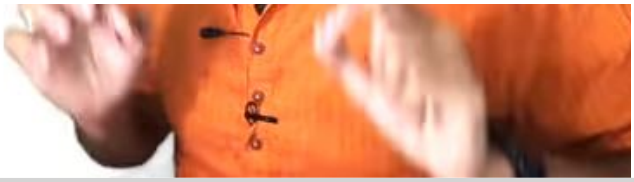
00:02

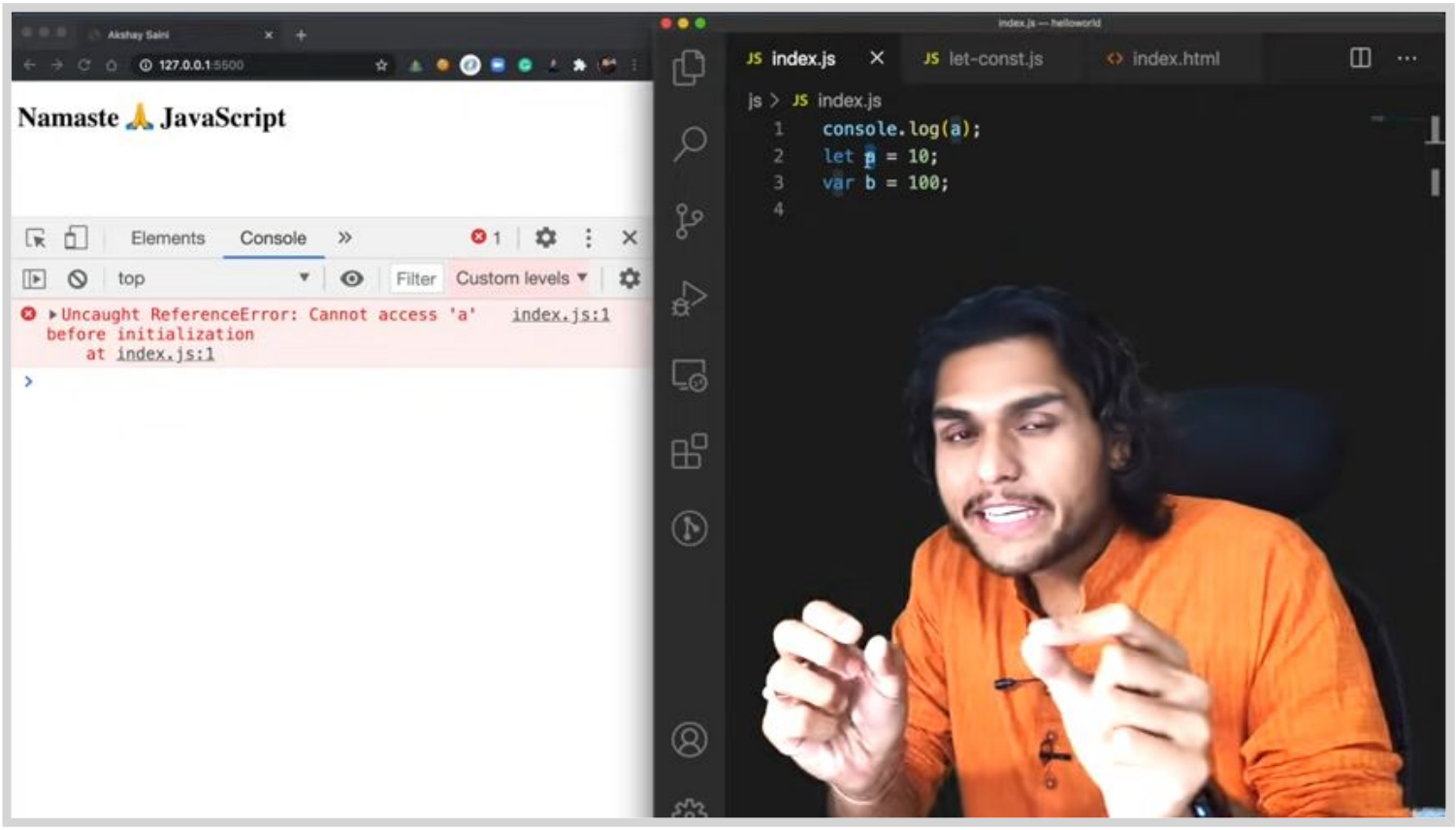
Temporal Dead zone is the time since when this "let" variable is hoisted & till is is initialised with some value.

