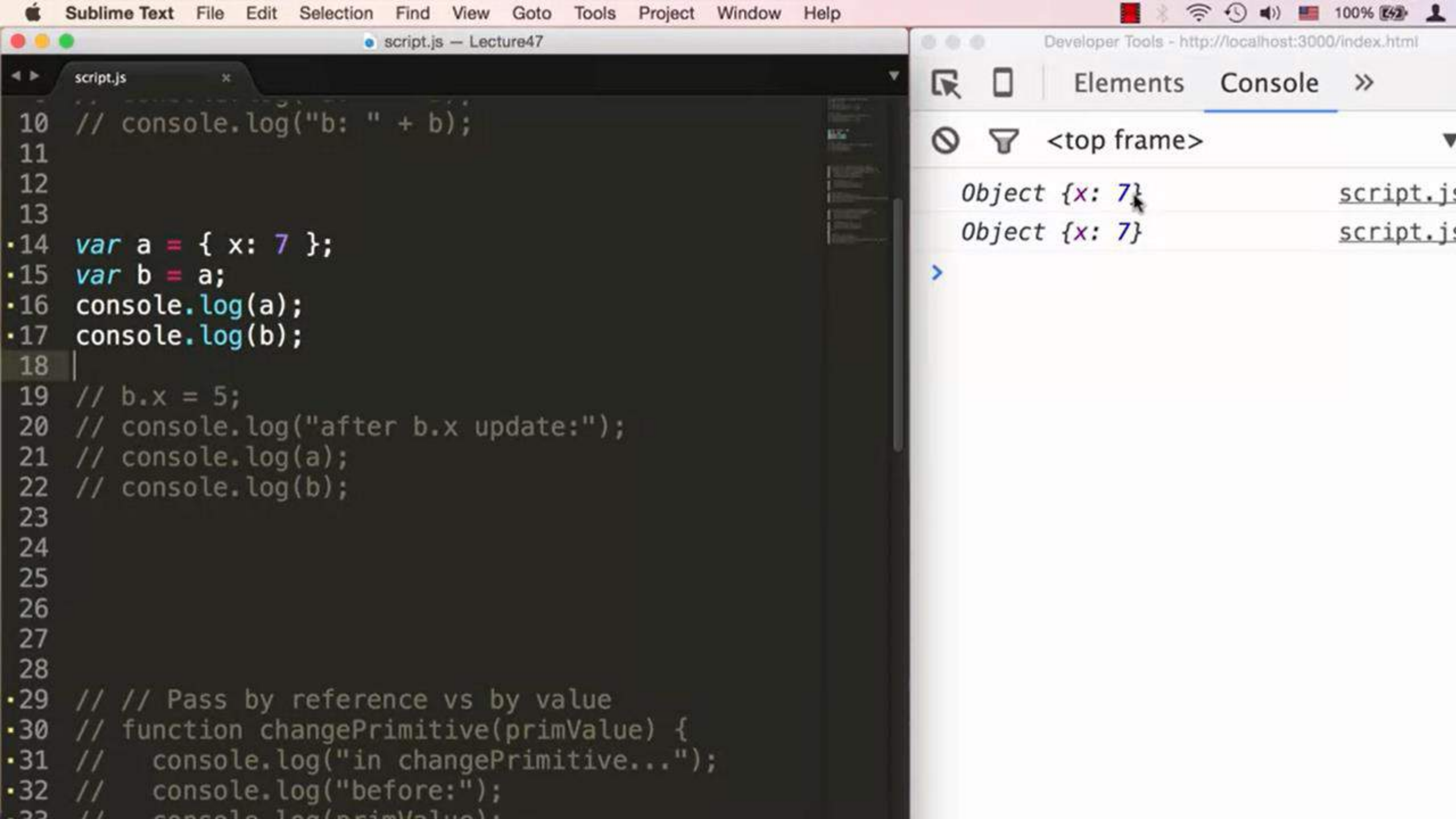
Passed by reference

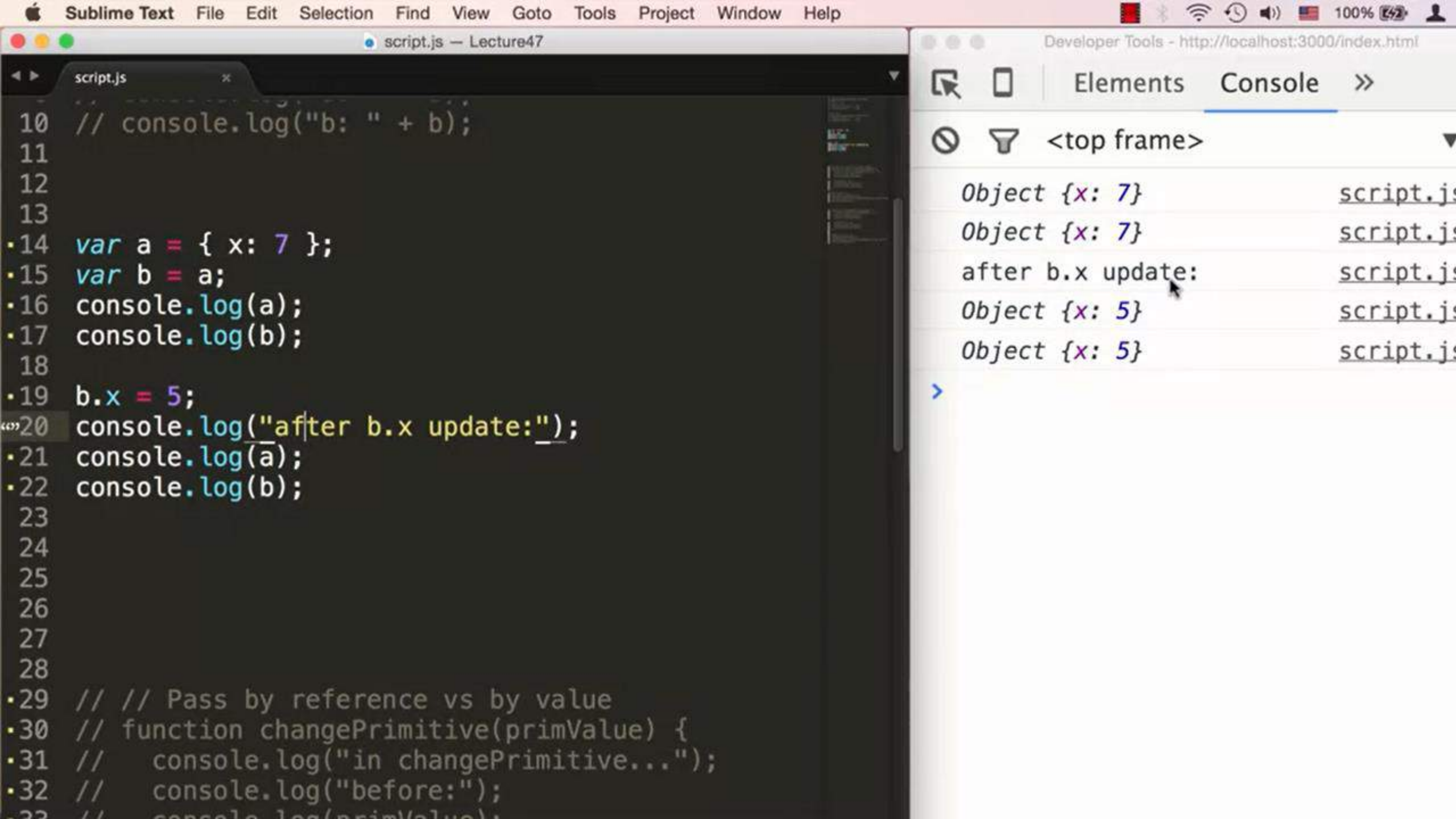
```
var a = {x: 7};
var b = a;
b.x = 5;
```

