

Day 4 – [24th June 2025]

TOPICS COVERED

JavaScript Coercion: Abstract vs Strict Equality

Today, I explored how JavaScript handles equality comparison differently when using `==` (abstract) and `===` (strict).

- `==` performs type coercion before comparing values:
 `"5" == 5` // true – string converted to number
 `null == undefined` // true
 `0 == false` // true
- `===` checks both value and type without conversion:
 `"5" === 5` // false – string vs number
 `0 === false` // false – number vs boolean
 `5 === 5` // true – same type and value

Hoisting:

I also learned about hoisting in JavaScript, where declarations are conceptually moved to the top of their scope before code runs.

`var` is hoisted and initialized with `undefined`:

```
console.log(x); // undefined
```

```
var x = 10;
```

`let` and `const` are hoisted too but not initialized, which leads to a Temporal Dead Zone error if accessed before declaration.

Function declarations are fully hoisted and can be called before their definition.

TOOLS USED:

Visual Studio Code (VS Code)

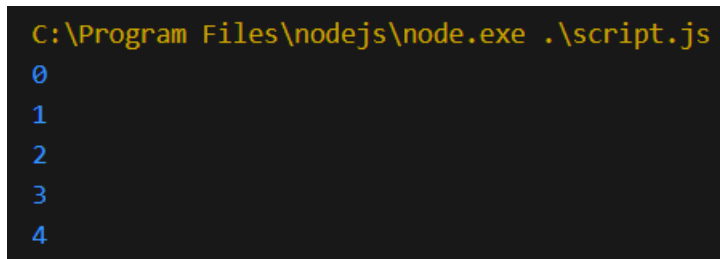
Chrome Browser (JavaScript Console)

TASK:

Practice writing different types of loops:

1. for Loop

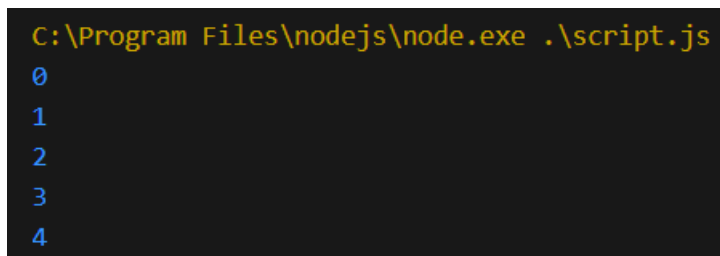
```
for (let i = 0; i < 5; i++) {  
  console.log(i);  
}
```

A terminal window showing the command 'C:\Program Files\nodejs\node.exe .\script.js' and its output, which is the numbers 0, 1, 2, 3, and 4 on separate lines.

```
C:\Program Files\nodejs\node.exe .\script.js  
0  
1  
2  
3  
4
```

2. while Loop

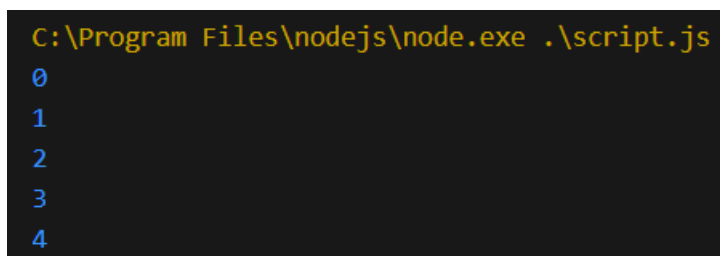
```
let i = 0;  
while (i < 5) {  
  console.log(i);  
  i++;  
}
```

A terminal window showing the command 'C:\Program Files\nodejs\node.exe .\script.js' and its output, which is the numbers 0, 1, 2, 3, and 4 on separate lines.

```
C:\Program Files\nodejs\node.exe .\script.js  
0  
1  
2  
3  
4
```

3. do-while Loop

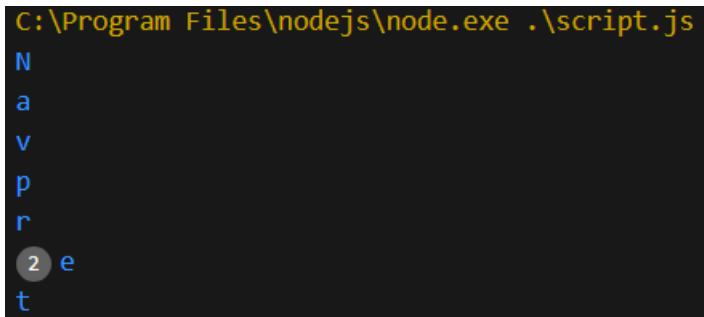
```
let i = 0;  
do {  
  console.log(i);  
  i++;  
} while (i < 5);
```

A terminal window showing the command 'C:\Program Files\nodejs\node.exe .\script.js' and its output, which is the numbers 0, 1, 2, 3, and 4 on separate lines.

```
C:\Program Files\nodejs\node.exe .\script.js  
0  
1  
2  
3  
4
```

4. for-of Loop (for arrays, strings)

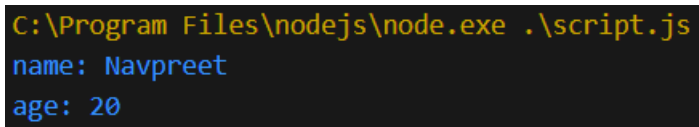
```
for (let i of "Navpreet") {  
  console.log(i);  
}
```



A terminal window with a black background and yellow text. The command `C:\Program Files\nodejs\node.exe .\script.js` is at the top. Below it, the characters of the string "Navpreet" are printed on separate lines: N, a, v, p, r, e, t. The character 'e' on the sixth line is highlighted with a blue circle and a blue cursor.

5. for-in Loop (for objects)

```
let obj = { name: "Navpreet", age: 20 };  
for (let key in obj) {  
  console.log(key + ": " + obj[key]);  
}
```



A terminal window with a black background and yellow text. The command `C:\Program Files\nodejs\node.exe .\script.js` is at the top. Below it, the object's properties are printed: `name: Navpreet` and `age: 20` on separate lines.