



# LECTURE 19 — DOM IN JAVASCRIPT

---

## 🧠 FIRST-THOUGHT PRINCIPLE (Seedha Samjho)

👉 Web page ko samajhne ka sabse easy tareeka:

- **HTML** → ghar ka **नक्शा** (structure)
- **CSS** → ghar ki **सजावट** (style)
- **JavaScript + DOM** → ghar me kaam karne wale log (dynamic changes)

### 📌 Meaning

Browser tumhari HTML file ko ek **Object Tree** me convert karta hai — isi ko **DOM (Document Object Model)** kehte hain.

JavaScript isi DOM ko **read / change / add / delete** karti hai.

---

## 📦 BASE HTML (ONE ELEMENT EXAMPLE)

```
<!DOCTYPE html>

<html>

  <head>
    <title>Lecture 19</title>
  </head>

  <body>
    <h1 id="title" class="heading">Hello Bhai</h1>
  </body>

</html>
```

## 🌳 DOM TREE (HIGH-LEVEL VIEW)

window

  └ document

    └ html

```
└── body
    └── h1#title.heading ("Hello Bhai")
```

👉 Browser HTML ko **tree structure** me tod data hai jise JS easily samajh sake.

---

## 🌐 GLOBAL OBJECT & DOCUMENT

### ◆ window

Browser ka **global object**.

### ◆ document

`window` ke andar HTML ka **object representation**.

```
console.log(window.document); // poora HTML
console.log(document.body);   // <body>
console.log(document.head);   // <head>
console.log(document.title);  // page title
console.log(document.URL);   // page URL
```

### 💡 Concept Link

Jaise C++ me `#include <iostream>` se cheezein milti hain,  
waise JS me **window object** se `console.log`, `Math.random` milte hain.

---

## ❓ DOM KYA HAI (LINE ME)

👉 Jab hum browser ya Live Server me HTML run karte hain:

- Browser HTML padhta hai
  - Ek **document object** banata hai
  - Isi object tree ko **DOM** kehte hain
  - JavaScript isi DOM ko control karti hai
- 

## 🎯 ELEMENTS KO ACCESS Kaise KAREIN

## ① By ID (Single Element)

```
const el = document.getElementById('title');

console.log(el.innerHTML);
```

Output:

```
Hello Bhai
```

## ② By Class Name (Multiple Elements)

```
const list = document.getElementsByClassName('heading');

console.log(list[0]);
```

Output:

```
<h1 id="title" class="heading">Hello Bhai</h1>
```

### ⚠ Important Note

- `getElementsByClassName()` → **HTMLCollection**
- Ye array jaisa hota hai, **array nahi hota**
- Array methods chahiye → `Array.from()`

## ❓ QUESTION

Agar 100 elements same class ke ho?

## ✓ ANSWER

Haan bhai, **sab access honge** (loop/index se)

```
<div class="box">One</div>

<div class="box">Two</div>

<div class="box">Three</div>
```

```
const items = document.getElementsByClassName('box');
```

```
Array.from(items).forEach(item => {  
  console.log(item.innerHTML);  
});
```

Output:

One

Two

Three

---

## 🔍 “BEHIND THE SCENES” — ELEMENT OBJECT

```
console.dir(el);  
👉 console.dir() element ko JavaScript object ki tarah dikhata hai.
```

```
{  
  id: "title",  
  className: "heading",  
  tagName: "H1",  
  innerHTML: "Hello Bhai",  
  innerText: "Hello Bhai",  
  textContent: "Hello Bhai",  
  style: CSSStyleDeclaration {...},  
  __proto__: HTMLHeadingElement  
    → HTMLElement  
      → Element  
      → Node  
      → EventTarget  
      → Object  
}
```

- ❖ Browser <h1> ko **HTMLHeadingElement object** me convert karta hai.
- 

## ELEMENT MODIFY DEMO

```
el.innerText = "Namaste Bhai";  
  
el.style.color = "crimson";  
  
el.className = "heading hero";  
  
el.classList.add("new");  
  
el.classList.remove("heading");
```

### Final HTML Output

```
<h1 id="title" class="hero new" style="color: crimson;">  
    Namaste Bhai  
</h1>
```

---

## ❓ HTML ELEMENTS KO OBJECT KYU BANAYA JATA HAI?

### Seedha Jawaab:

JavaScript sirf **objects** ke saath kaam kar sakti hai.

Isliye browser HTML elements ko **objects me convert** karta hai.

---

## SHORT NOTES

- JS objects ke properties/methods samajhti hai
  - Har HTML element → DOM object ban jata hai
  - JS se hum:
    - .**innerText** → text change
    - .**style** → CSS change
    - .**appendChild()** → new element add
  - Actual implementation browser me C++/Rust me hota hai
  - JS ko sirf **wrapper object** milta hai
-

## DEBUG TIP

```
console.log(document); // poora DOM tree  
console.dir(document); // detailed object view
```

 `documnet`  
 `document`

---

## MINI RECAP (WHY / HOW)

- Browser HTML → DOM tree
  - DOM = Object model
  - JS DOM ko object jaise handle karti hai
  - ID → single element
  - Class → HTMLCollection
  - `console.dir()` → object structure
- 

## QUICK TIPS

- `innerText /.textContent` → text only
  - `innerHTML` → HTML + tags
  - `classList.add/remove/toggle` → best practice
  - `className` → puri class replace
  - `Array.from()` → collection → array
  - Debug → `console.dir(element)`
- 

## SUMMARY – EK NAZAR ME

- DOM = Browser ka Object Model of HTML
  - Har HTML element → JS object
  - Access:
    - ID → single
    - Class → multiple (HTMLCollection)
  - `document` print karke poora page inspect kar sakte ho
  - Objects me convert isliye hote hain kyunki JS page ko **dynamic** banati hai
-