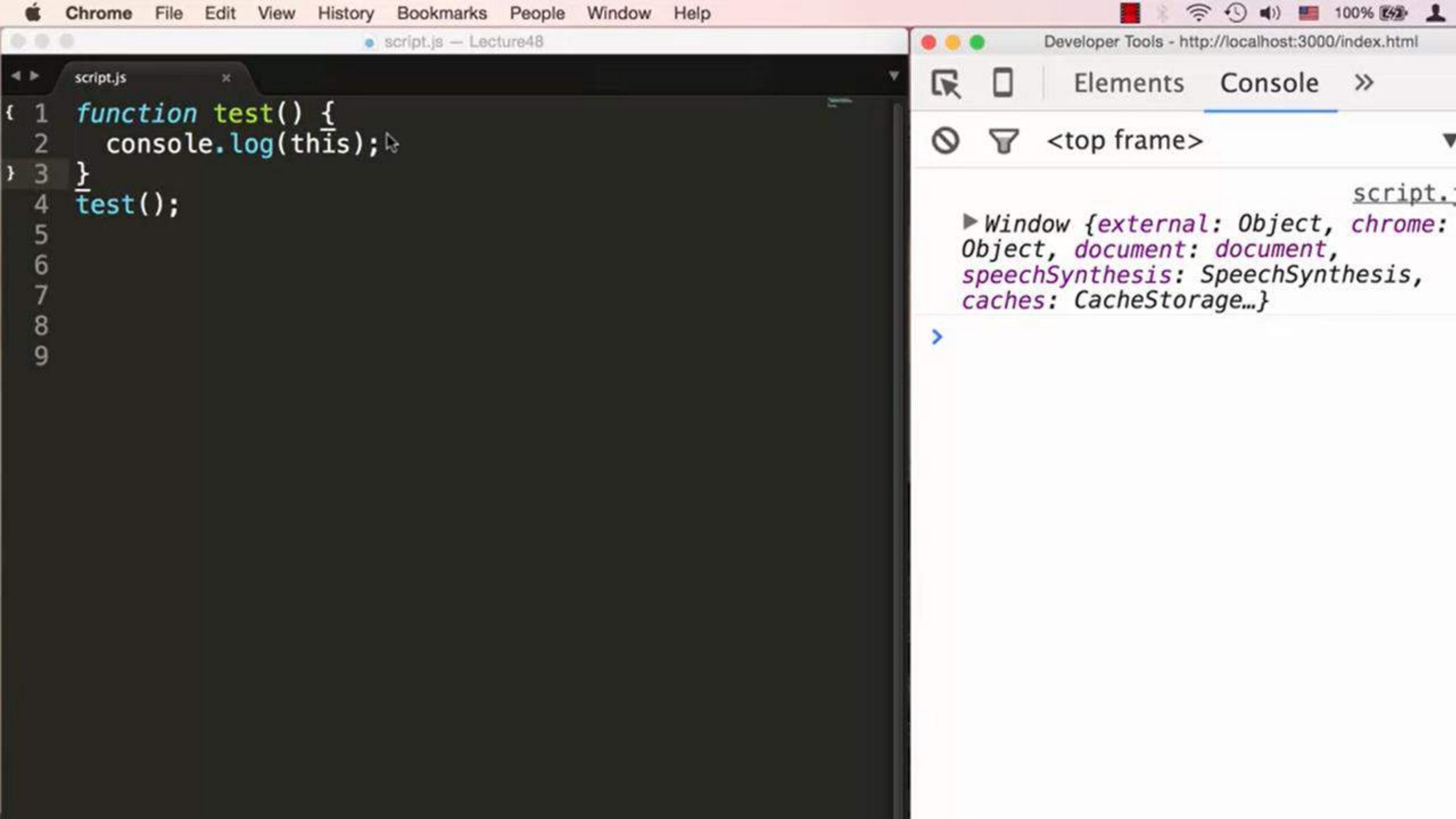


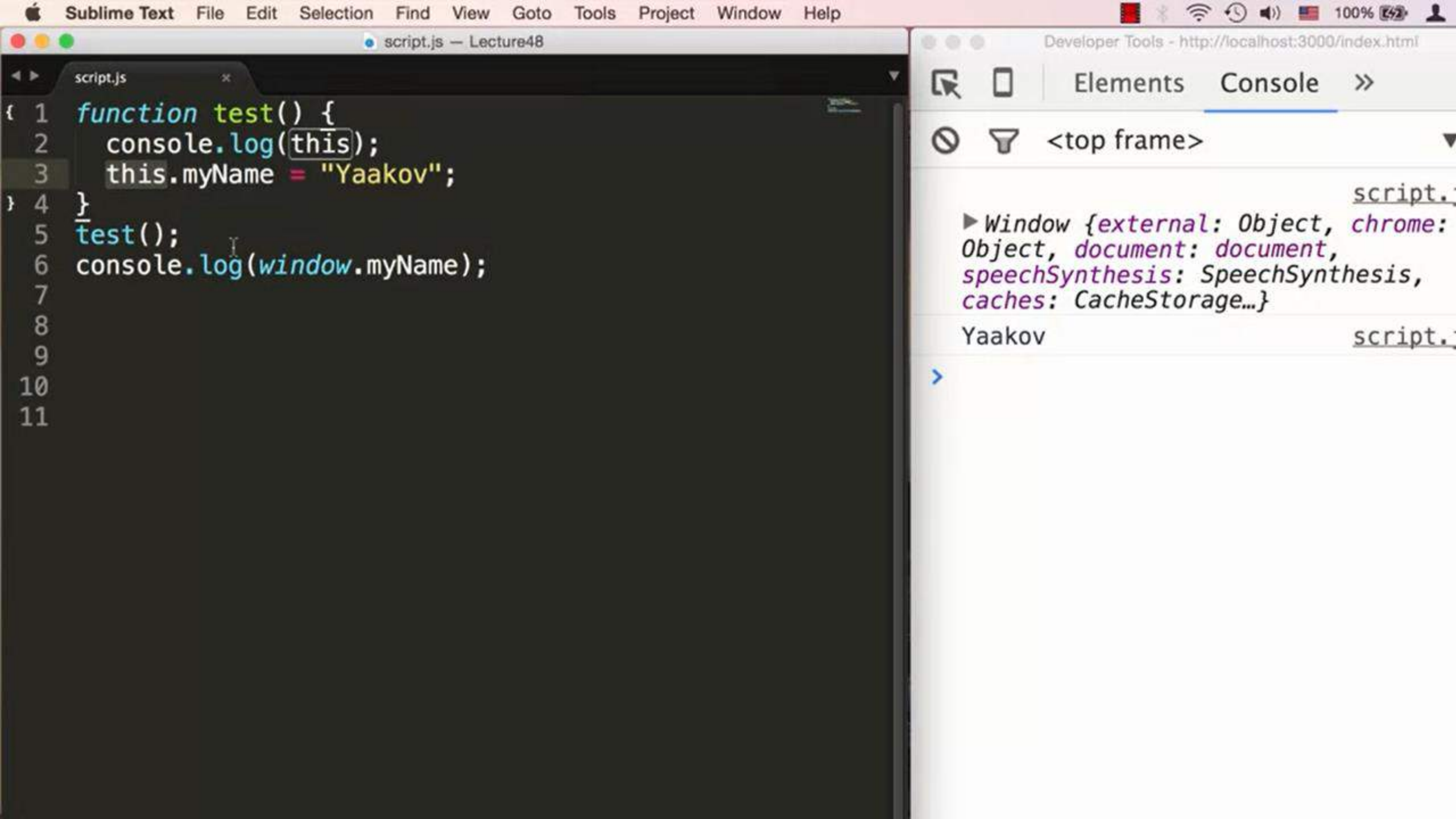

Sublime Text File Edit Selection Find View Goto Tools Project Window Help
script.js — Lecture47
script.js
// Copy by Reference vs by Value
var a = 7;
var b = a;
console.log("a: " + a);
console.log("b: " + b);
// b = 5;
// console.log("after b update:");
// console.log("a: " + a);
// console.log("b: " + b);
// var a = { x: 7 };
// var b = a;
// console.log(a);
// console.log(b);
// b.x = 5;
// console.log("after b.x update:");
// console.log(a);
// console.log(b);
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
a: 7
b: 7