00:05

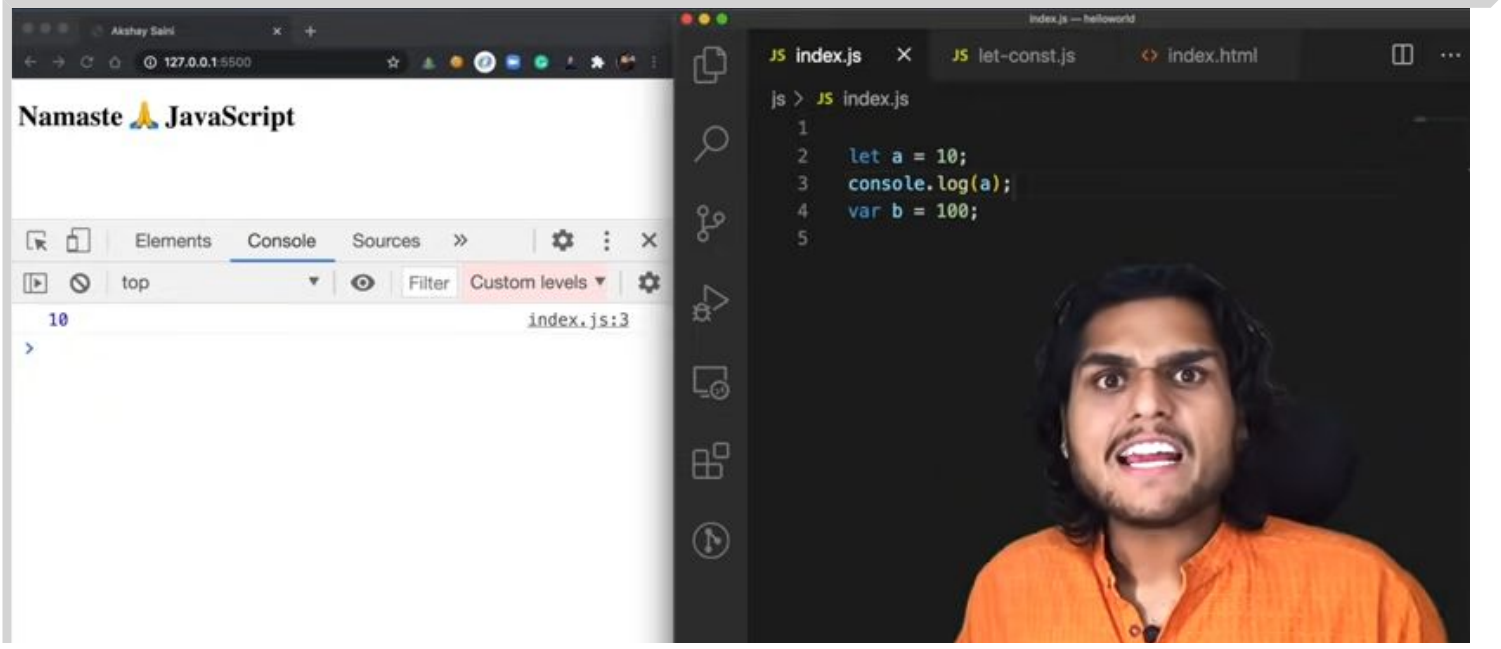


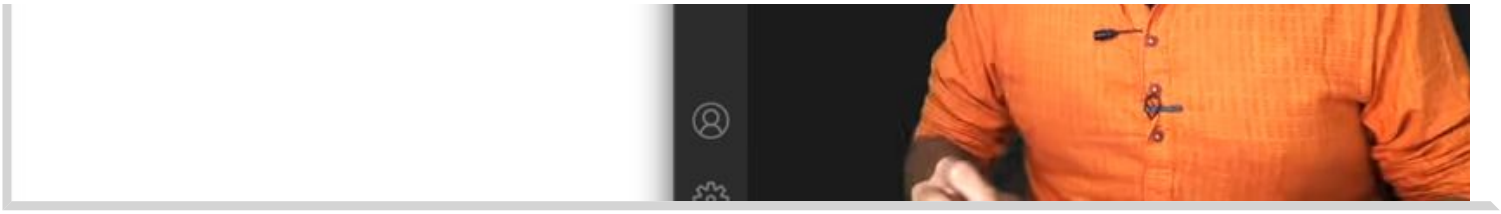
Yes, let&const declarations are hoisted.

