**Q1.What is Hoisting?**

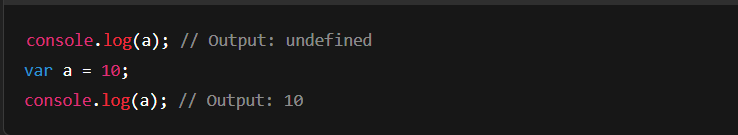
Hoisting in JavaScript is a behavior where variable and function declarations are moved to the top of their scope before code execution. This means you can use functions and variables before actually declaring them in your code.

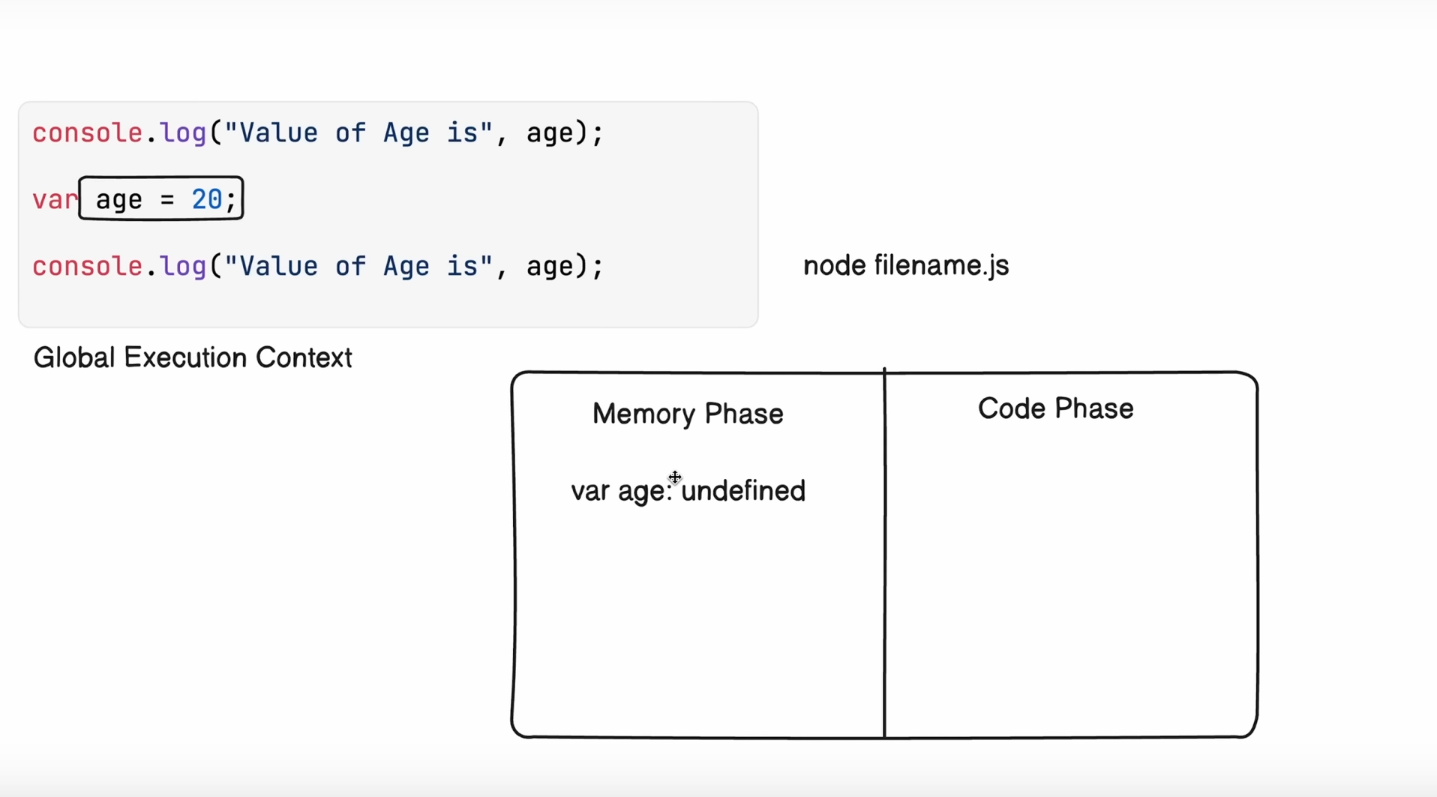
**How Hoisting Works with var, let, and const**