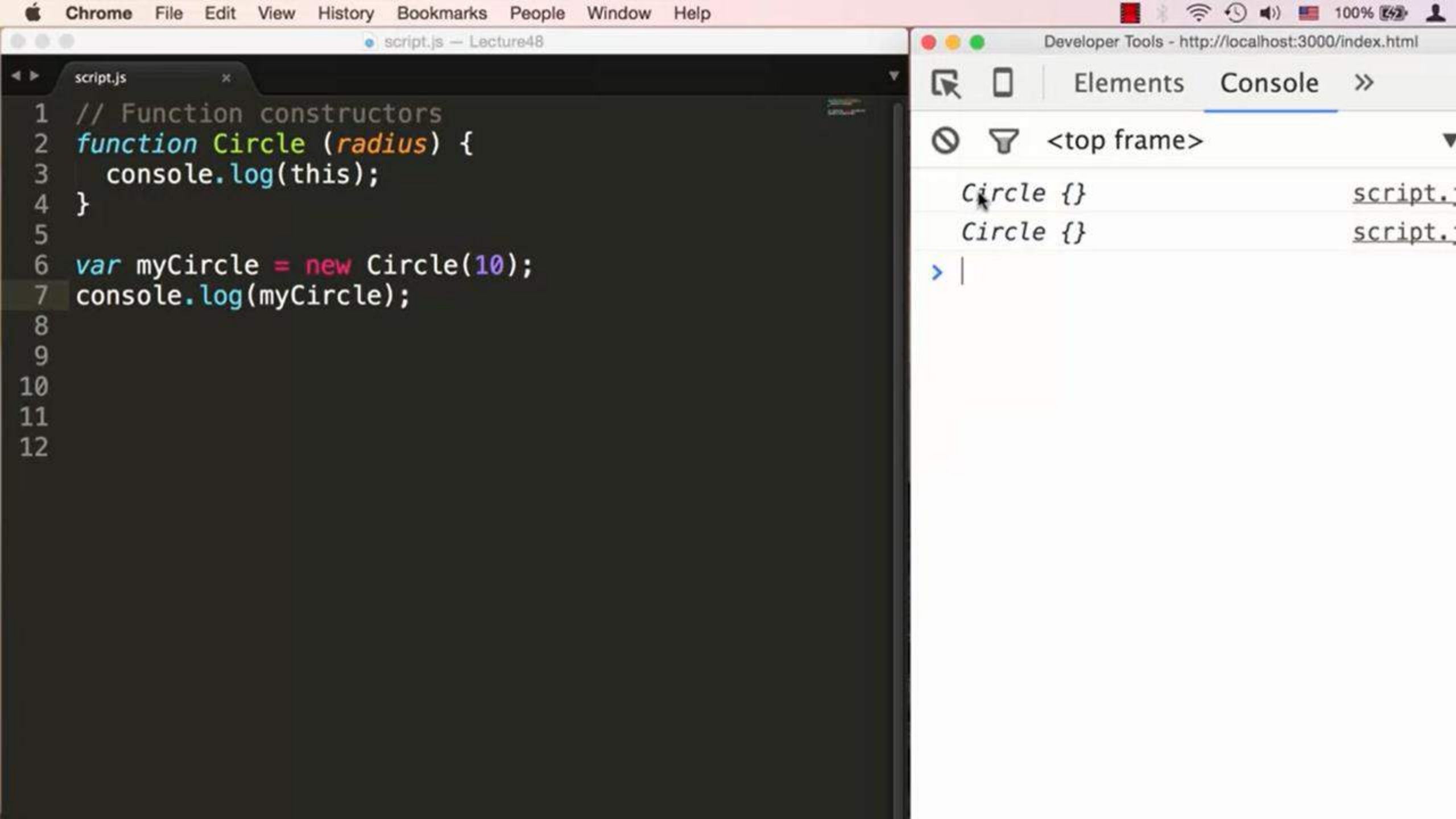

Chrome File Edit View History Bookmarks People Window Help
100%
script.js — Lecture47
script.js
1 // Copy by Reference vs by Value
2 var a = 7;
3 var b = a;
4 console.log("a: " + a);
5 console.log("b: " + b);
6
7 b = 5;
8 console.log("after b update:");
9 console.log("a: " + a);
10 console.log("b: " + b);
11
12
13
14 // var a = { x: 7 };
15 // var b = a;
16 // console.log(a);
17 // console.log(b);
18
19 // b.x = 5;
20 // console.log("after b.x update:");
21 // console.log(a);
22 // console.log(b);
23
24
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
a: 7
b: 7
after b update:
a: 7
b: 5
script.j
script.j
script.j
script.j
script.js

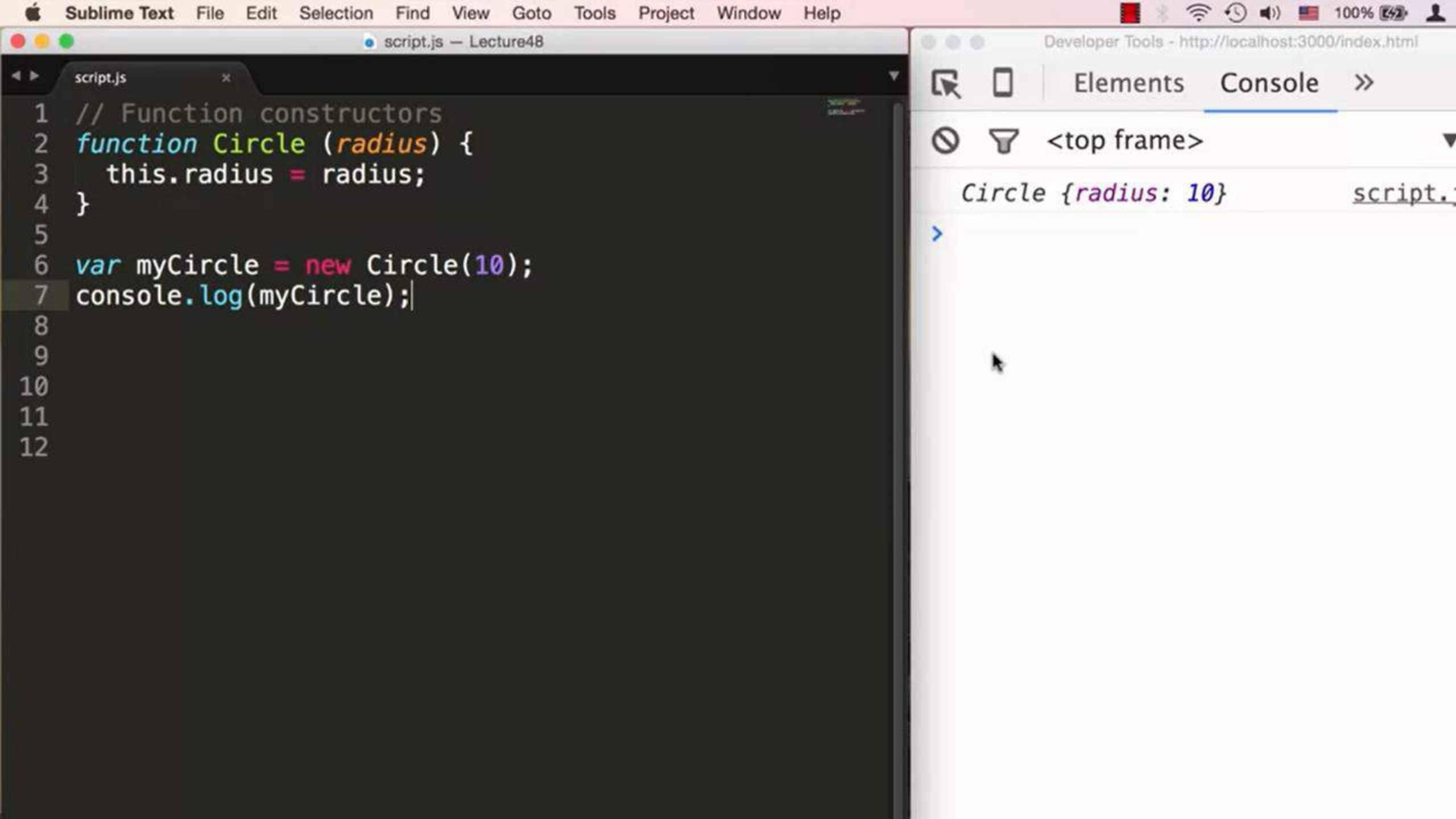

Sublime Text File Edit Selection Find View Goto Tools Project Window Help
100%
script.js — Lecture47
script.js
// console.log("b: " + b);
var a = { x: 7 };
var b = a;
console.log(a);
console.log(b);
// b.x = 5;
// console.log("after b.x update:");
// console.log(a);
// console.log(b);
// // Pass by reference vs by value
// function changePrimitive(primValue) {
//   console.log("in changePrimitive...");
//   console.log("before:");
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
Object {x: 7}
script.js
Object {x: 7}
script.js