02:34



02:44

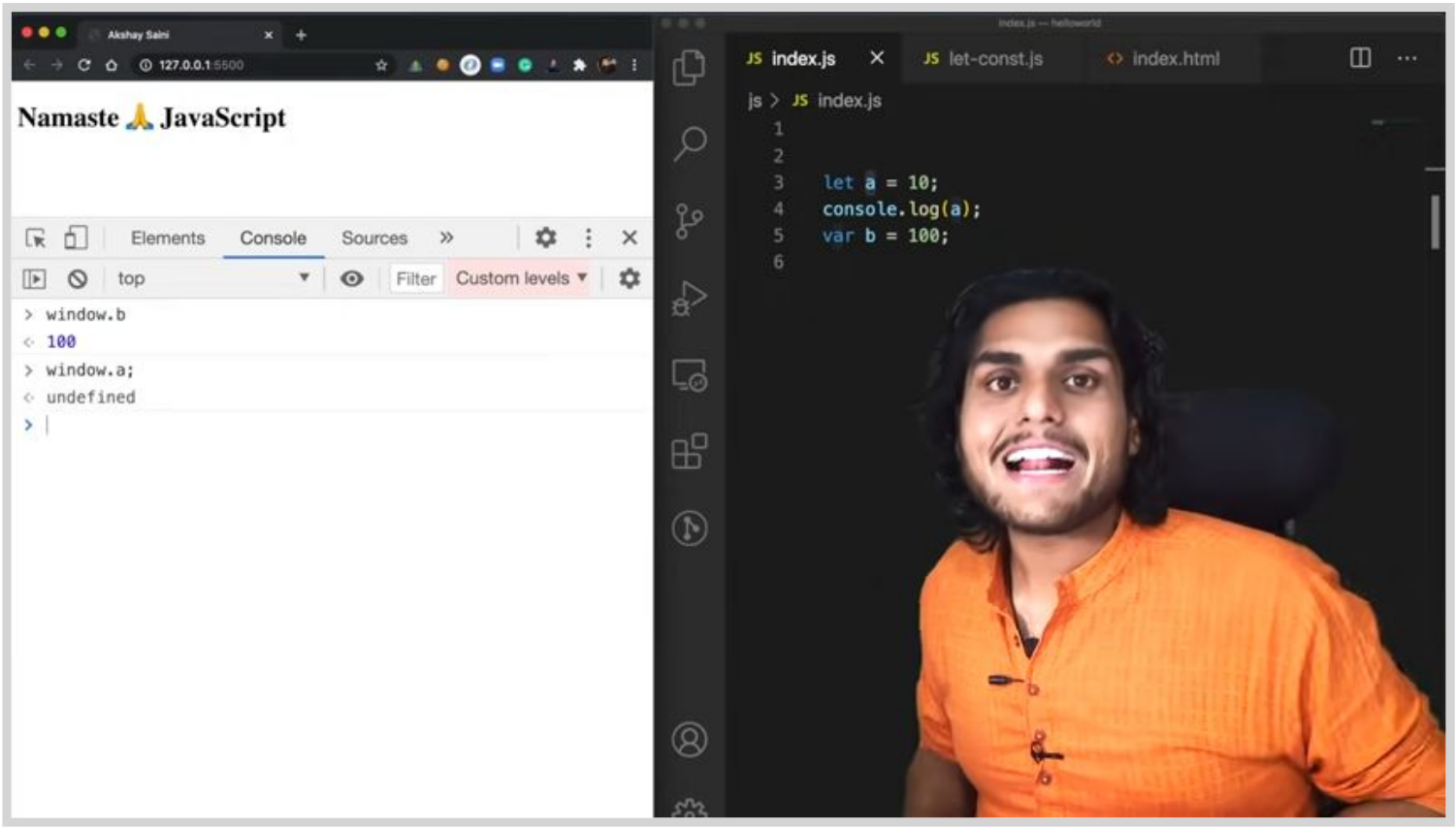




let & const are stored in a different memory space than Global. You cannot access them before you put any values in them.

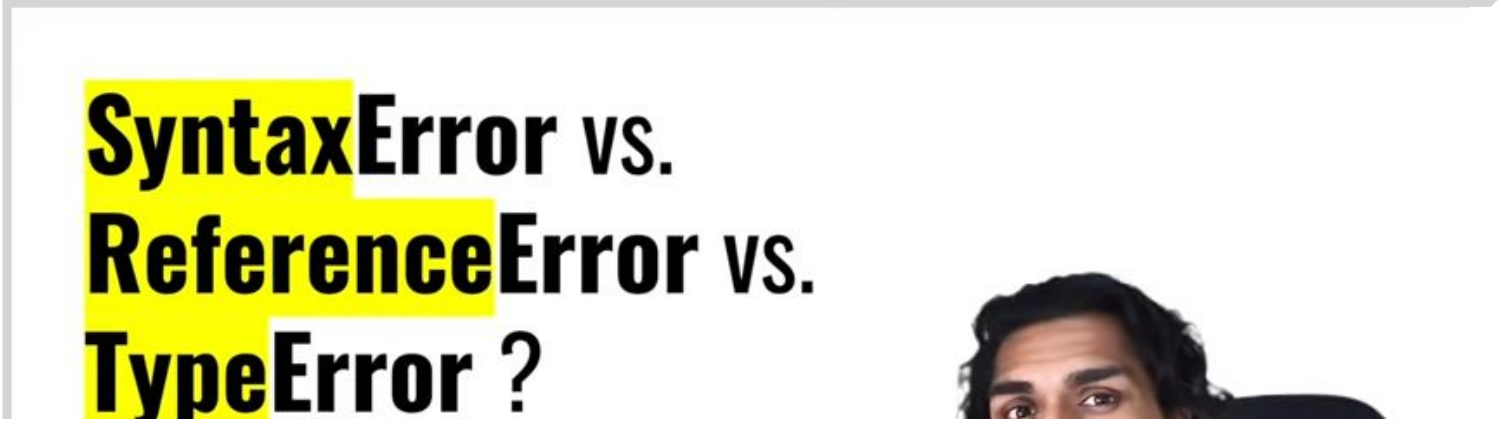
(in case of window - the "global object" is **Window**

