**Day 41**





**“Web Development + Security”**

**Advanced JavaScript :**

**What is an IIFE?**

IIFE stands for Immediately Invoked Function Expression. It’s a function in JavaScript that runs automatically as soon as it’s defined — without being called separately.

**Syntax:**(function () {

console.log("I run automatically!");

})();

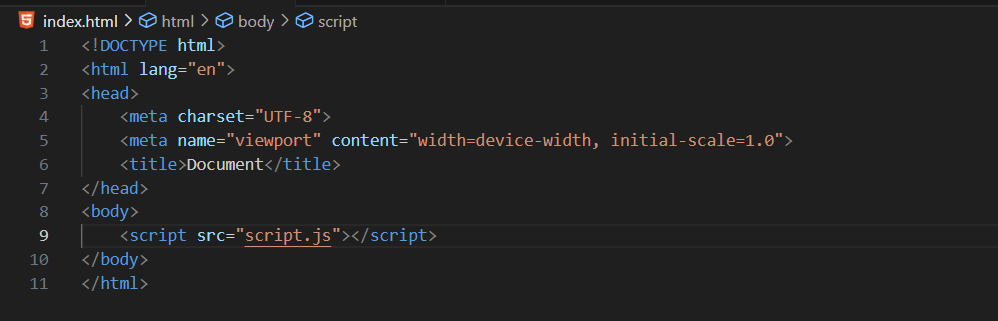
**Why use IIFE?**

IIFEs are used to:

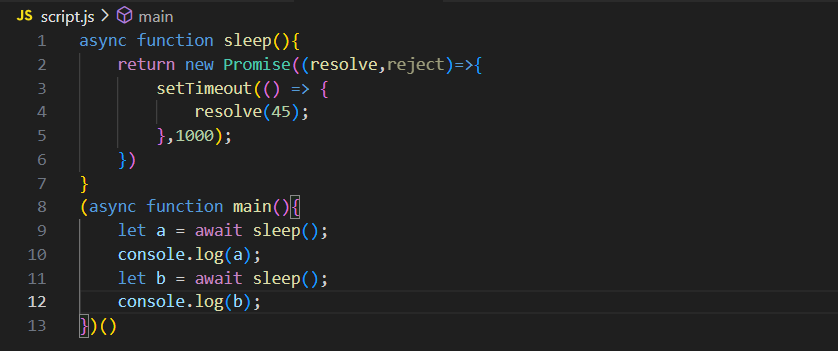
1. Avoid polluting the global scope (variables stay private)
2. Execute code immediately
3. Create a local scope for variables

A basic example of immediately invoked function:

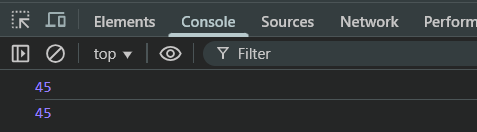
Index.html:



Script.js:



Console:

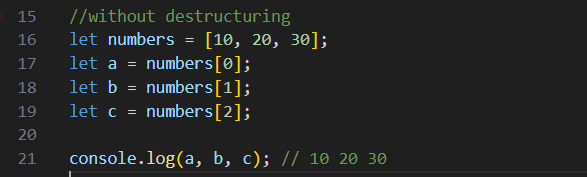


**What is Destructuring?**

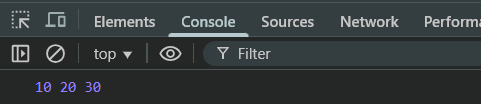
Destructuring means unpacking values from arrays or objects into separate variables —in a clean and short way. Instead of manually accessing each property or index, you can extract them directly.

Let’s understand with the help of array:

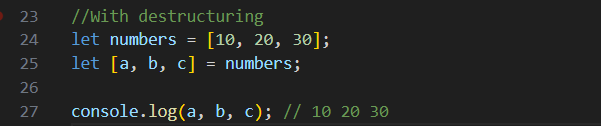
Without destructuing:



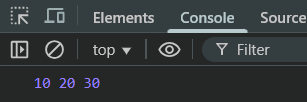
Console:



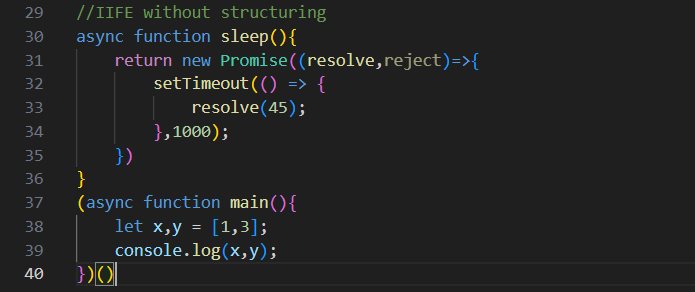
With Destructuring:



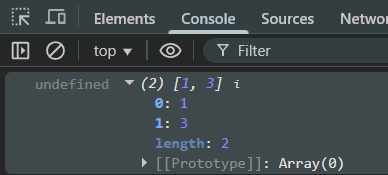
Console:



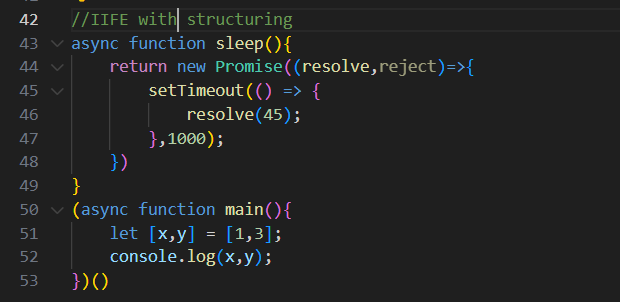
Similarly, IIFE without structuring:



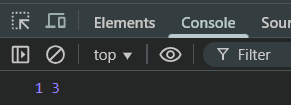
Console:

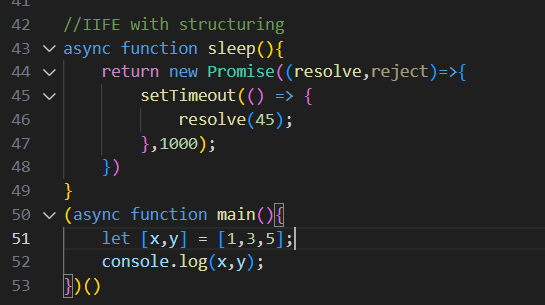


Now, IIFE with structuring:

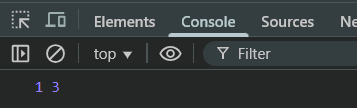


Console:



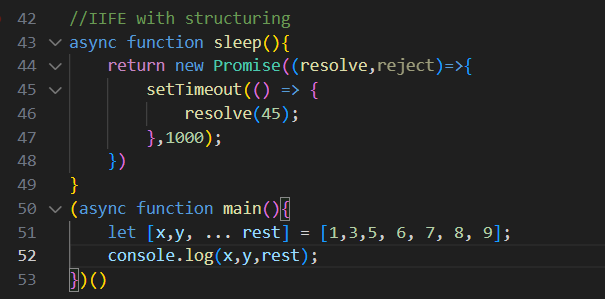
Now, In case we want give third variable value and have not decided in which it will be stored, then it will behave like this:  


Console:

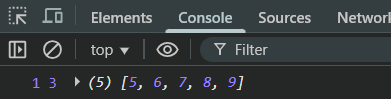


In order to tackle this issue, we need to use “rest” in this as show below:

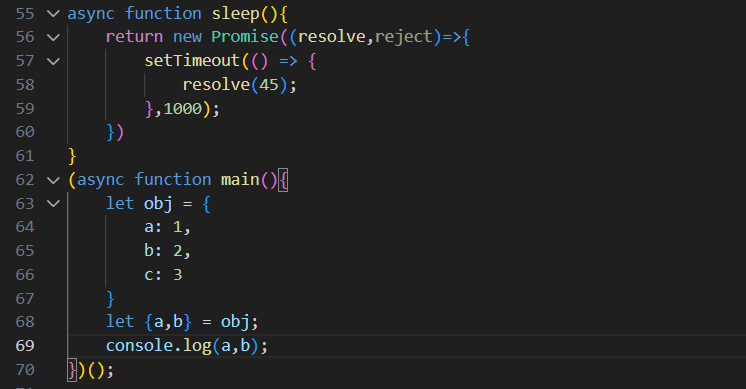
Script.js: here the 5,6,7,8,9 are stored in ‘rest’



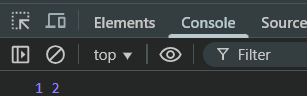
Console:



Also, we can destructure in the object as well:



Console:

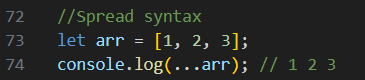


**What is Spread Syntax?**

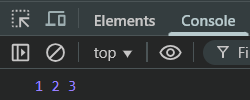
The spread syntax (...) is used to expand (or “spread out”) the contents of an array, object, or string into individual elements. It helps copy, combine, and pass values easily.

Example: a very basic example

Script.js

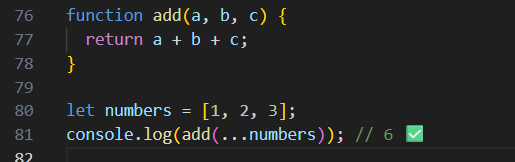


Console:

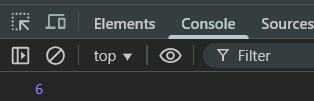


Example: adding the elements of array using the spread syntax

Script.js:



Console:



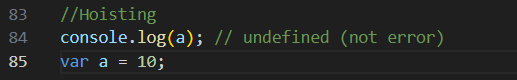
**What is Hoisting?**

Hoisting means JavaScript moves declarations (not initializations) to the top of their scope — *before the code executes*.

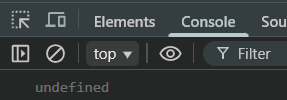
In simple words: You can use variables or functions before declaring them, because the declaration is “hoisted” (lifted) by JavaScript internally.

Example: var hoisting

Script.js:

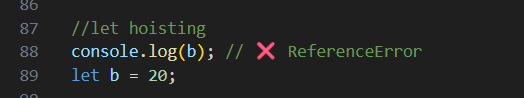


Console:

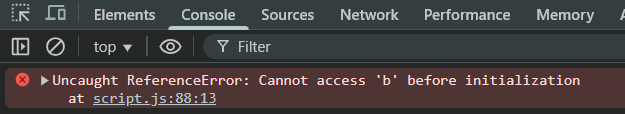


Example: let and const are not fully hoisted

Script.js:

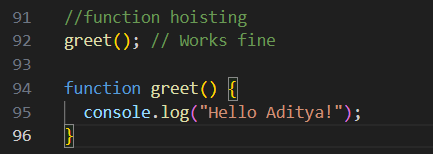


Console:

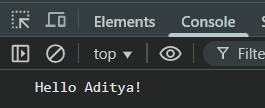


Example: Function declarations are fully hoisted — you can call them before defining them.

Script.js:



Console:



--The End--