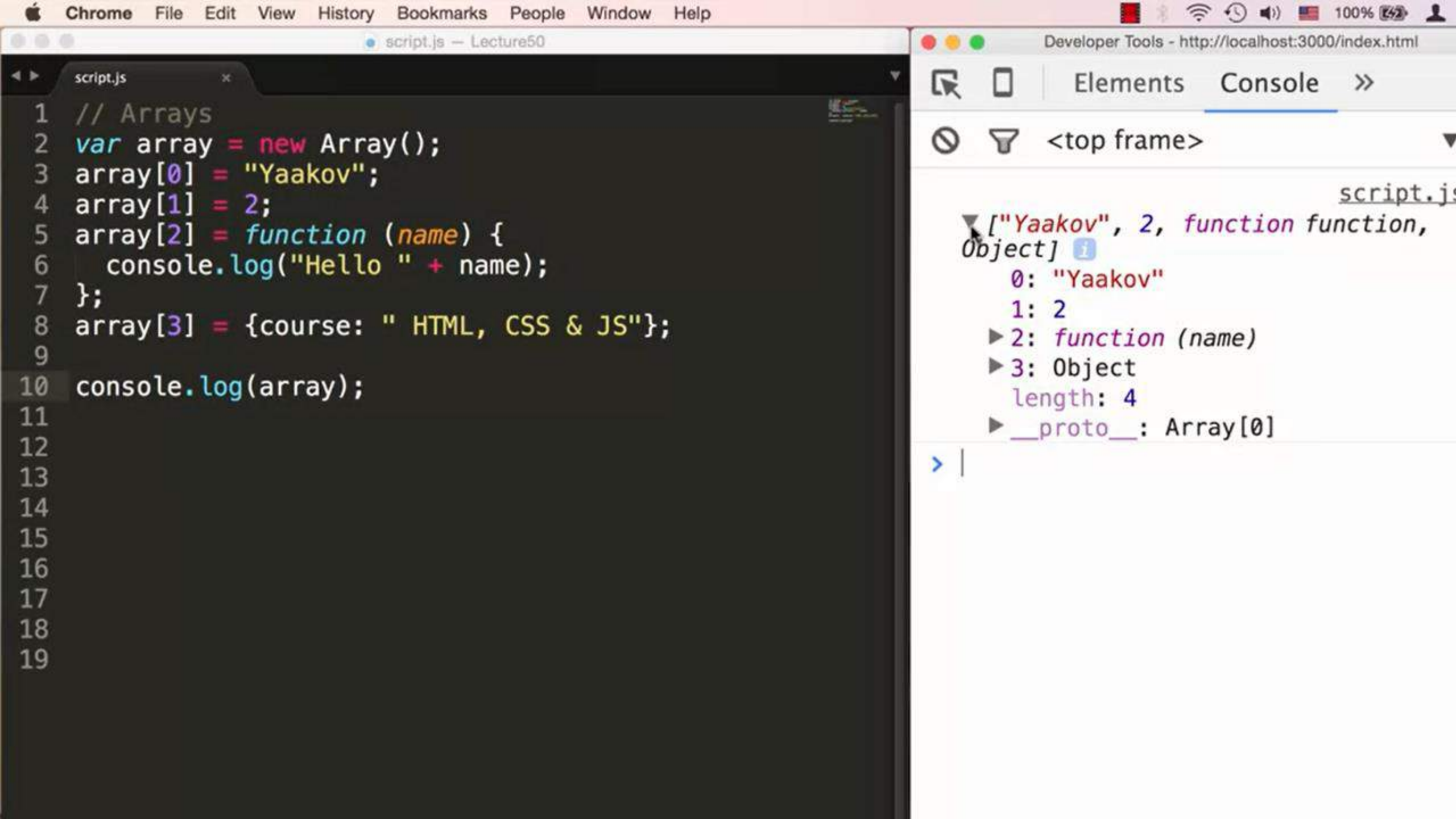

Sublime Text File Edit Selection Find View Goto Tools Project Window Help
script.js — Lecture47
script.js
// console.log("b: " + b);
var a = { x: 7 };
var b = a;
console.log(a);
console.log(b);
b.x = 5;
console.log("after b.x update:");
console.log(a);
console.log(b);
// // Pass by reference vs by value
// function changePrimitive(primValue) {
//   console.log("in changePrimitive...");
//   console.log("before:");
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
Object {x: 7}
Object {x: 7}
after b.x update:
Object {x: 5}
Object {x: 5}
script.js

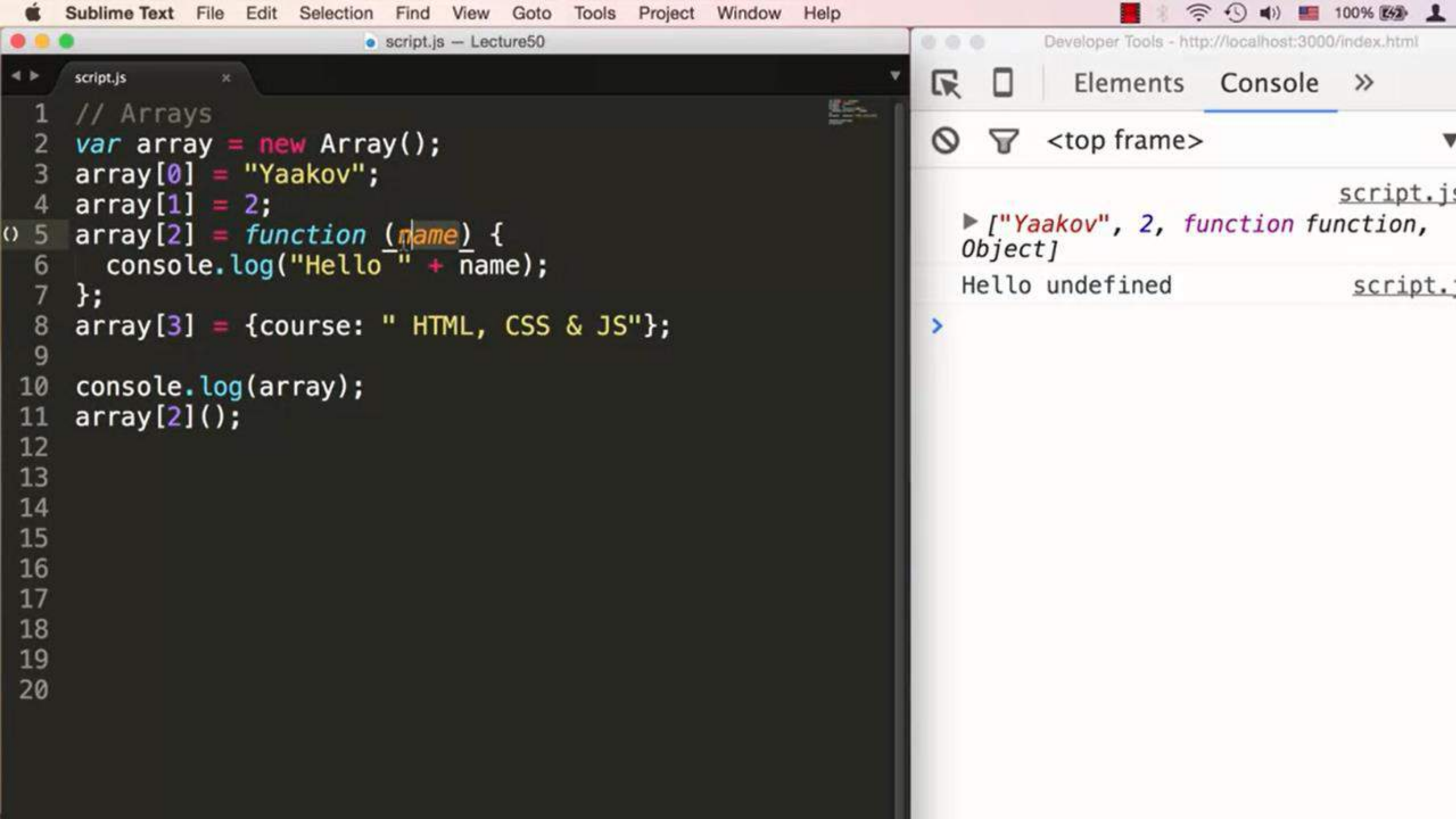

Chrome File Edit View History Bookmarks People Window Help
100%
script.js — Lecture48
script.js
1 function test() {
2 console.log(this);
3 }
4 test();
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
script.
▶Window {external: Object, chrome: Object, document: document, speechSynthesis: SpeechSynthesis, caches: CacheStorage…}
>