10:23

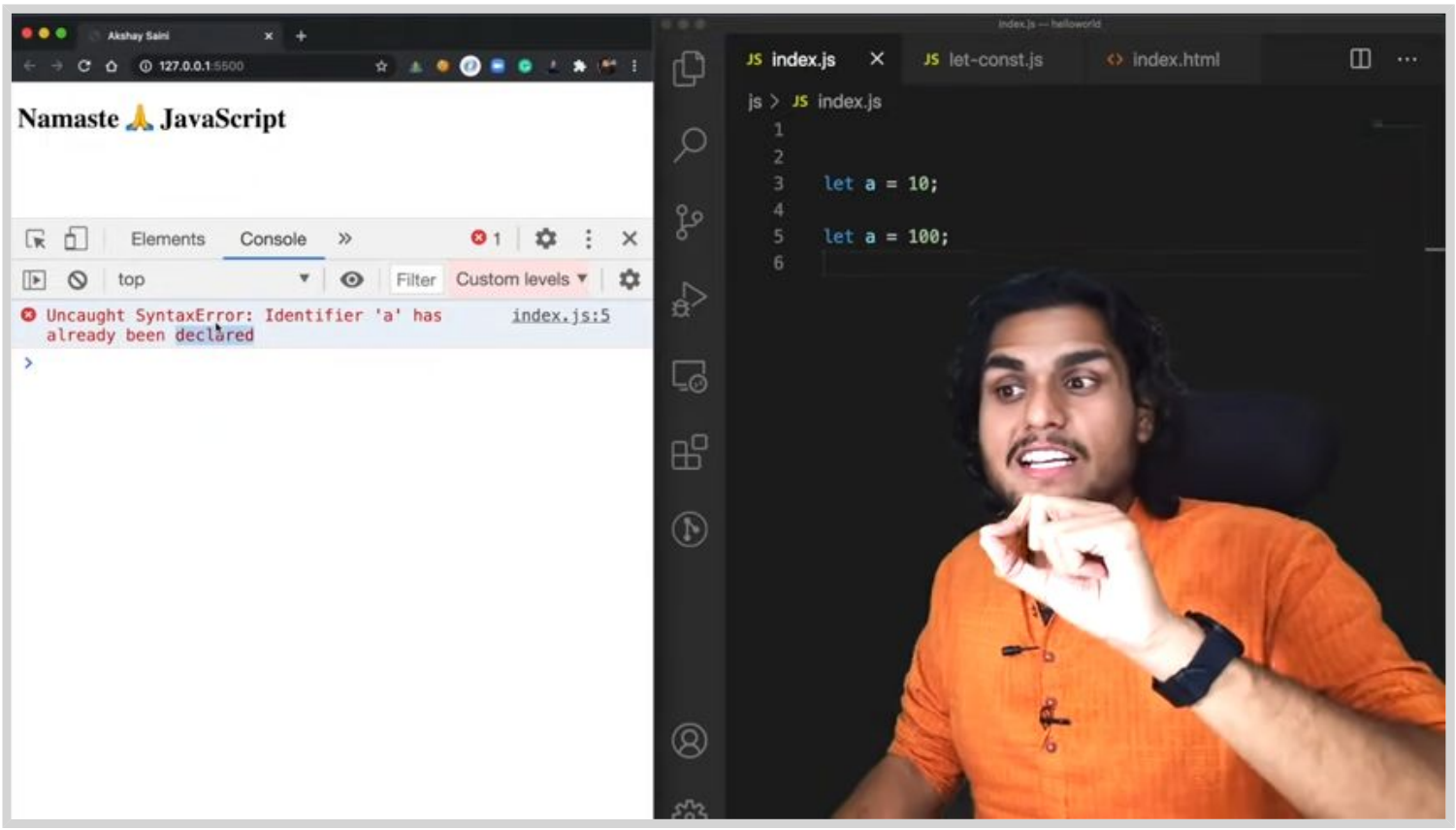


let a - is stored in a different memory space.

