

ASSIGNMENT - 6

Q.1 - Find the output if we change the value of var in the module, imported in the main file.

Ans - In Node.js, if you export a primitive value and change it in the main file, the original module does not get affected because primitive values are passed by value. However, if you export an object and change one of its properties, the change is reflected everywhere because objects are passed by reference.

```
<!-- HW-1: Changing var in imported module -->
<!DOCTYPE html>
<html lang="en">
<head><meta charset="UTF-8"><title>HW-1</title></head>
<body>
<h2>HW-1: Changing Module Variable</h2>

|
<script>
let moduleVar = { value: 5 };

function changeModuleVar() {
  moduleVar.value = 10;
}

console.log("Before:", moduleVar.value); // 5
changeModuleVar();
console.log("After:", moduleVar.value); // 10
</script>
</body>
</html>
```

Q.2 - Find the difference b/w Asynchronous and synchronous, strict and non -strict in mjs.

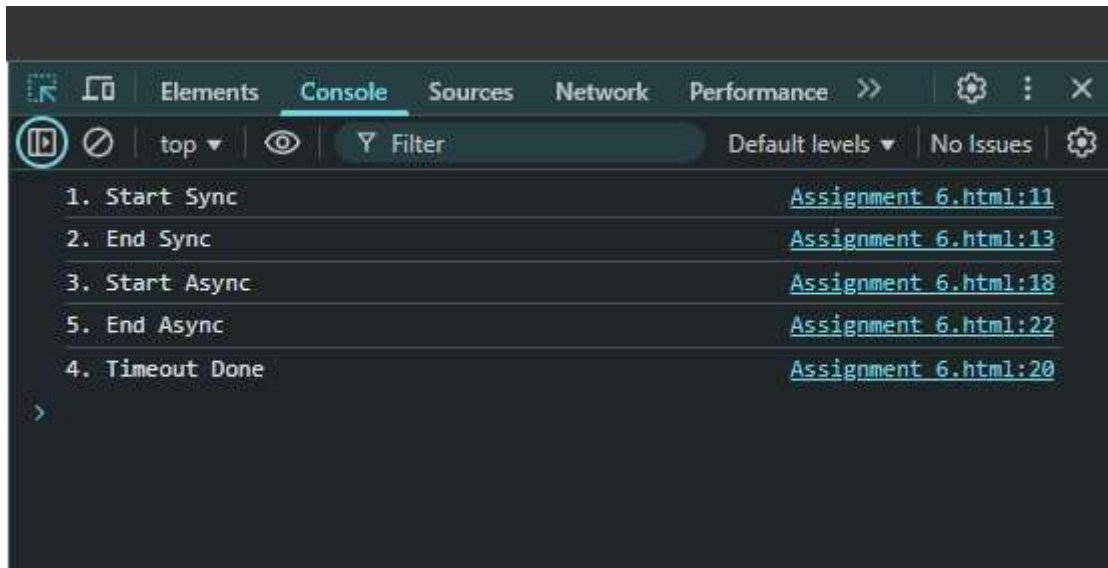
Ans - In Node.js, synchronous code runs one line at a time and stops the program from moving forward until the current task is finished. On the other hand, asynchronous code lets certain tasks, like reading files or calling APIs, run in the background so the rest of the program can keep going. Strict mode (written as 'use strict';) is a feature in JavaScript that makes the rules stricter to help catch errors and write safer code. Non-strict mode is more relaxed and can allow risky behavior, like using variables that haven't been declared.

SYNCHRONOUS & ASYNCHRONOUS

```
<!DOCTYPE html>
<html>
<head>
  <title>Synchronous vs Asynchronous</title>
</head>
<body>
  <h2>Check Console for Output</h2>
  <script>
    // Synchronous
    function syncFunc() {
      console.log("1. Start Sync");
      for (let i = 0; i < 1e8; i++) {}
      console.log("2. End Sync");
    }

    // Asynchronous
    function asyncFunc() {
      console.log("3. Start Async");
      setTimeout(() => {
        console.log("4. Timeout Done");
      }, 1000);
      console.log("5. End Async");
    }

    syncFunc();
    asyncFunc();
  </script>
</body>
</html>
```