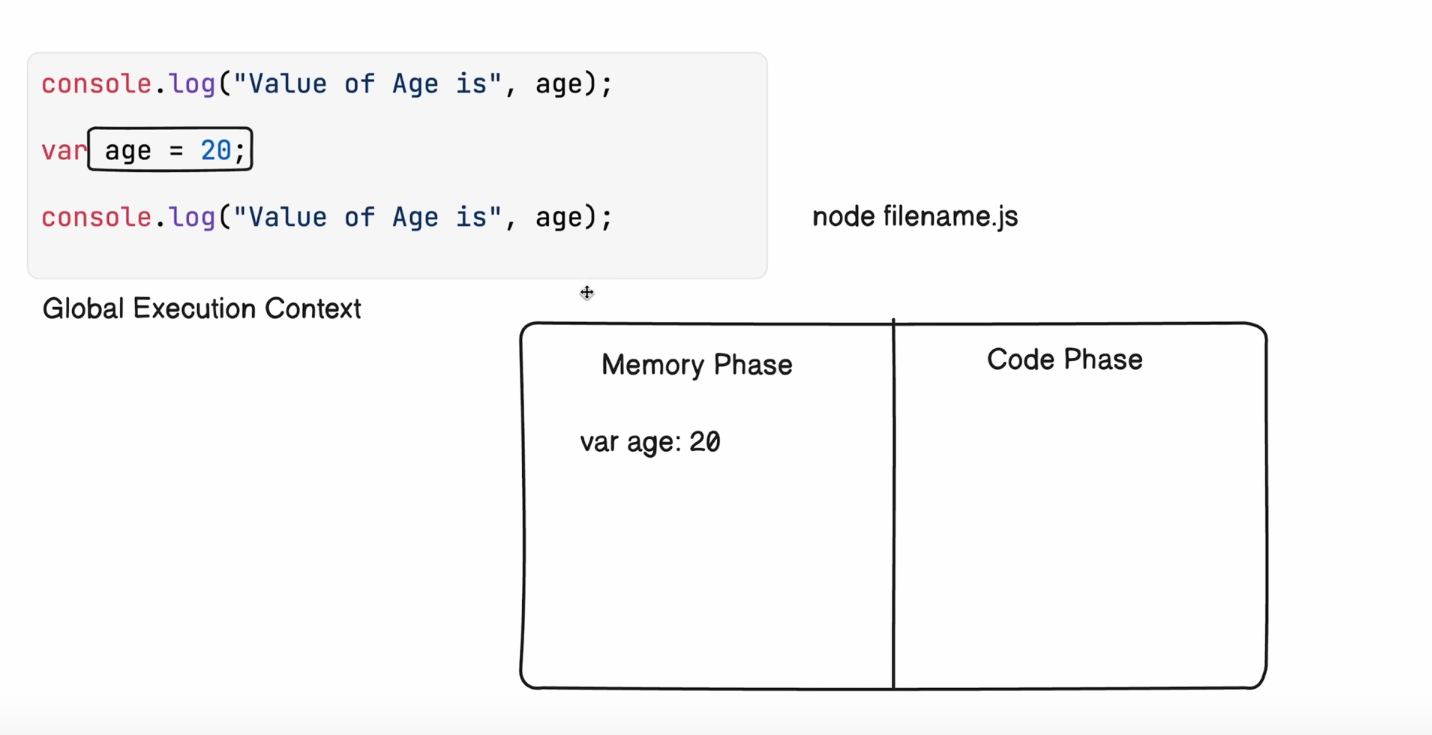
**1. Hoisting with var**

* Variables declared with var are **hoisted**, but their value is initialized as undefined.
* This means you can reference the variable before it's declared, but it will return undefined.

