Topics:  
datatypes: 1. Primitive 2. Reference

databases in js

object cloning (deep and shallow)

array and object methods

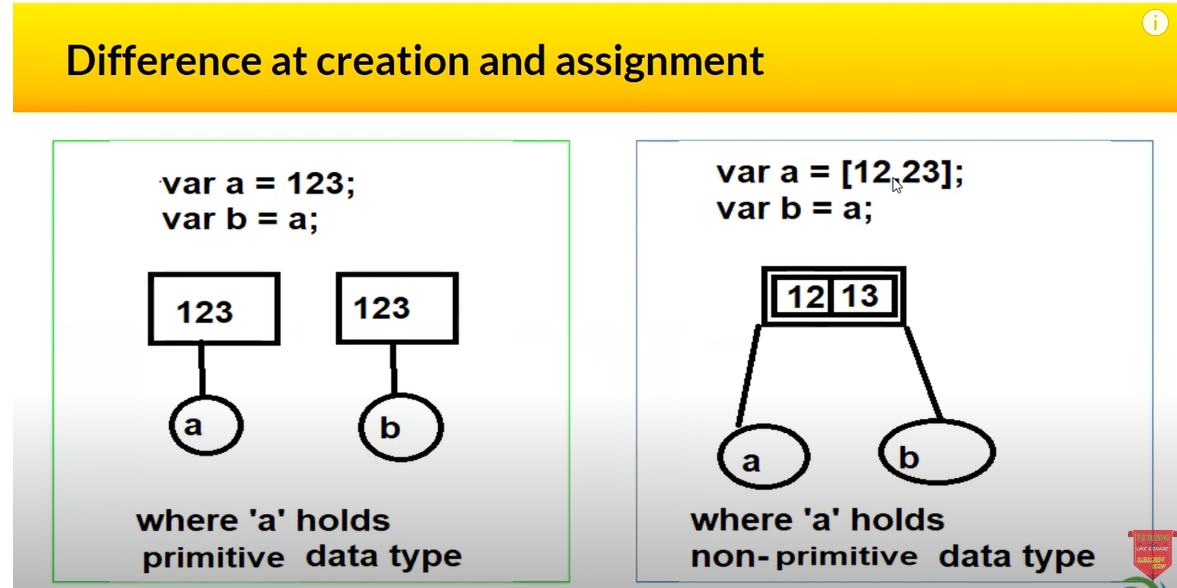
button functionality to get count 1

01-09-2023:

github1s.com

<https://vscode.dev/> 🡪 on line vs code

let and const cannot access before its initialization



If we put debugger at line no 1, which is having var a = 10; it will give undefined which is also known as memory execution phase.