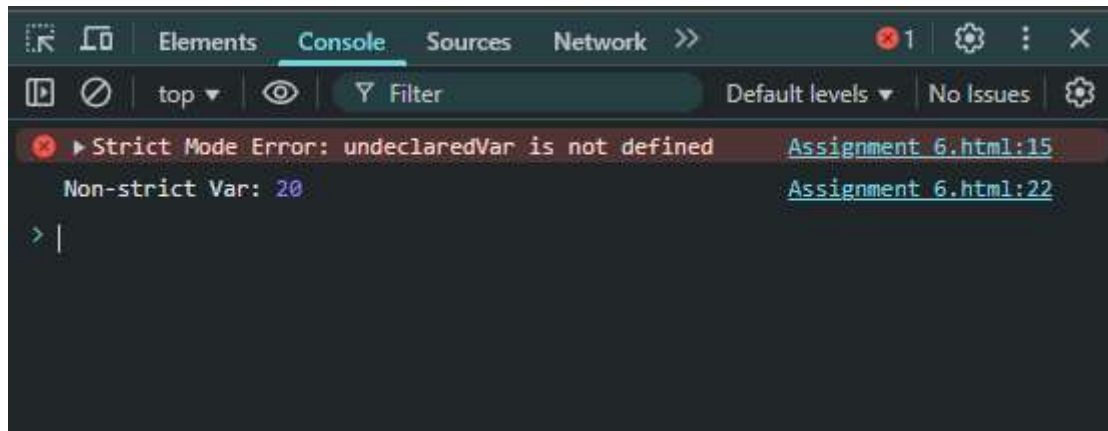


SYNC & ASYNC

```
<!DOCTYPE html>
<html>
<head>
  <title>Strict vs Non-strict</title>
</head>
<body>
  <h2>Open Console to See Error in Strict Mode</h2>
  <script>
    // Strict Mode
    function strictExample() {
      'use strict';
      try {
        undeclaredVar = 10;
      } catch (e) {
        console.error("Strict Mode Error:", e.message);
      }
    }

    // Non-strict Mode
    function nonStrictExample() {
      undeclaredVar2 = 20;
      console.log("Non-strict Var:", undeclaredVar2);
    }

    strictExample();
    nonStrictExample();
  </script>
</body>
</html>
```



Q.3 - Find the difference b/w writing myName and myName:myName in module export.

Ans - In Node.js, when using module.exports, writing myName is a shorthand for myName: myName, meaning the property name and the variable name are the same. It simplifies the code when both names match. However, myName: myName is the longer, explicit form that does the same thing. This longer form is useful when you want the exported property name to be different from the variable name, like myName: userName. Both forms create an object, but the shorthand is more concise when the names are identical.

Code & Output

```

<!DOCTYPE html>
<html>
<head>
  <title>Export Style Difference</title>
</head>
<body>
  <h2>Check Console for Output</h2>

  <script>
    // Variable
    const myName = "Vansh";

    // Shorthand export (key and value are the same)
    const export1 = { myName };

    // Full form export (explicit key and value)
    const export2 = { myName: myName };

    // Custom key export
    const export3 = { userName: myName };

    console.log("export1 (shorthand):", export1);    // { myName: 'Vansh' }
    console.log("export2 (long form):", export2);    // { myName: 'Vansh' }
    console.log("export3 (custom key):", export3);   // { userName: 'Vansh' }

    // Accessing values
    console.log("Access export1.myName:", export1.myName); // Vansh
    console.log("Access export3.userName:", export3.userName); // Vansh
  </script>
</body>
</html>

```



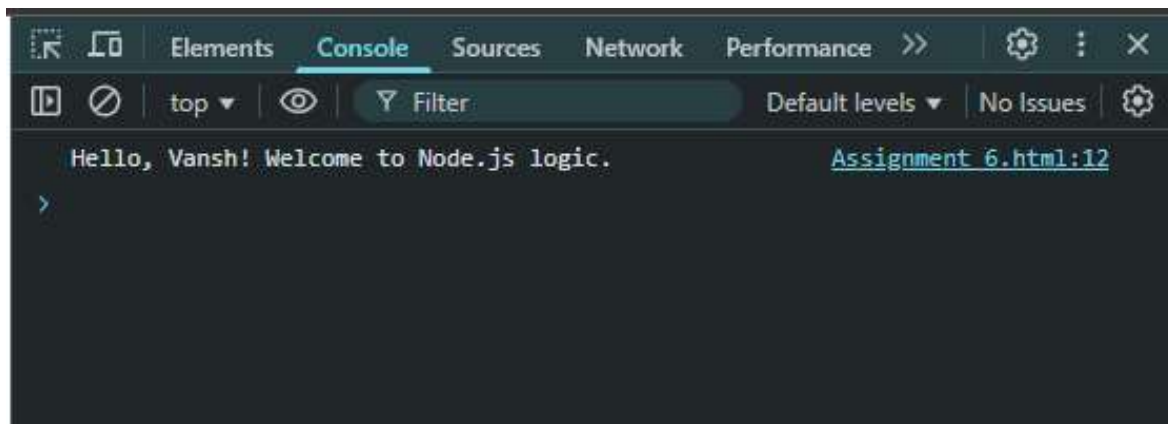
Q.4 - Do some logic on [index.js](#)

Ans -

CODE -1

```
<!DOCTYPE html>
<html>
<head>
  <title>Using index.js Logic</title>
</head>
<body>
  <h2>Check the console for output</h2>

  <script src="index.js"></script>
  <script>
    const message = greet("Vansh");
    console.log(message);
  </script>
</body>
</html>
```



CODE -2

```
<!DOCTYPE html>
<html>
<head>
  <title>Greet User</title>
</head>
<body>
  <h2 id="output">Greeting will appear here</h2>

  <script src="index.js"></script>
  <script>
    const message = greet("Vansh");
    document.getElementById("output").innerText = message;
  </script>
</body>
</html>
```