**Example:**

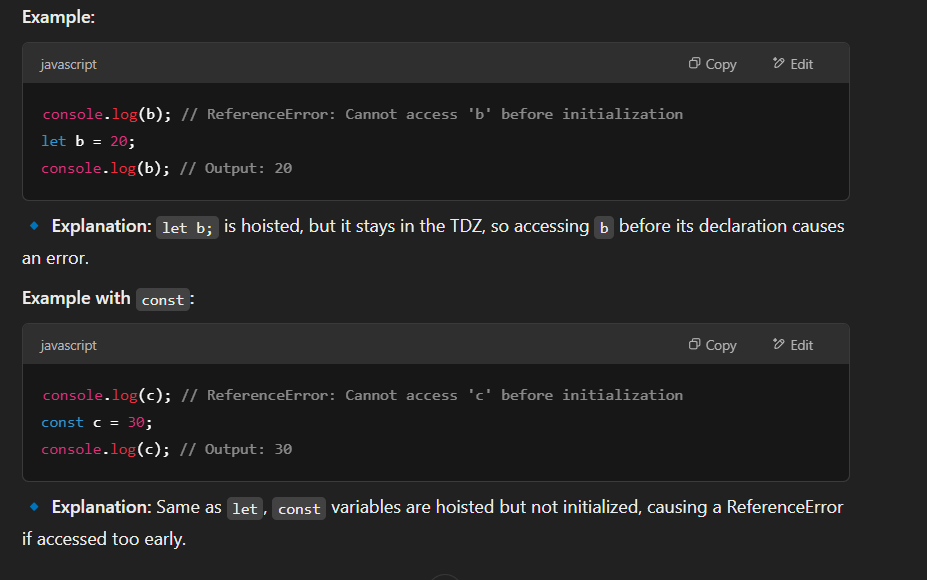




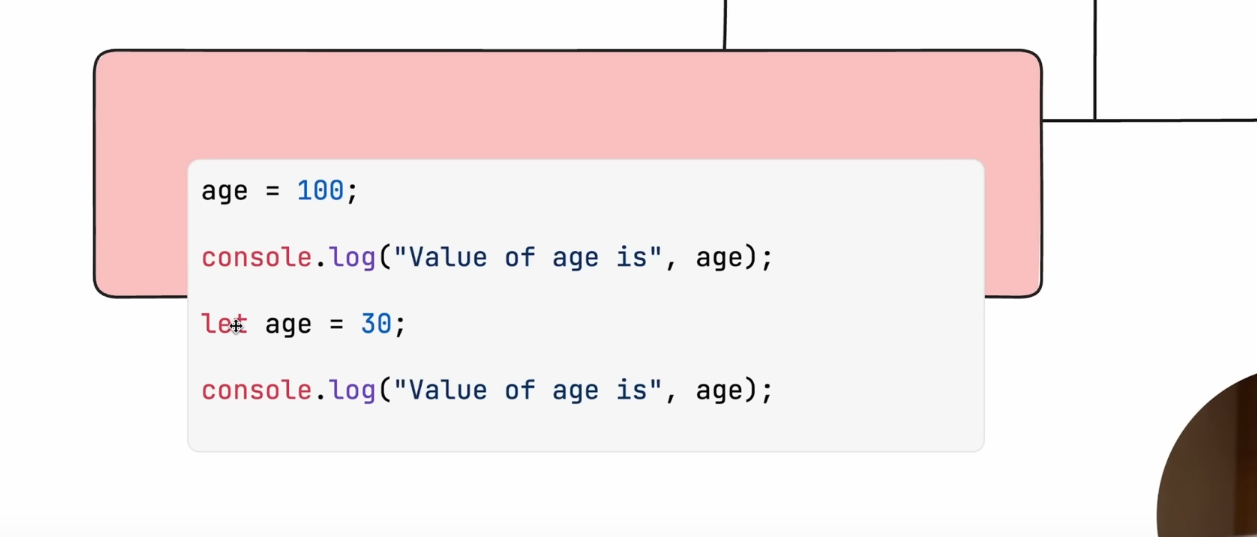


**2. Hoisting with let and const**

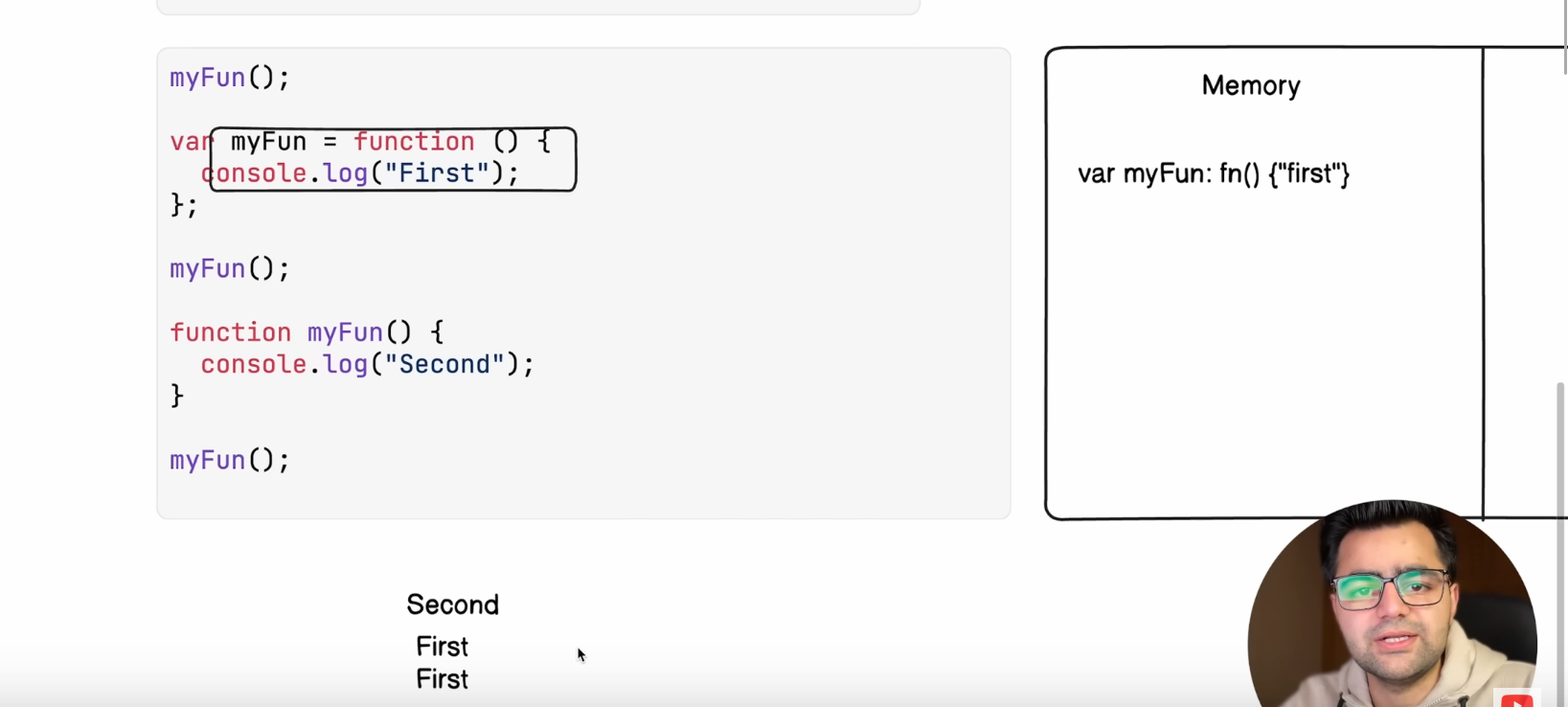
* Variables declared with let and const are also **hoisted**, but they are **not initialized**.
* They remain in the **Temporal Dead Zone (TDZ)** until the line where they are declared is executed.
* Accessing them before declaration results in a **ReferenceError**.