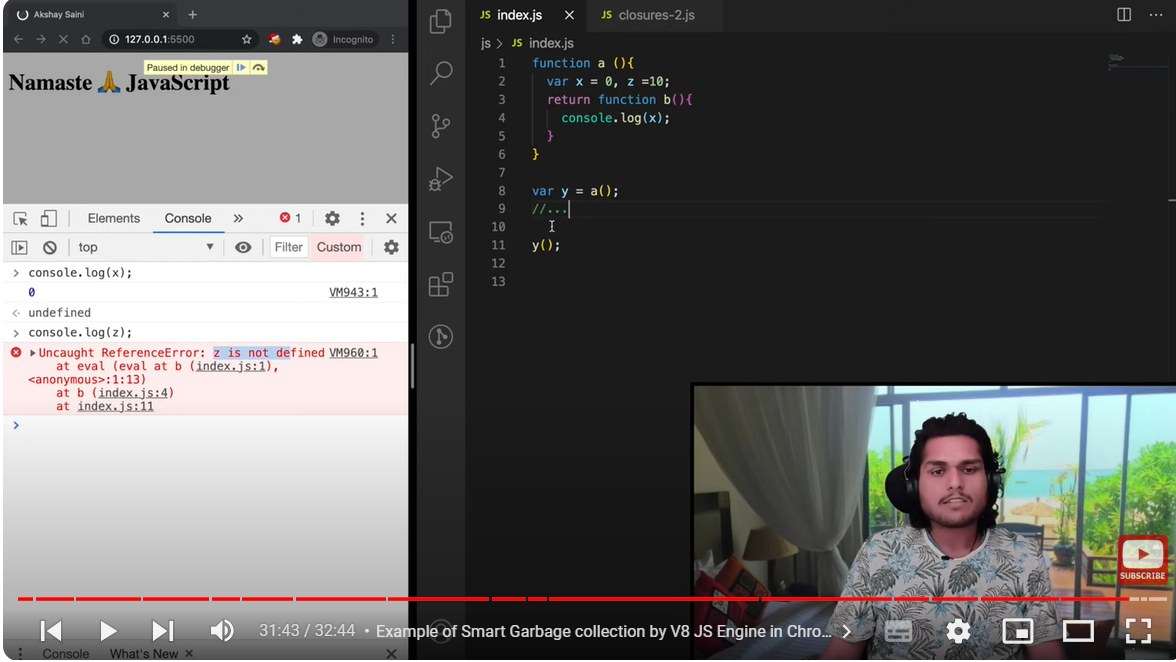
**Shortest js program :** empty js file(even for empty file js engine will create an global execution context)  
gives an window object and this keyword

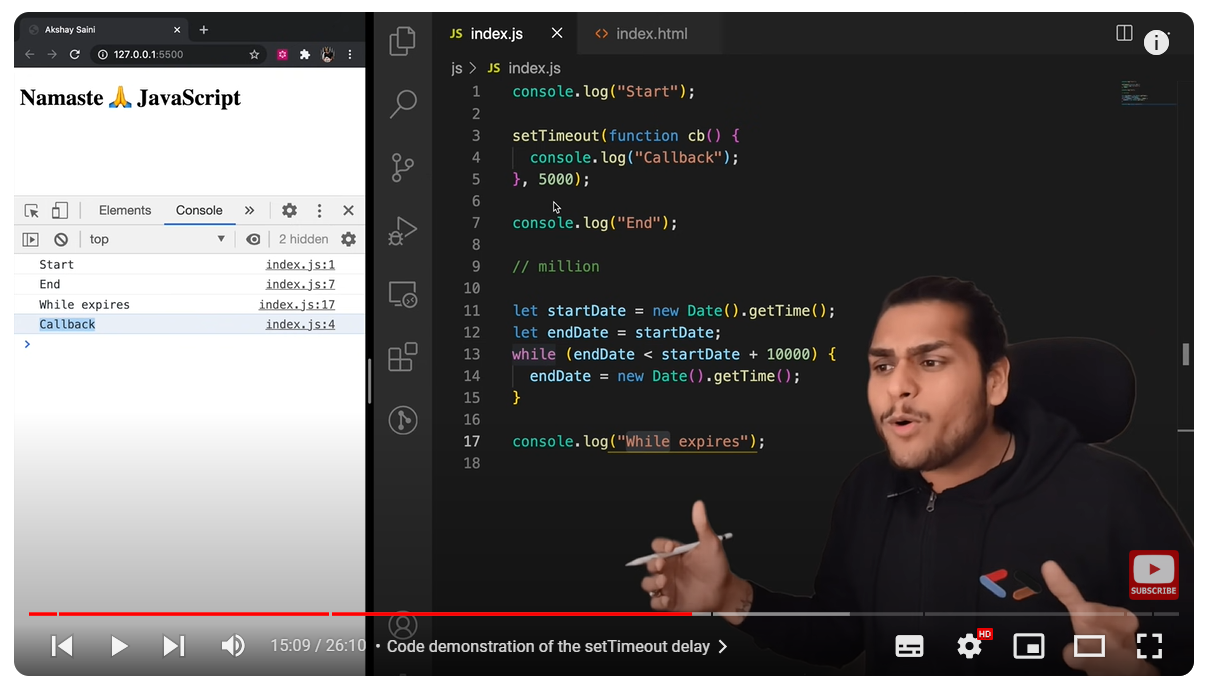
Mutable: *whose state can be modified after it is created eg: let*