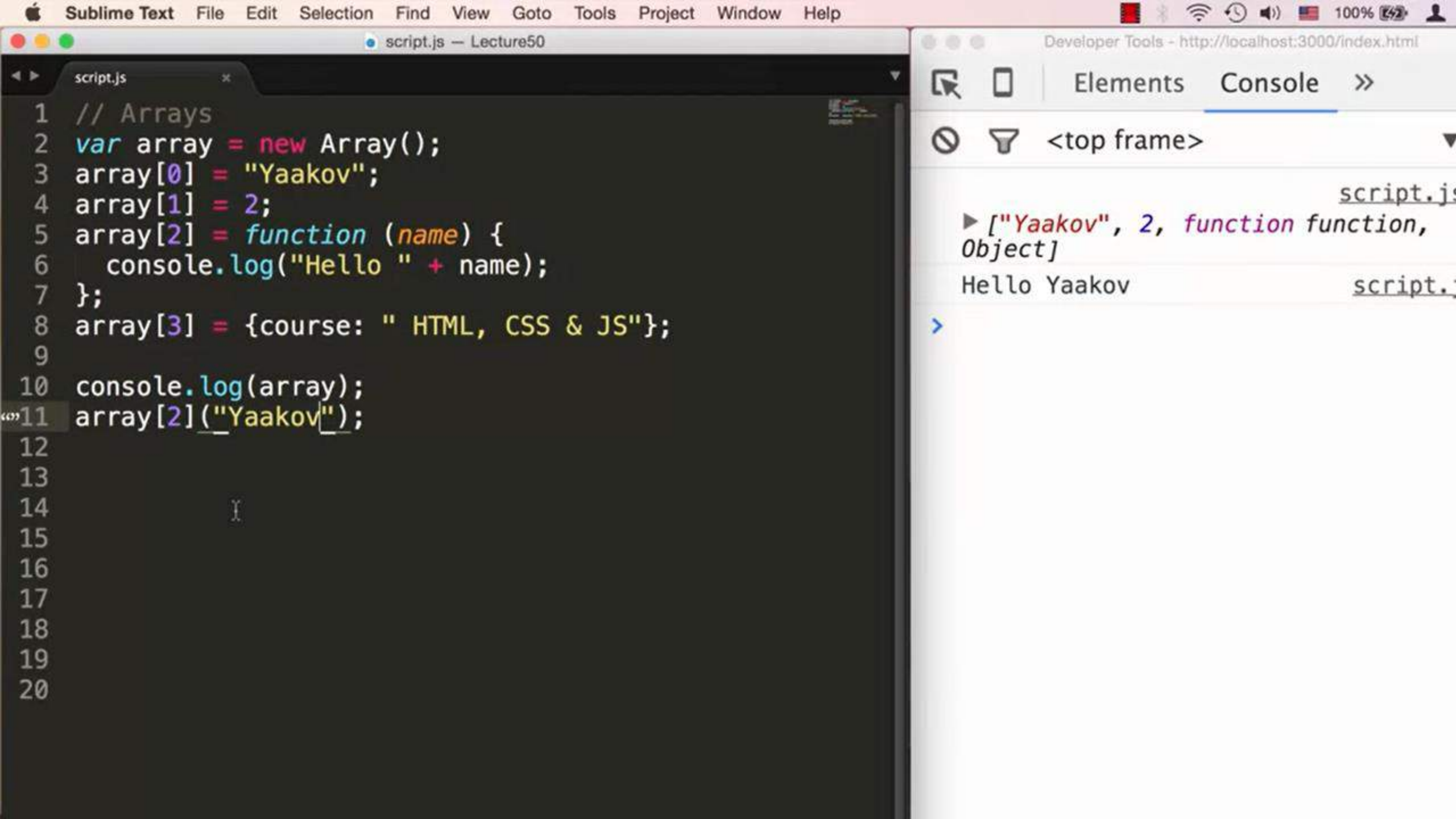

Sublime Text File Edit Selection Find View Goto Tools Project Window Help
100%
script.js — Lecture48
script.js
function test() {
console.log(this);
this.myName = "Yaakov";
}
test();
console.log(window.myName);
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
script.
Window {external: Object, chrome: Object, document: document, speechSynthesis: SpeechSynthesis, caches: CacheStorage…}
Yaakov
script.

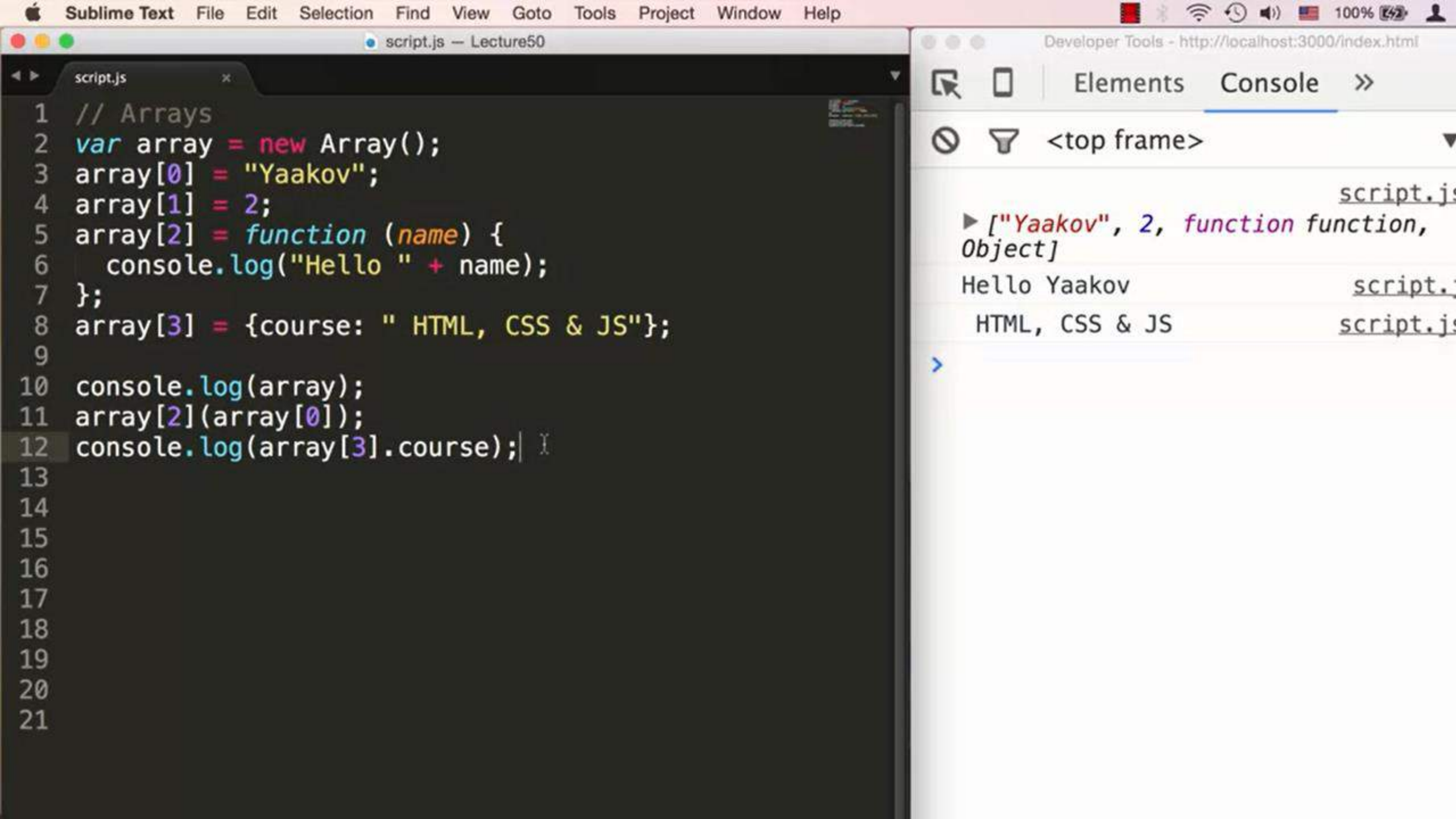

Chrome File Edit View History Bookmarks People Window Help
100%
script.js — Lecture48
script.js
1 // Function constructors
2 function Circle (radius) {
3 console.log(this);
4 }
5
6 var myCircle = new Circle(10);
7 console.log(myCircle);
8
9
10
11
12
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
Circle {}
script.
Circle {}
script.
>