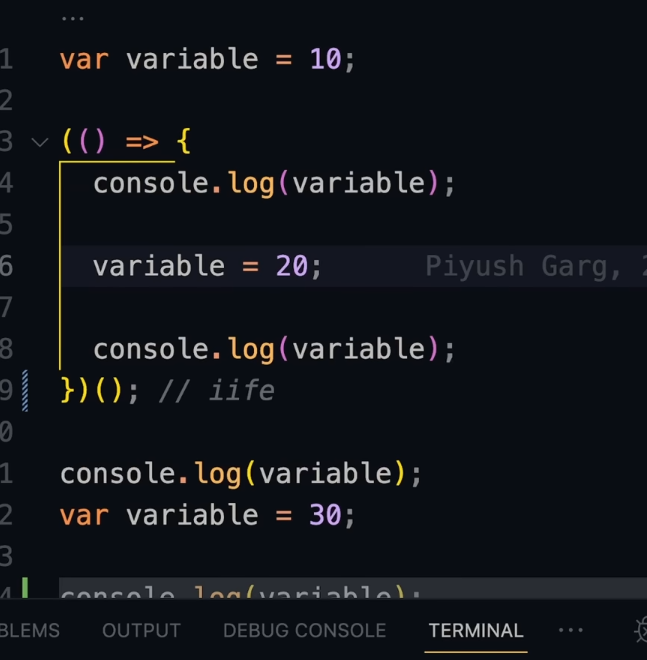
**Temporal Dead Zone**

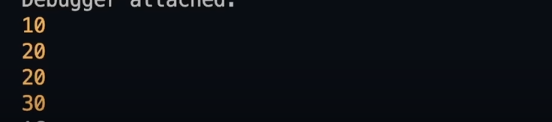


**Q2. Hoisting with Function?**

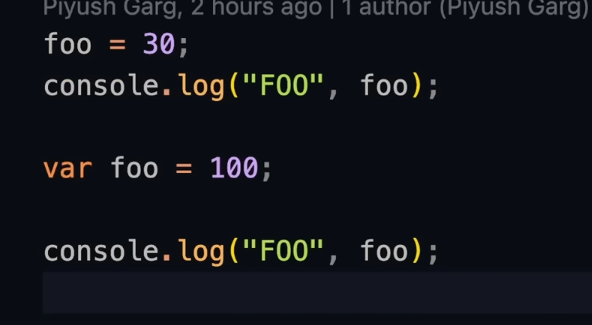
****

**Q3. Hoisting with IIFe?**

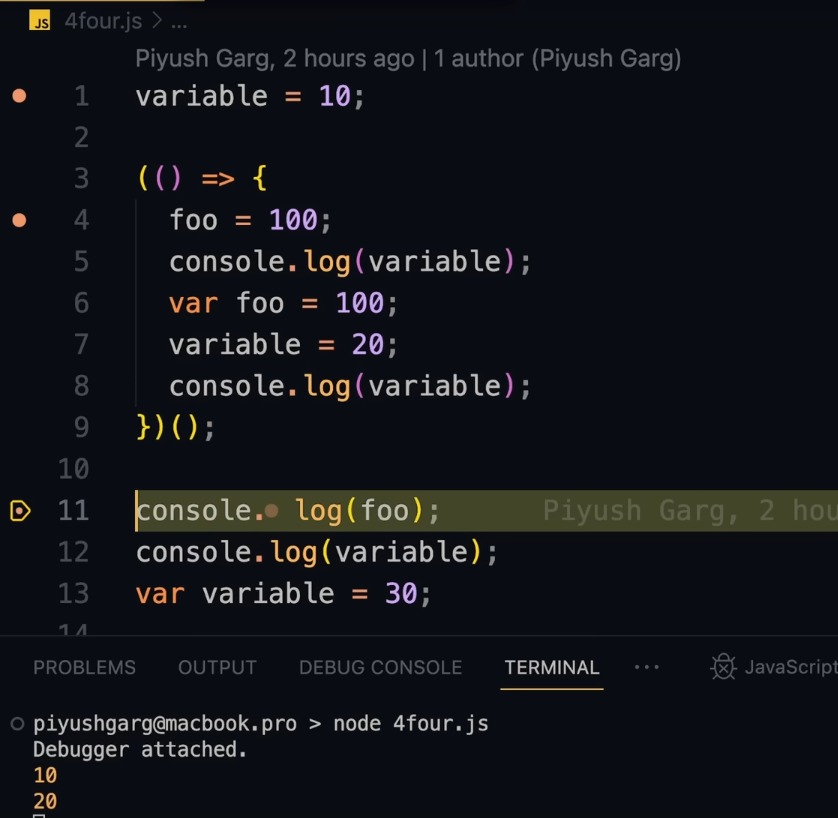