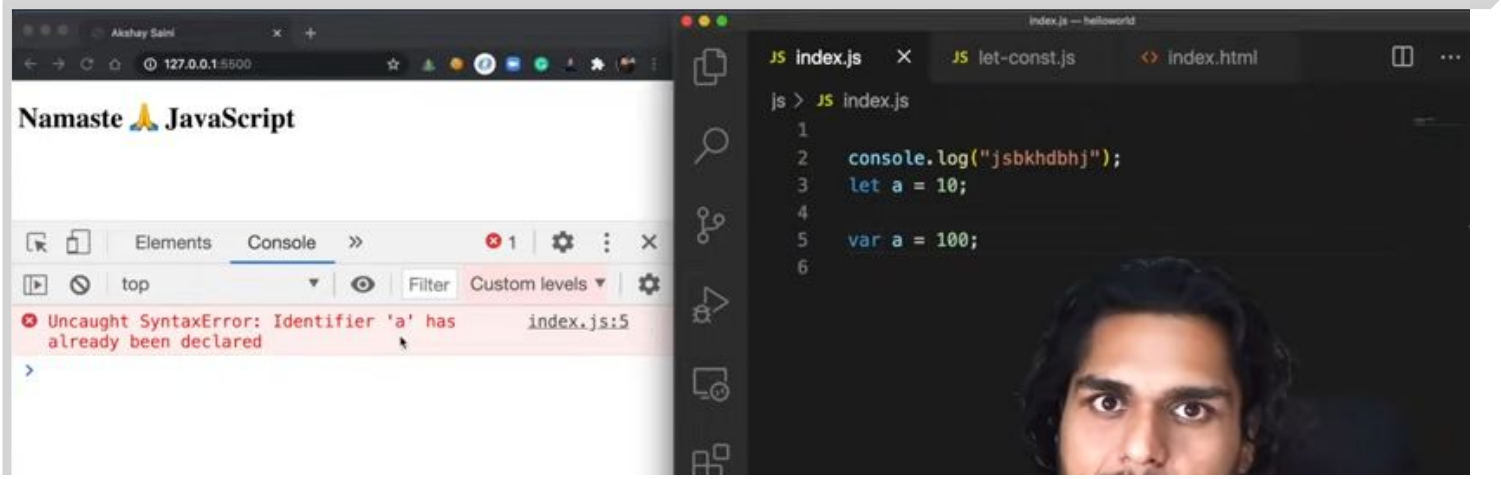
00:07

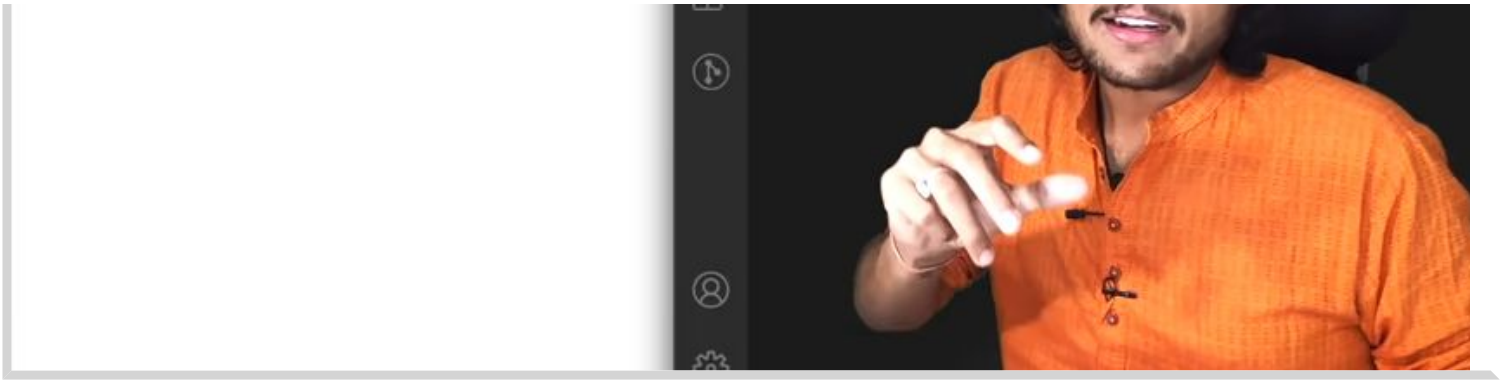


11:23


Akshay Saini
127.0.0.1:5500
Namaste 🙏 JavaScript
Elements
Console
top
Filter
Custom levels
Uncaught SyntaxError: Identifier 'a' has already been declared
index.js:5
index.js — helloworld
index.js
let-const.js
index.html
let a = 10;
let a = 100;