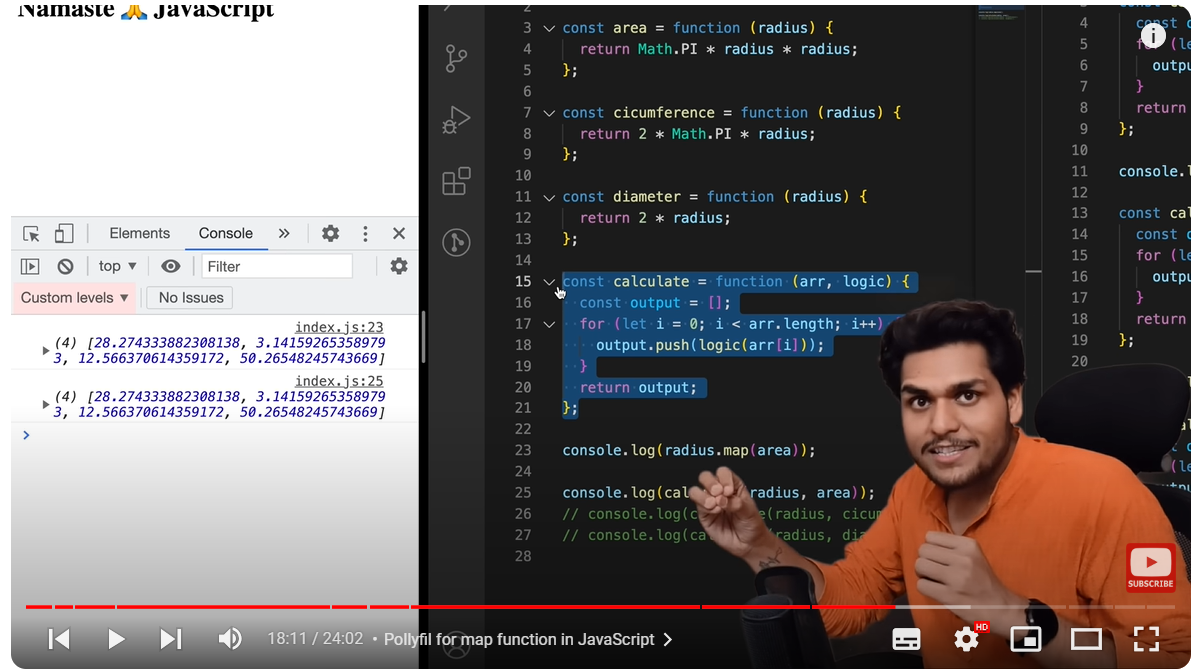
Immutable: *whose state cannot be changed  eg: const*