****

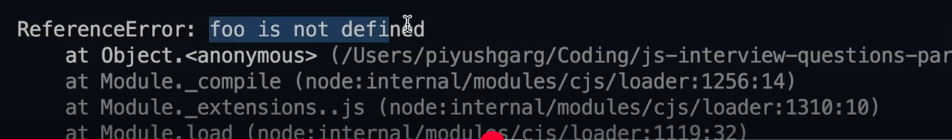
****

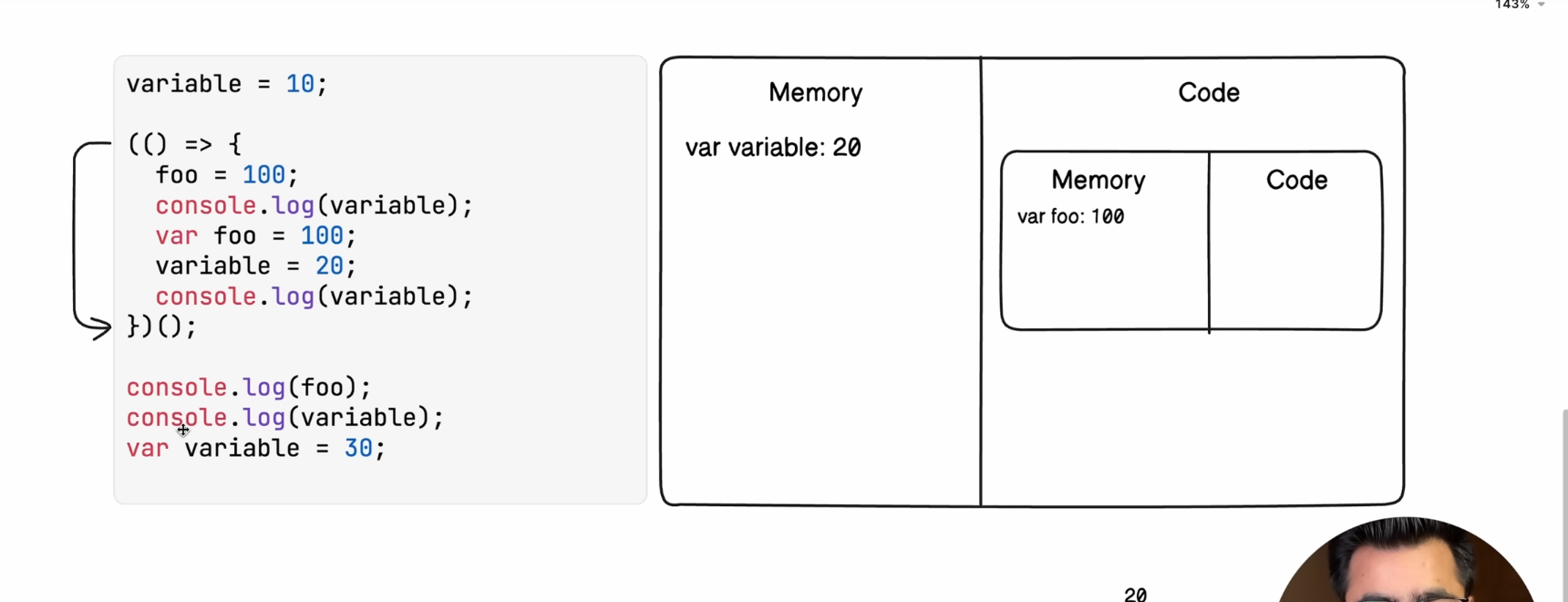
**Q4. Normal Hoisting ?**

****

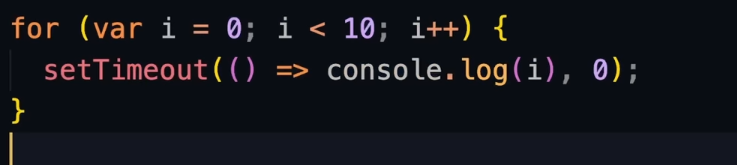
****

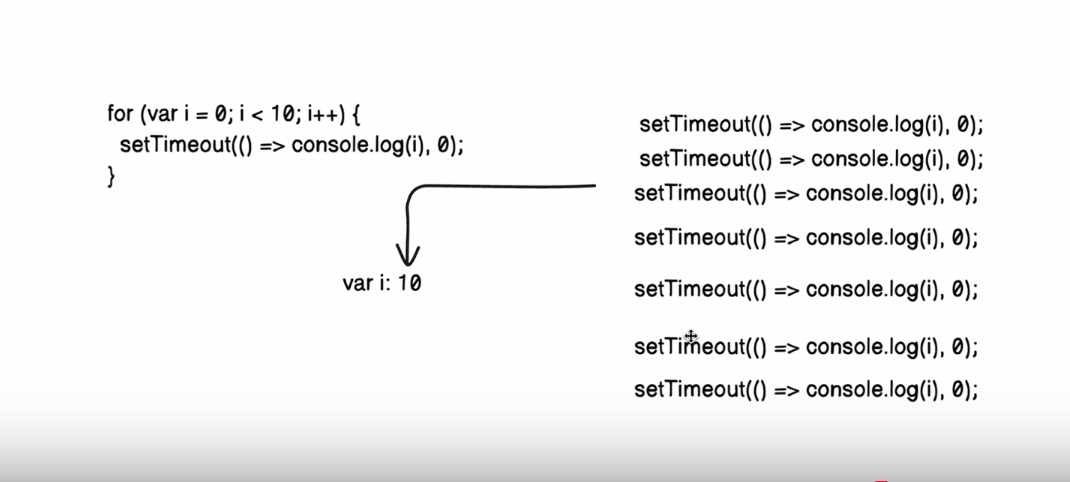