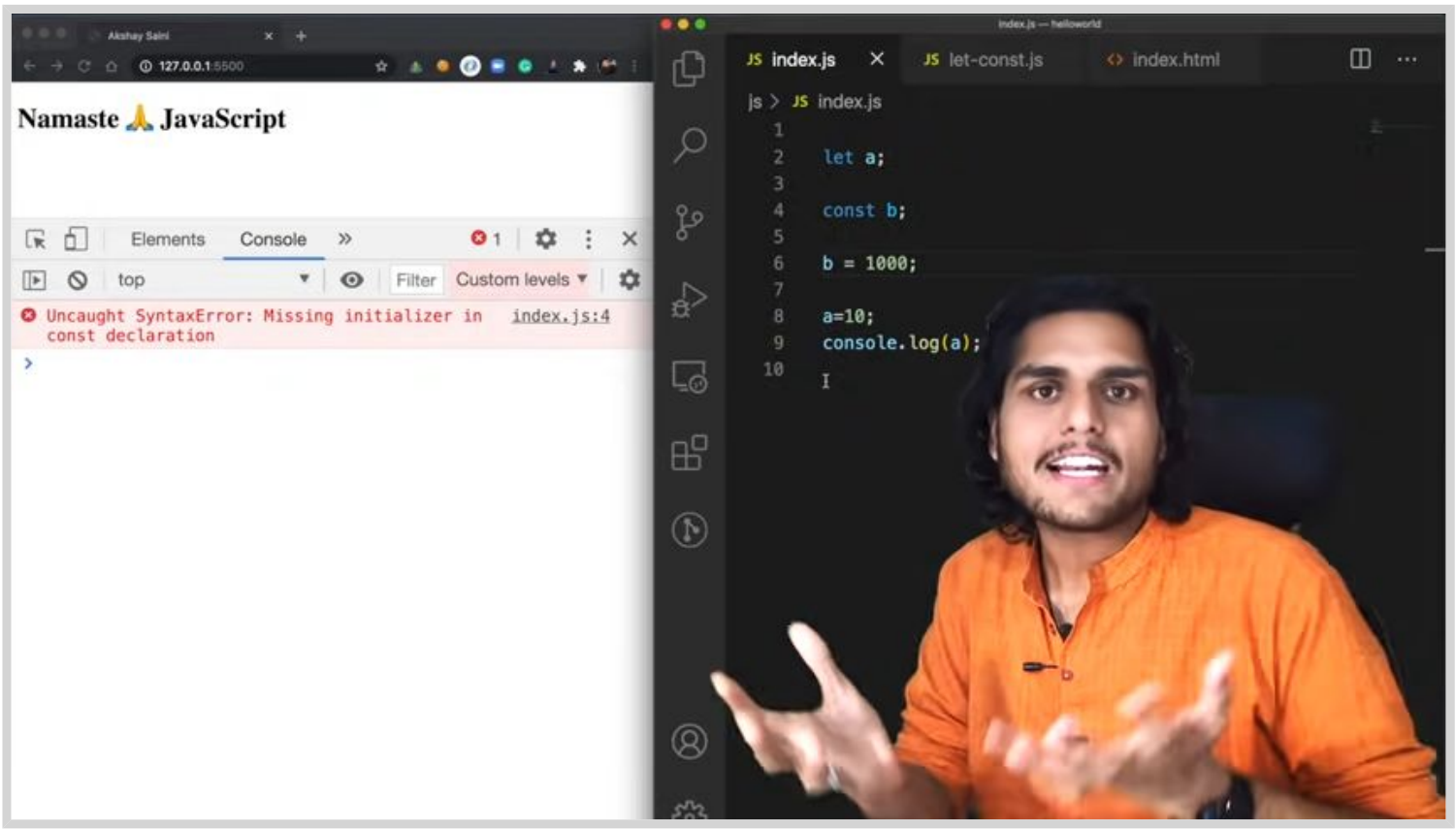
12:15


Akshay Saini
127.0.0.1:5500
Namaste 🙏 JavaScript
Elements
Console
top
Filter
Custom levels
Uncaught SyntaxError: Identifier 'a' has already been declared
index.js:5
index.js — helloworld
index.js
let-const.js
index.html
console.log("jsbkhdbhj");
let a = 10;
var a = 100;