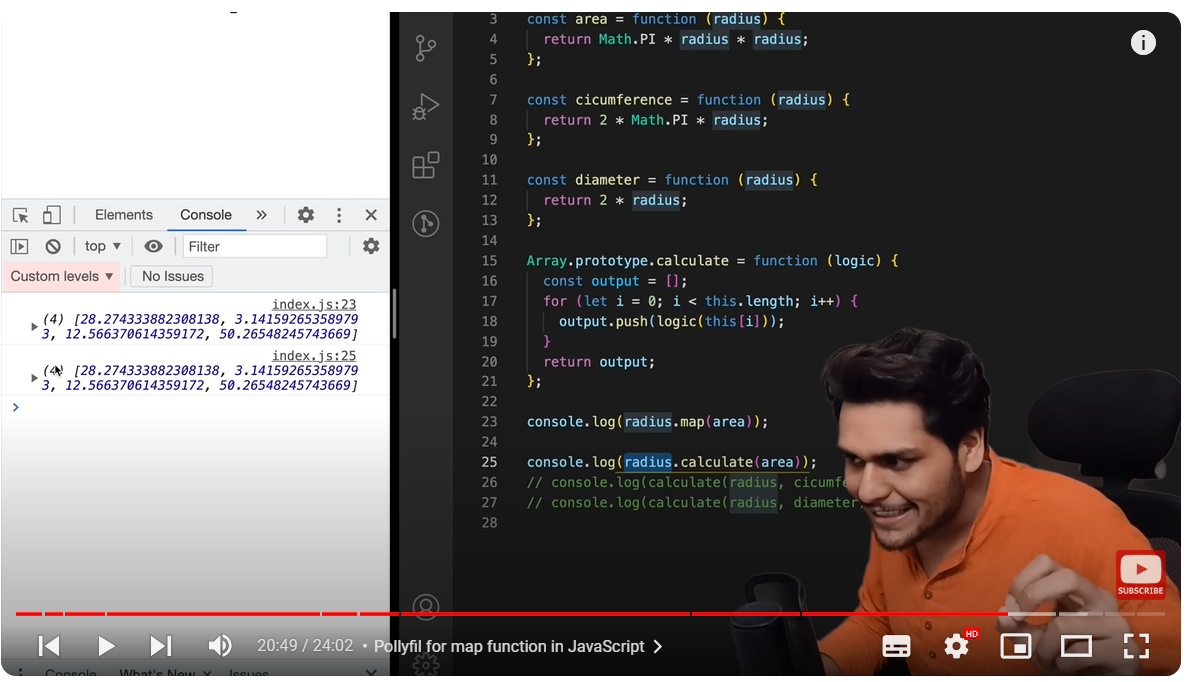
****

**Disadvantages of Closure:**

****

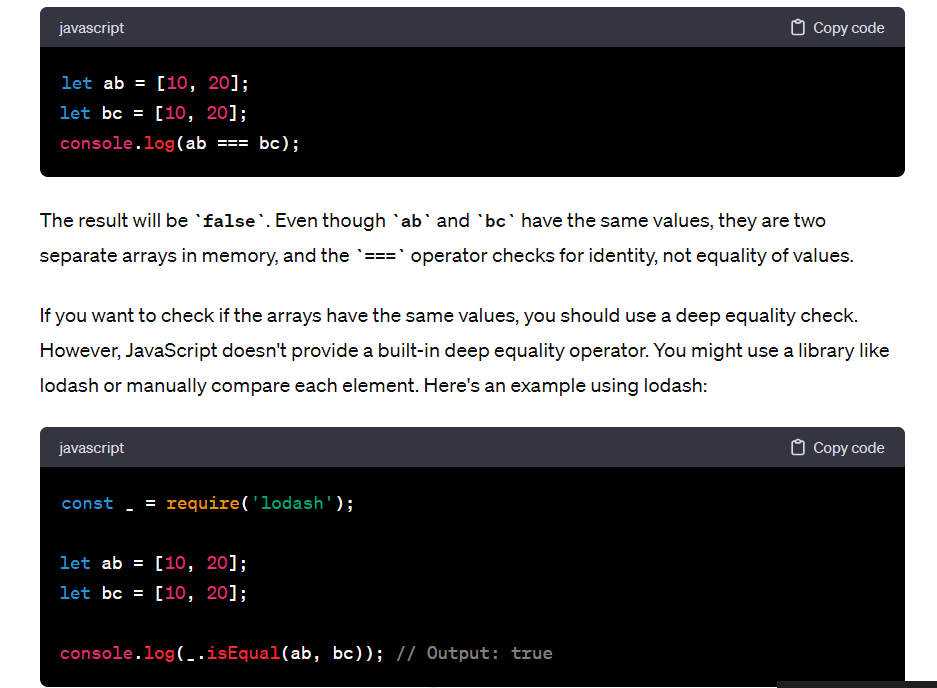
****

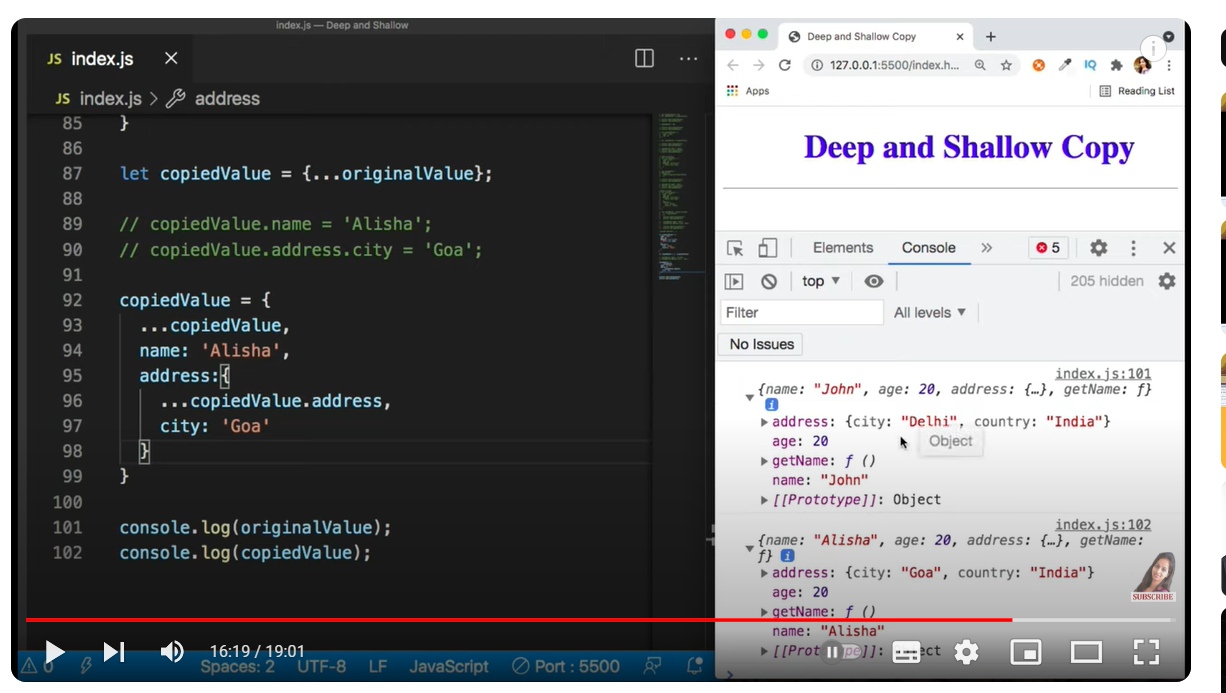
****

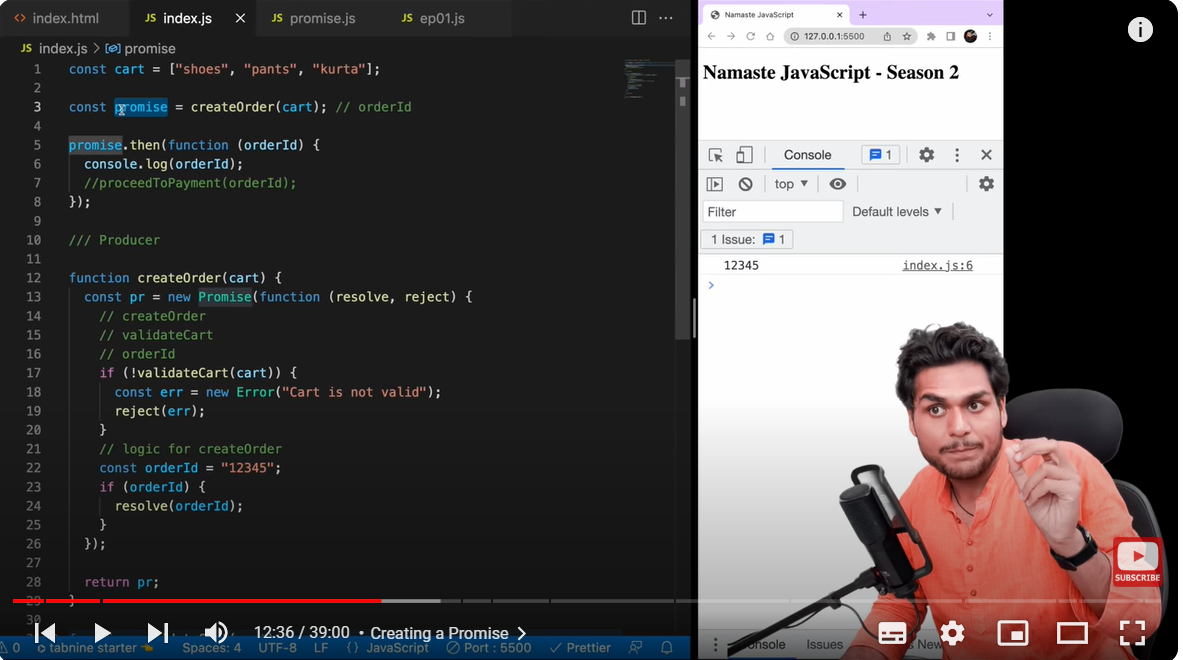
****

You cannot use Typescript in Chrome. Typescript is superset of JS and you must first compile .TS file to .JS