****

****

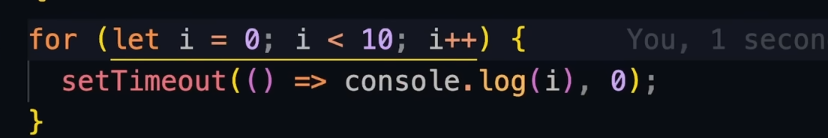
****

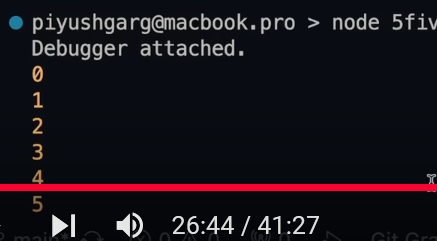
**Q5. Hoisting with Loop?**

****

****

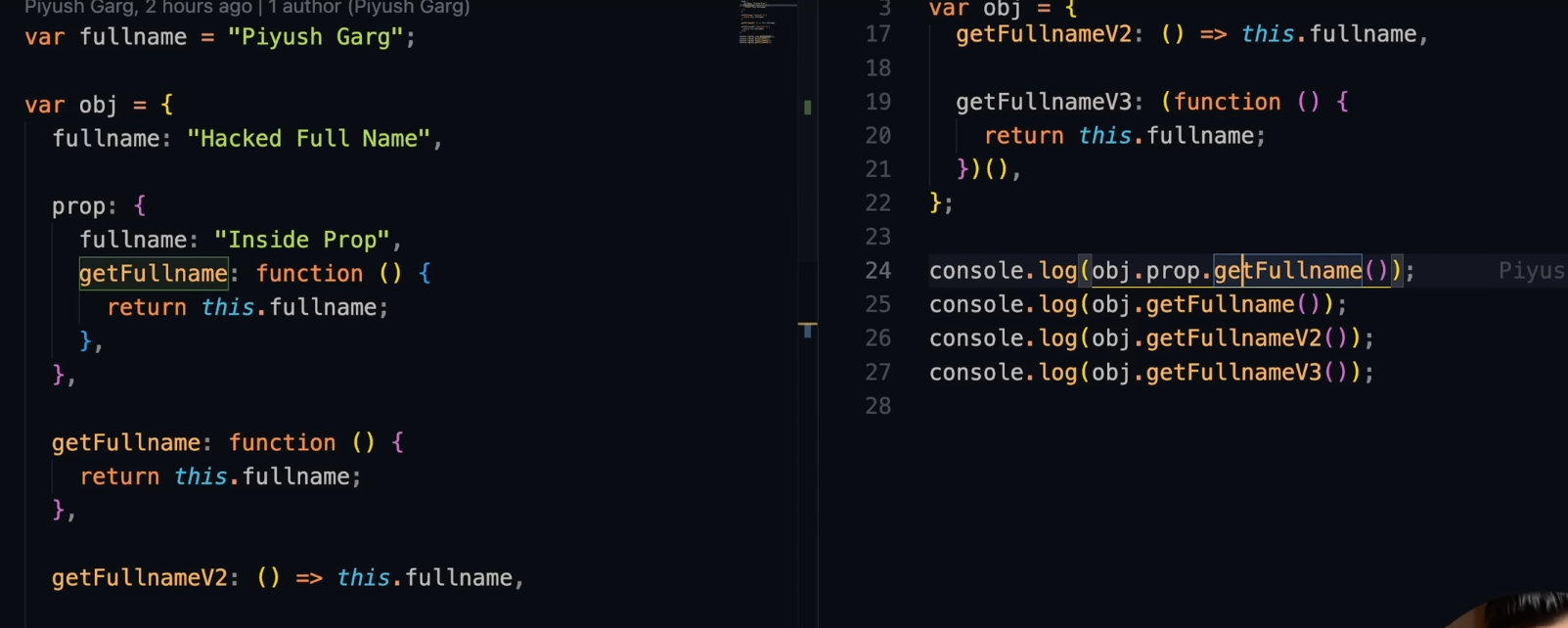
**Var is a global scope that’s why it print I as 10, for each itteration**