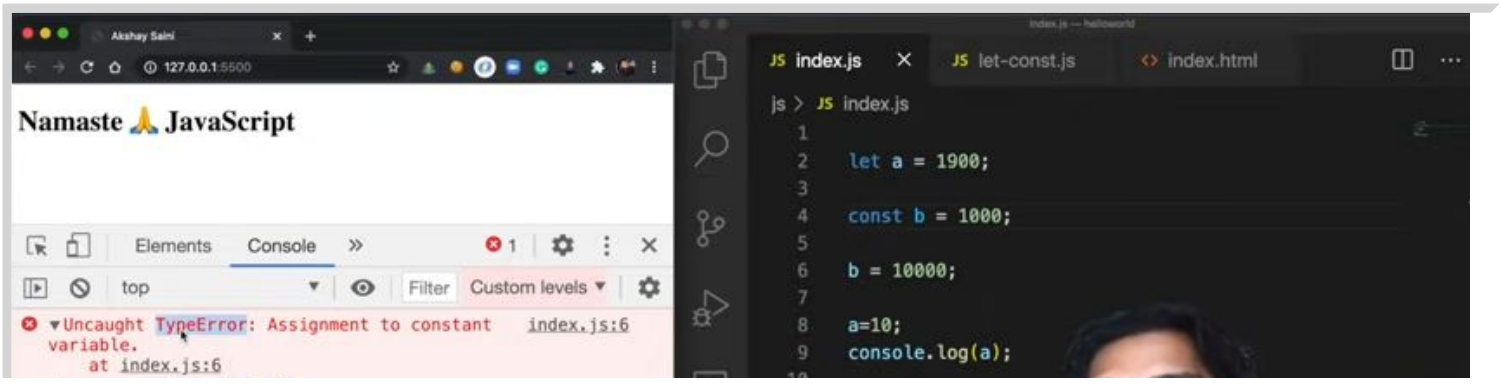
CONSTis even more strict than LET

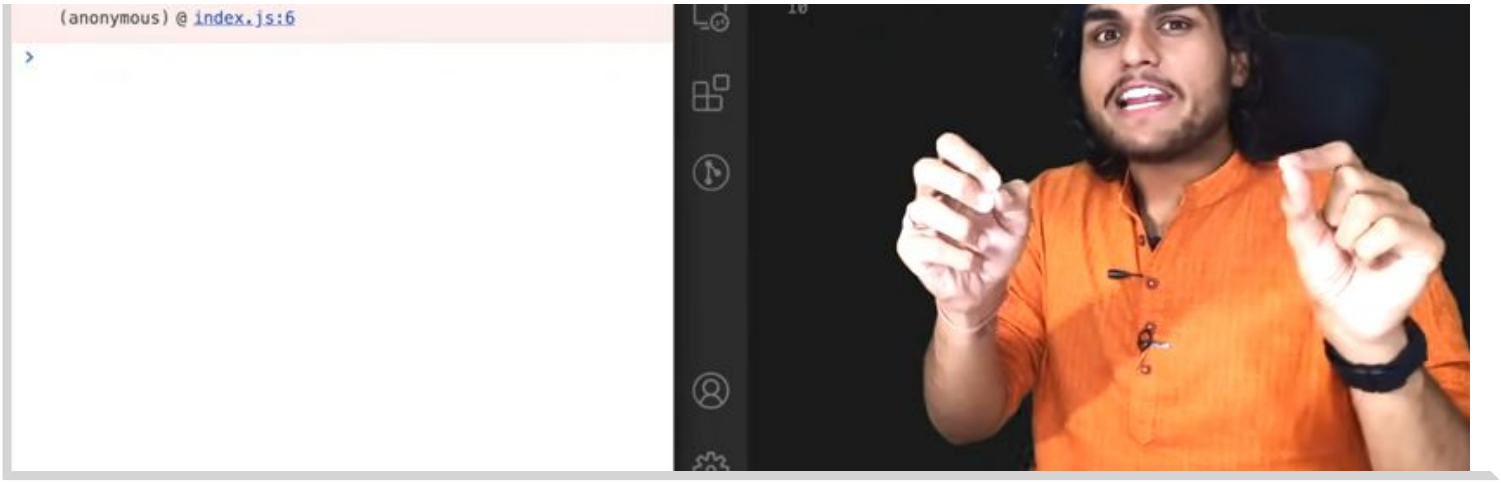
14:15



CONSTexpects to be "initialised" at the same time its being "declared"