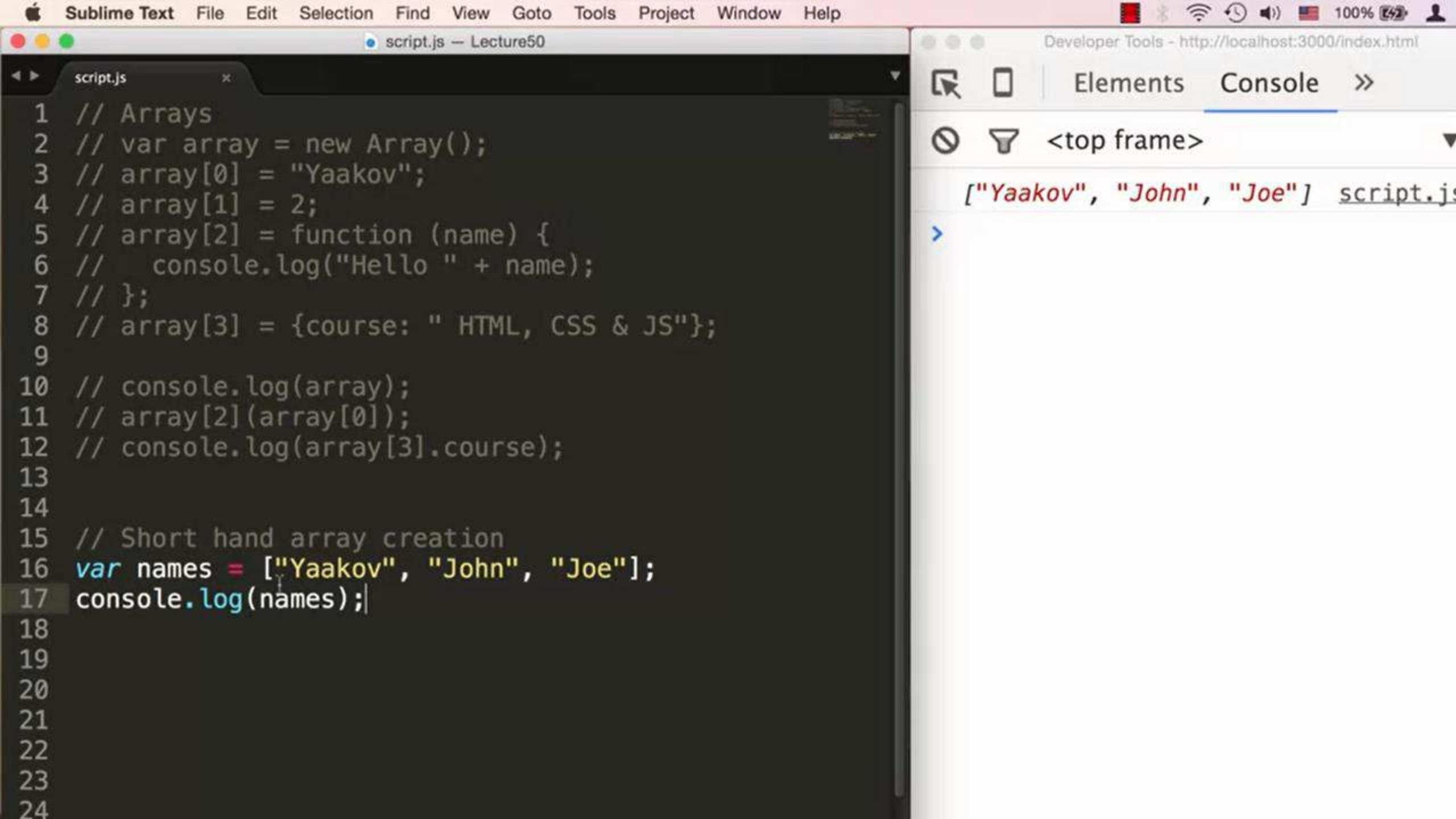

Sublime Text File Edit Selection Find View Goto Tools Project Window Help
100%
script.js — Lecture48
script.js
1 // Function constructors
2 function Circle (radius) {
3 this.radius = radius;
4 }
5
6 var myCircle = new Circle(10);
7 console.log(myCircle);
8
9
10
11
12
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
Circle {radius: 10}
script.
>

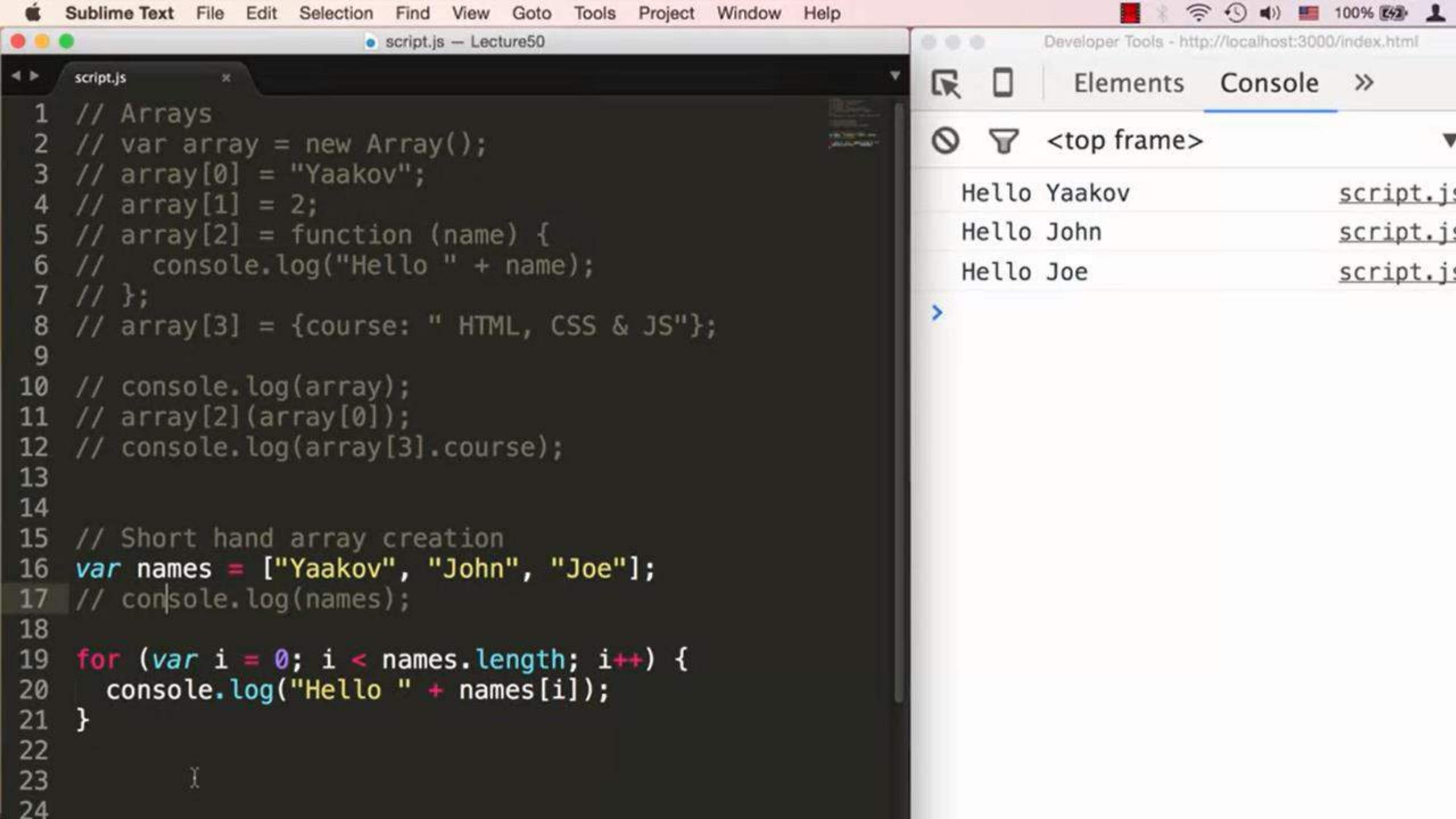

Chrome File Edit View History Bookmarks People Window Help
script.js — Lecture50
script.js
// Arrays
var array = new Array();
array[0] = "Yaakov";
array[1] = 2;
array[2] = function (name) {
console.log("Hello " + name);
};
array[3] = {course: " HTML, CSS & JS"};
console.log(array);
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
script.js
["Yaakov", 2, function function, Object]
0: "Yaakov"
1: 2
2: function (name)
3: Object
length: 4
__proto__: Array[0]

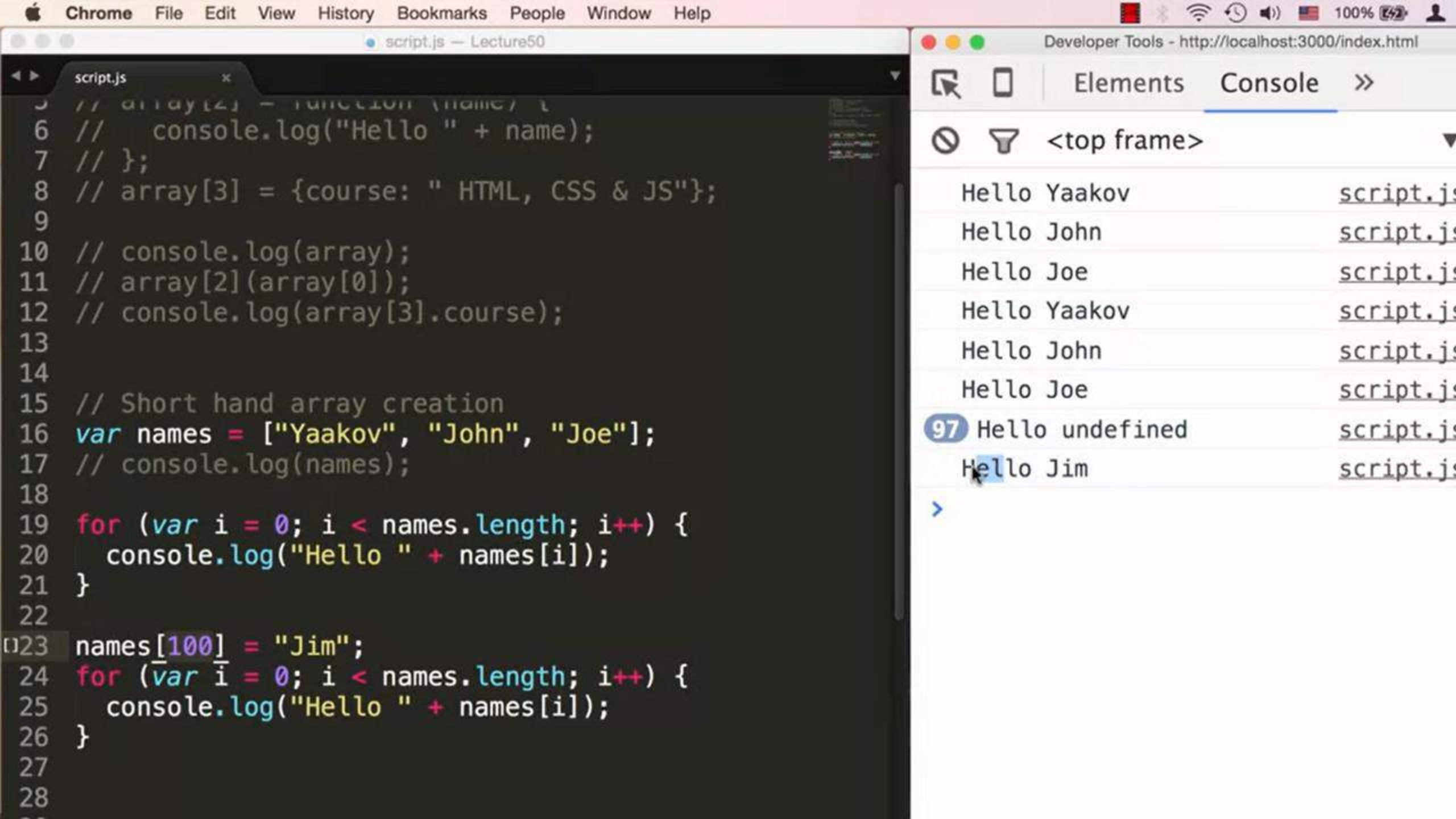

Sublime Text File Edit Selection Find View Goto Tools Project Window Help
script.js — Lecture50
script.js
// Arrays
var array = new Array();
array[0] = "Yaakov";
array[1] = 2;
array[2] = function (name) {
console.log("Hello " + name);
};
array[3] = {course: " HTML, CSS & JS"};
console.log(array);
array[2]();
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
script.js
["Yaakov", 2, function function, Object]
Hello undefined
script.