file. If you try use TS in browser, you just get an syntax error.

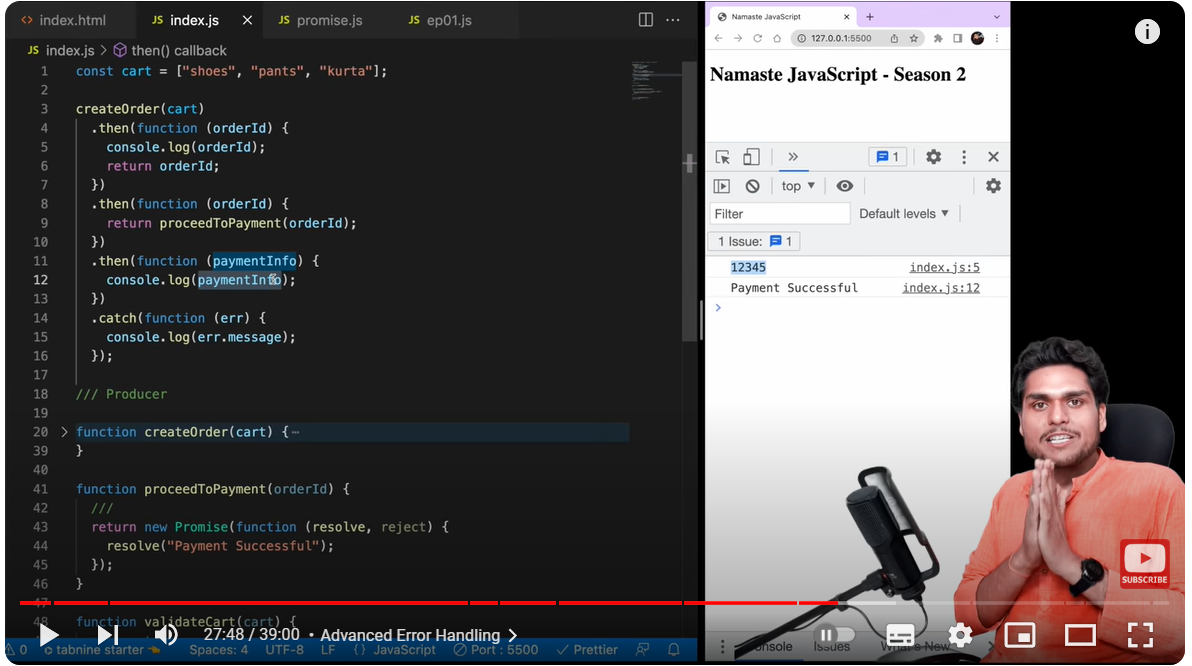
In [JavaScript](https://developer.mozilla.org/en-US/docs/Glossary/JavaScript), [primitive values](https://developer.mozilla.org/en-US/docs/Glossary/Primitive) are immutable — once a primitive value is created, it cannot be changed, although the variable that holds it may be reassigned another value. By contrast, [objects](https://developer.mozilla.org/en-US/docs/Glossary/Object) and [arrays](https://developer.mozilla.org/en-US/docs/Glossary/Array) are mutable by default — their properties and elements can be changed without reassigning a new value.