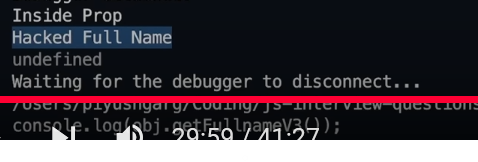
****

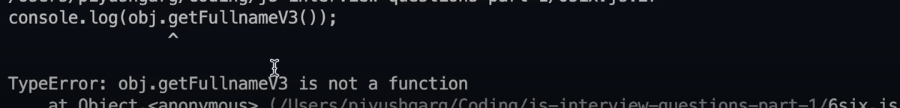
****

**Let is a global scope that’s why it print I as 1 to 9, for each iteration**

** Q6. Hoisting with Object?**

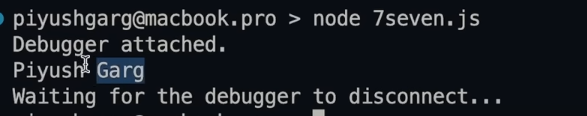
****

****

****

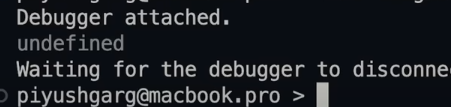
**Q7. Call with Object?**

****

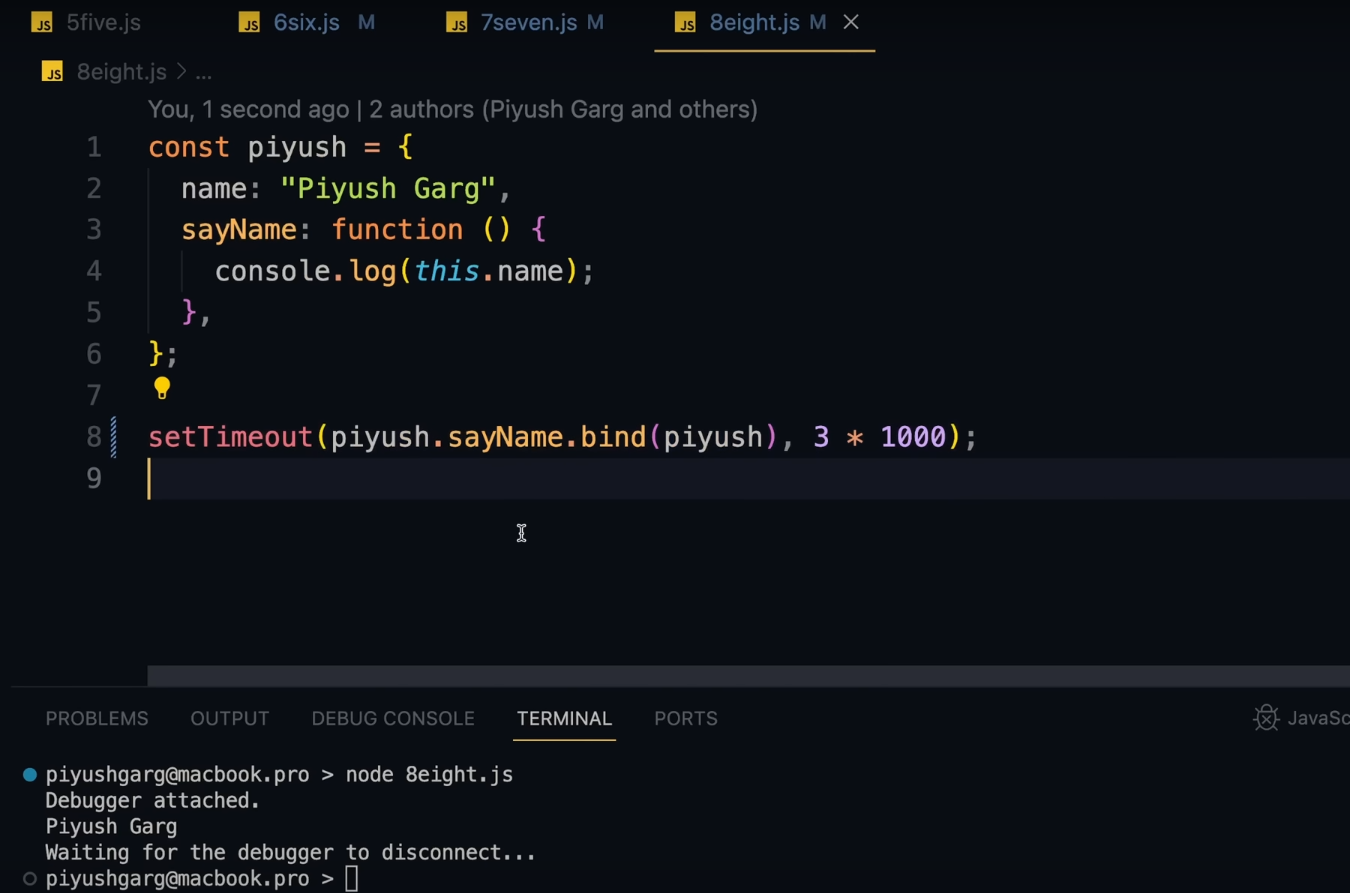