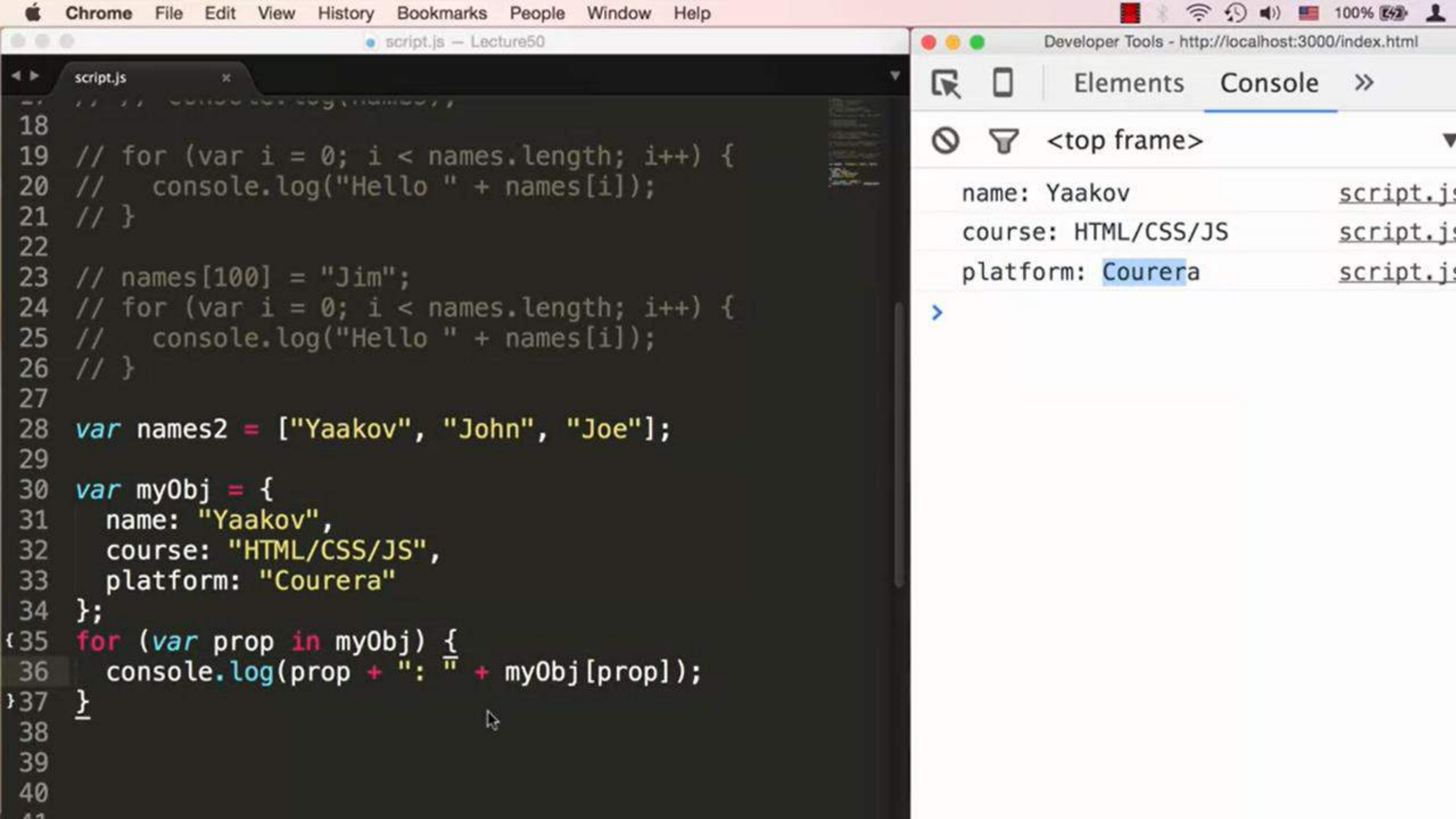

Sublime Text File Edit Selection Find View Goto Tools Project Window Help
script.js — Lecture50
script.js
// Arrays
var array = new Array();
array[0] = "Yaakov";
array[1] = 2;
array[2] = function (name) {
console.log("Hello " + name);
};
array[3] = {course: " HTML, CSS & JS"};
console.log(array);
array[2]("Yaakov");
Developer Tools - http://localhost:3000/index.html
Elements Console
<top frame>
script.js
["Yaakov", 2, function function, Object]
Hello Yaakov
script.

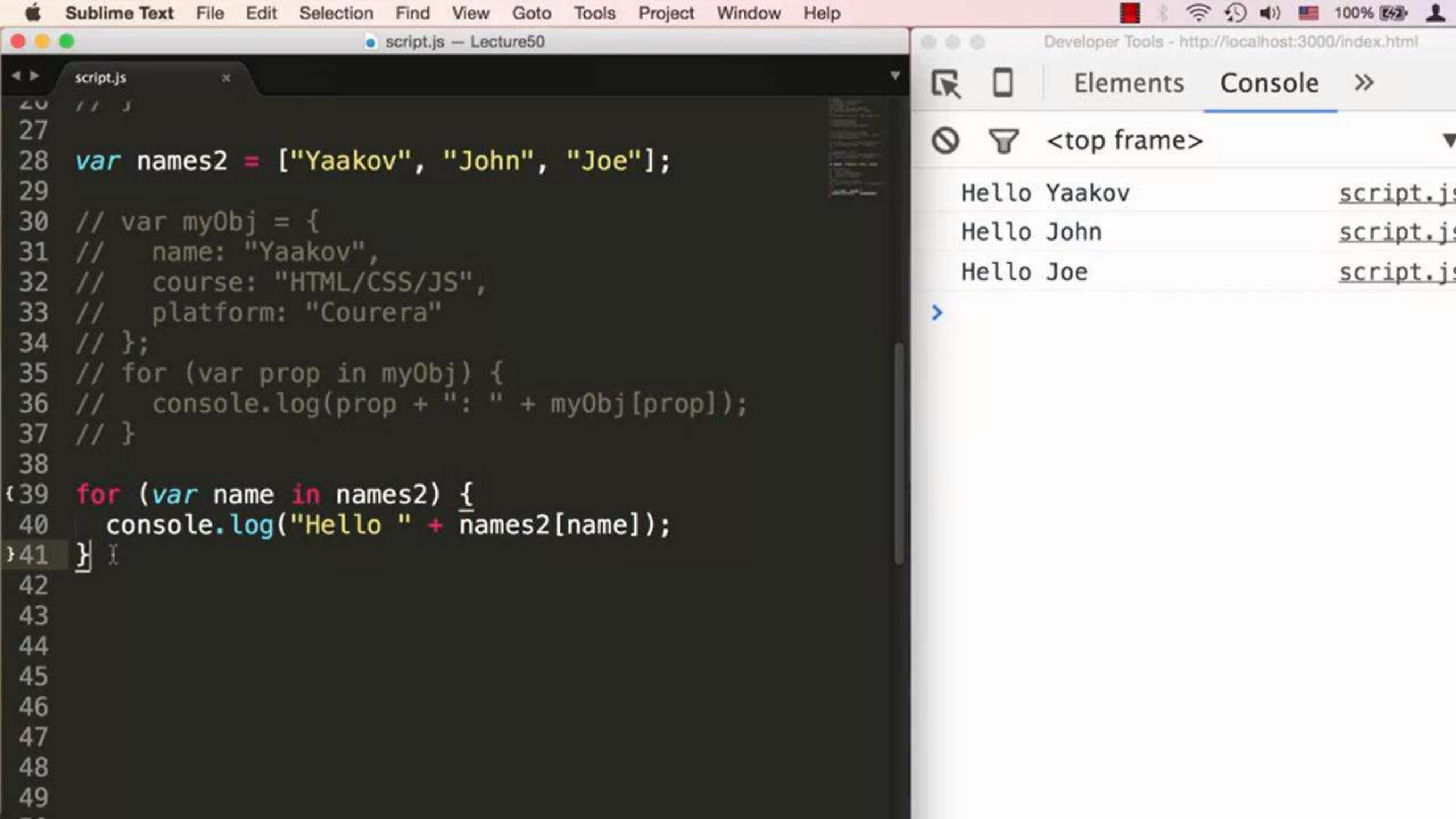

Sublime Text File Edit Selection Find View Goto Tools Project Window Help
script.js — Lecture50
script.js
// Arrays
var array = new Array();
array[0] = "Yaakov";
array[1] = 2;
array[2] = function (name) {
console.log("Hello " + name);
};
array[3] = {course: " HTML, CSS & JS"};
console.log(array);
array[2](array[0]);
console.log(array[3].course);
Developer Tools - http://localhost:3000/index.html
Elements Console
<top frame>
script.js
["Yaakov", 2, function function, Object]
Hello Yaakov
HTML, CSS & JS