****

****

****

****

****

****

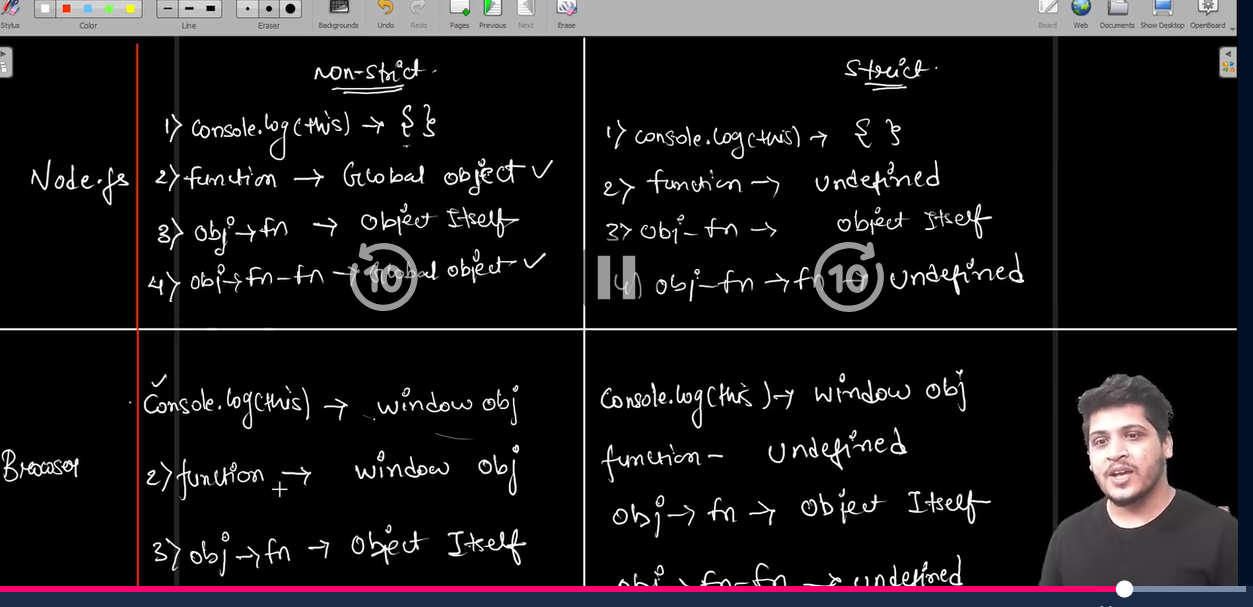
101646164879

When ever we call a promiseFunction, it will create a promise and return a promise, eg as below:

function promiseFunction() {  
 const pro = new Promise((resolve, reject) => {  
  
 });  
 return pro;

}  
  
const promiseResult = promiseFunction();  
console.log(promiseResult)

****

****