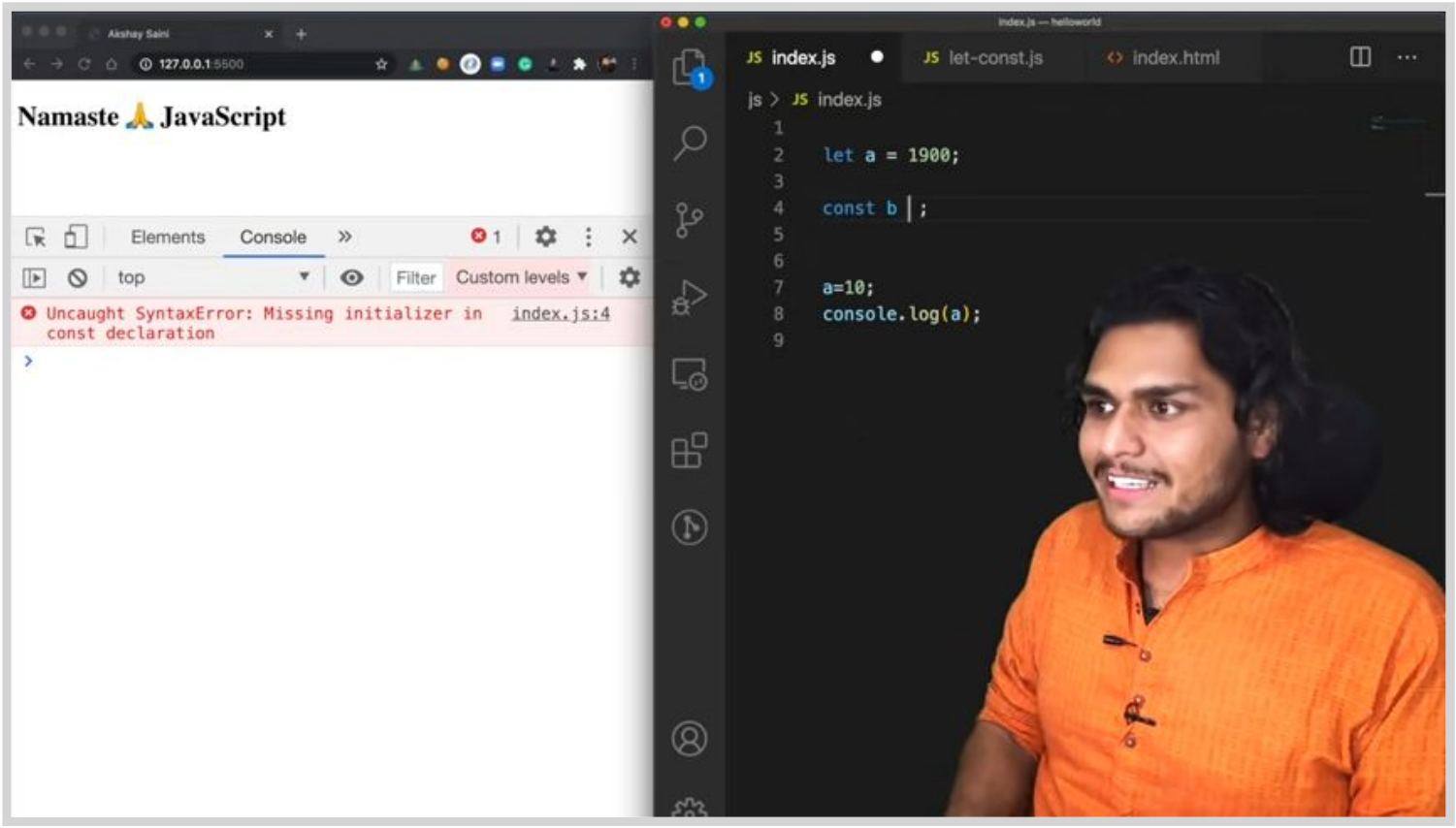
15:12





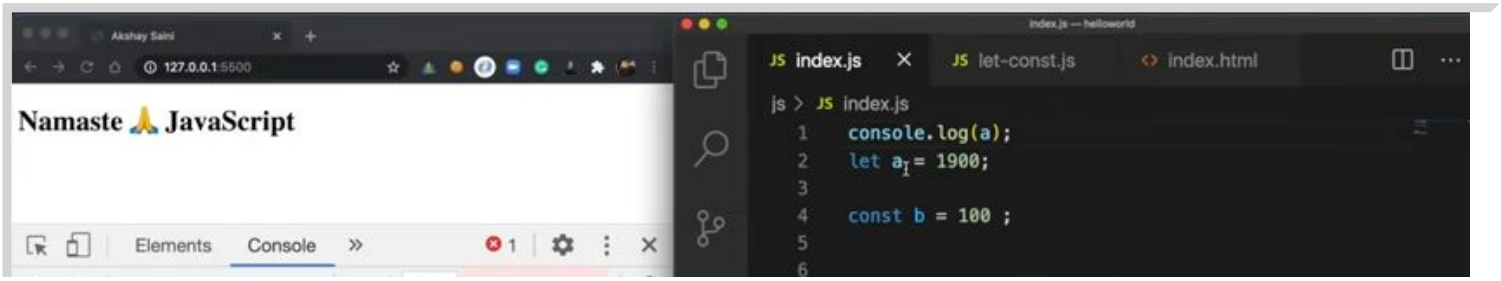
TypeError

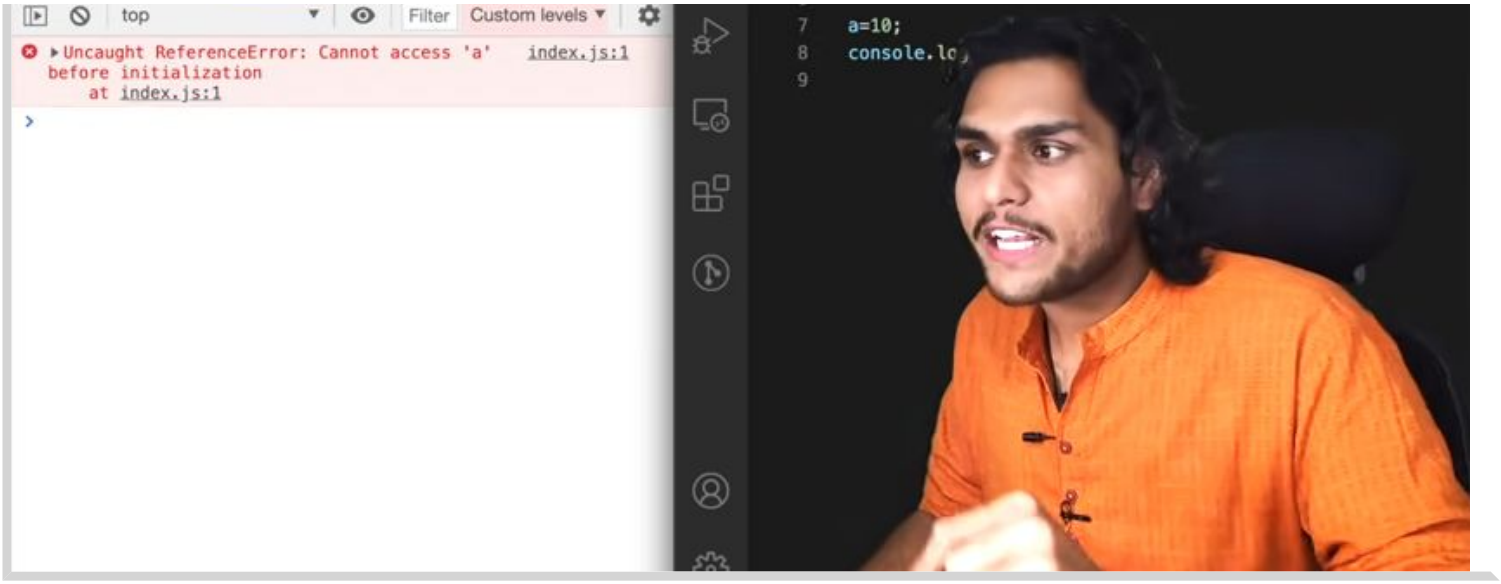
16:20



SyntaxError

17:18





("a" is in the Temporal Dead zone)

thats y it gives ReferenceError.

ReferenceError :- When JStries to find a specific variable inside the memory space & it connot access it.

-> The best way to avoid these "temporal Dead Zone" is to always put your "declaration" & "initialization" at the top