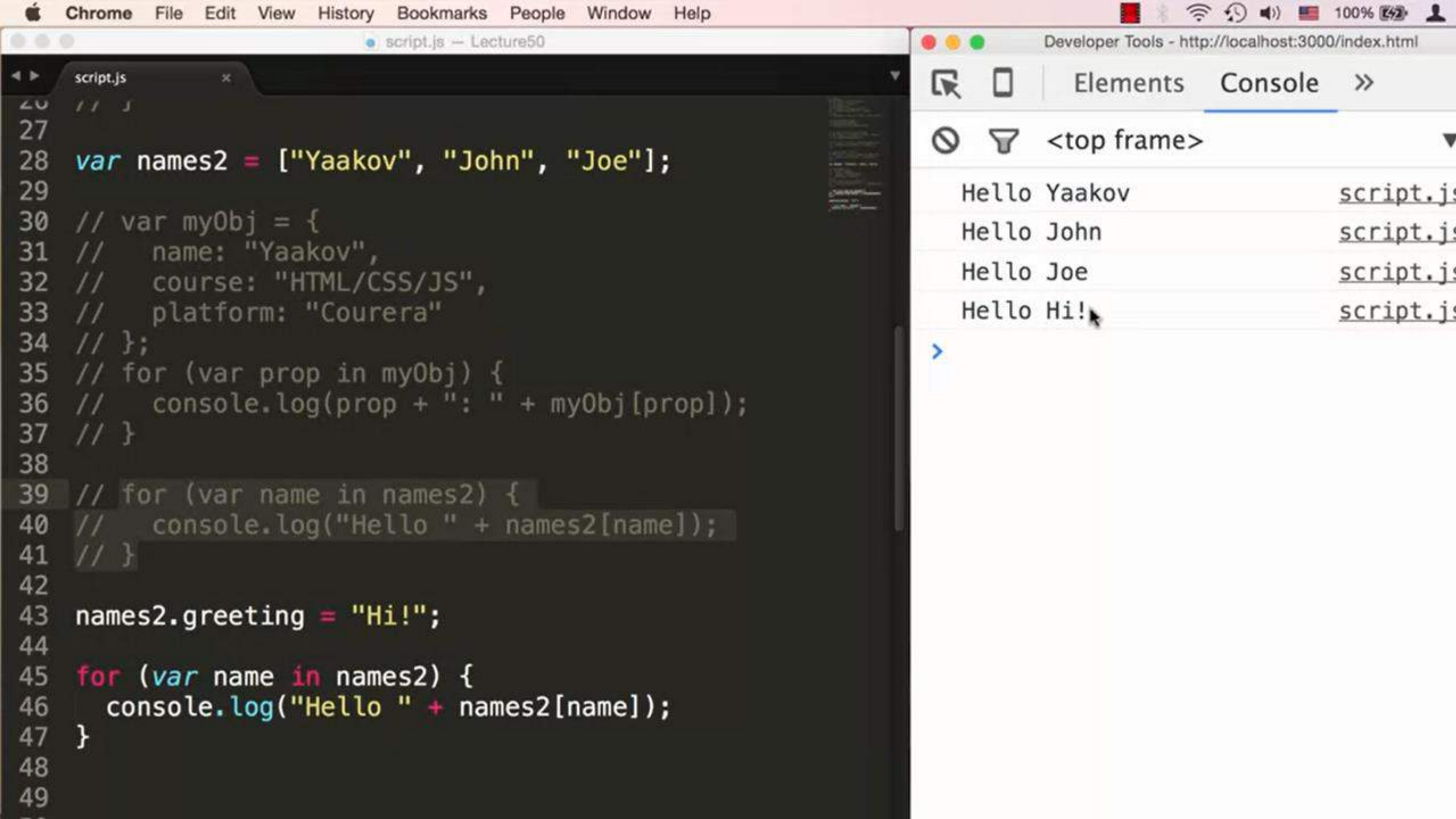

Sublime Text File Edit Selection Find View Goto Tools Project Window Help
script.js — Lecture50
script.js
// Arrays
// var array = new Array();
// array[0] = "Yaakov";
// array[1] = 2;
// array[2] = function (name) {
// console.log("Hello " + name);
// };
// array[3] = {course: " HTML, CSS & JS"};
// console.log(array);
// array[2](array[0]);
// console.log(array[3].course);
// Short hand array creation
var names = ["Yaakov", "John", "Joe"];
console.log(names);
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
["Yaakov", "John", "Joe"] script.js
>


Sublime Text File Edit Selection Find View Goto Tools Project Window Help
script.js — Lecture50
script.js
// Arrays
// var array = new Array();
// array[0] = "Yaakov";
// array[1] = 2;
// array[2] = function (name) {
//   console.log("Hello " + name);
// };
// array[3] = {course: " HTML, CSS & JS"};
// console.log(array);
// array[2](array[0]);
// console.log(array[3].course);
// Short hand array creation
var names = ["Yaakov", "John", "Joe"];
// console.log(names);
for (var i = 0; i < names.length; i++) {
  console.log("Hello " + names[i]);
}
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
Hello Yaakov
script.js
Hello John
script.js
Hello Joe
script.js


Chrome File Edit View History Bookmarks People Window Help
script.js — Lecture50
script.js
// console.log("Hello " + name);
// };
// array[3] = {course: " HTML, CSS & JS"};
// console.log(array);
// array[2](array[0]);
// console.log(array[3].course);
// Short hand array creation
var names = ["Yaakov", "John", "Joe"];
// console.log(names);
for (var i = 0; i < names.length; i++) {
console.log("Hello " + names[i]);
}
names[100] = "Jim";
for (var i = 0; i < names.length; i++) {
console.log("Hello " + names[i]);
}
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
Hello Yaakov script.js
Hello John script.js
Hello Joe script.js
Hello Yaakov script.js
Hello John script.js
Hello Joe script.js
97 Hello undefined script.js
Hello Jim script.js


Chrome File Edit View History Bookmarks People Window Help
script.js — Lecture50
script.js
// for (var i = 0; i < names.length; i++) {
//   console.log("Hello " + names[i]);
// }
// names[100] = "Jim";
// for (var i = 0; i < names.length; i++) {
//   console.log("Hello " + names[i]);
// }
var names2 = ["Yaakov", "John", "Joe"];
var myObj = {
name: "Yaakov",
course: "HTML/CSS/JS",
platform: "Courera"
};
for (var prop in myObj) {
console.log(prop + ": " + myObj[prop]);
}
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
name: Yaakov
course: HTML/CSS/JS
platform: Courera
script.js
script.js
script.js


Sublime Text File Edit Selection Find View Goto Tools Project Window Help
100%
script.js — Lecture50
script.js
27
28 var names2 = ["Yaakov", "John", "Joe"];
29
30 // var myObj = {
31 //   name: "Yaakov",
32 //   course: "HTML/CSS/JS",
33 //   platform: "Courera"
34 // };
35 // for (var prop in myObj) {
36 //   console.log(prop + ": " + myObj[prop]);
37 // }
38
39 for (var name in names2) {
40   console.log("Hello " + names2[name]);
41 }
42
43
44
45
46
47
48
49
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
Hello Yaakov
script.js
Hello John
script.js
Hello Joe
script.js


Chrome File Edit View History Bookmarks People Window Help
100%
script.js — Lecture50
script.js
27
28 var names2 = ["Yaakov", "John", "Joe"];
29
30 // var myObj = {
31 //   name: "Yaakov",
32 //   course: "HTML/CSS/JS",
33 //   platform: "Courera"
34 // };
35 // for (var prop in myObj) {
36 //   console.log(prop + ": " + myObj[prop]);
37 // }
38
39 // for (var name in names2) {
40 //   console.log("Hello " + names2[name]);
41 // }
42
43 names2.greeting = "Hi!";
44
45 for (var name in names2) {
46   console.log("Hello " + names2[name]);
47 }
48
49
Developer Tools - http://localhost:3000/index.html
Elements
Console
<top frame>
Hello Yaakov script.js
Hello John script.js
Hello Joe script.js
Hello Hi! script.js
>