****

**Q8.Acess Name From Object With SetTimeout?**

****

****

**Q8.Acess Name From Object With SetTimeout With bind Function?**

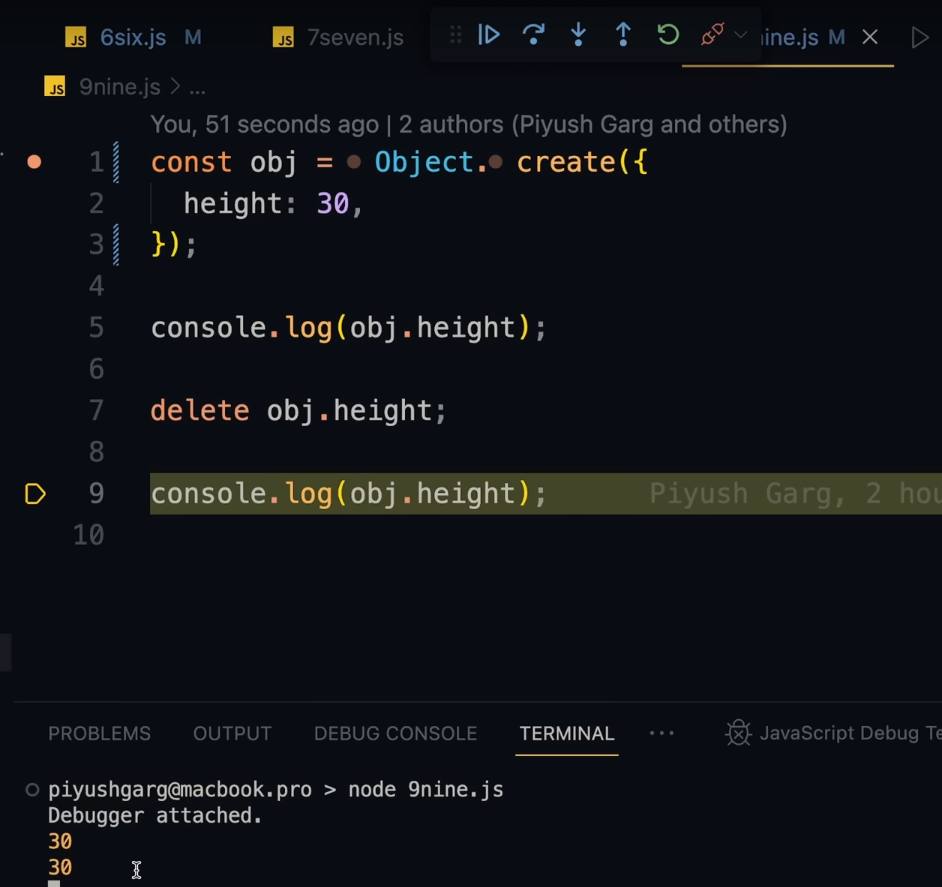
****

Here we bind piyush.sayName with piyush object , and whith the helo of bind it retuen a function .

**Q9.Acess height From Object Without deletion and after deletion?**

****

**Q10.Acess height From Object Without deletion and after deletion with Object.create({})?**

****