

**D-02 Training**

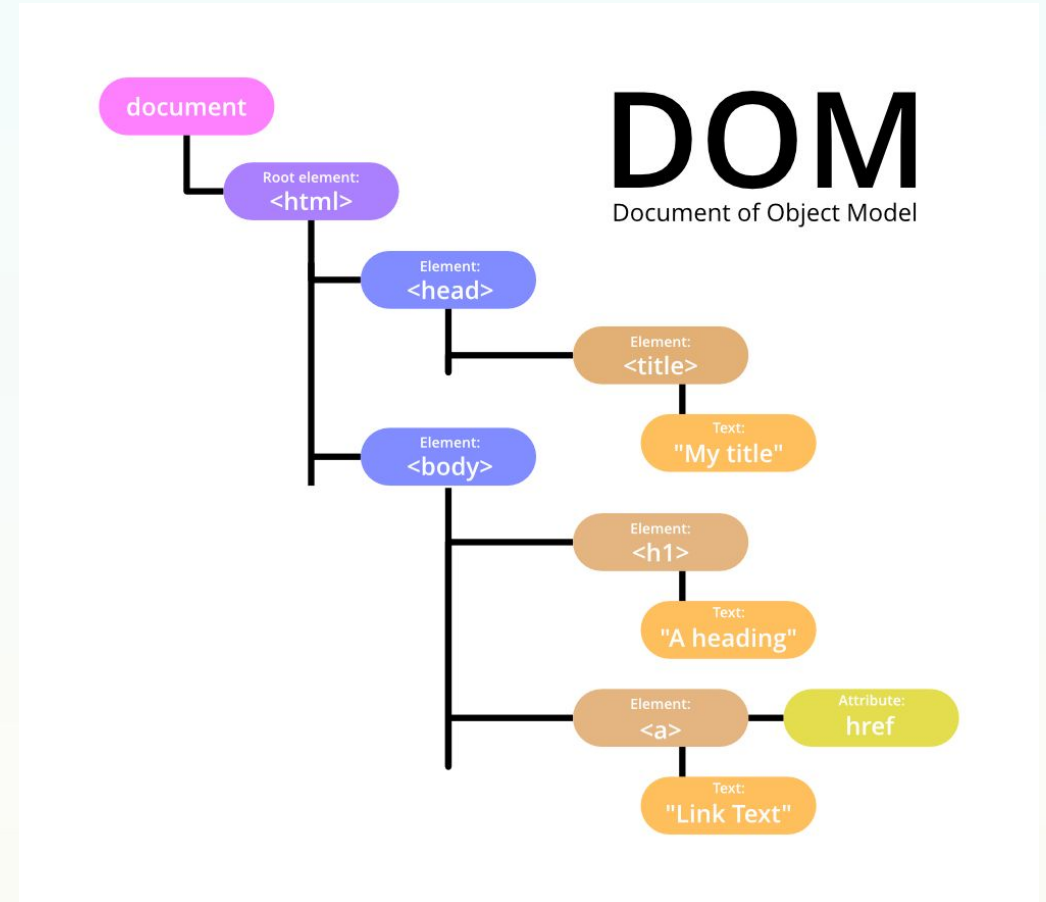
# JS DOM

## Document Object Model

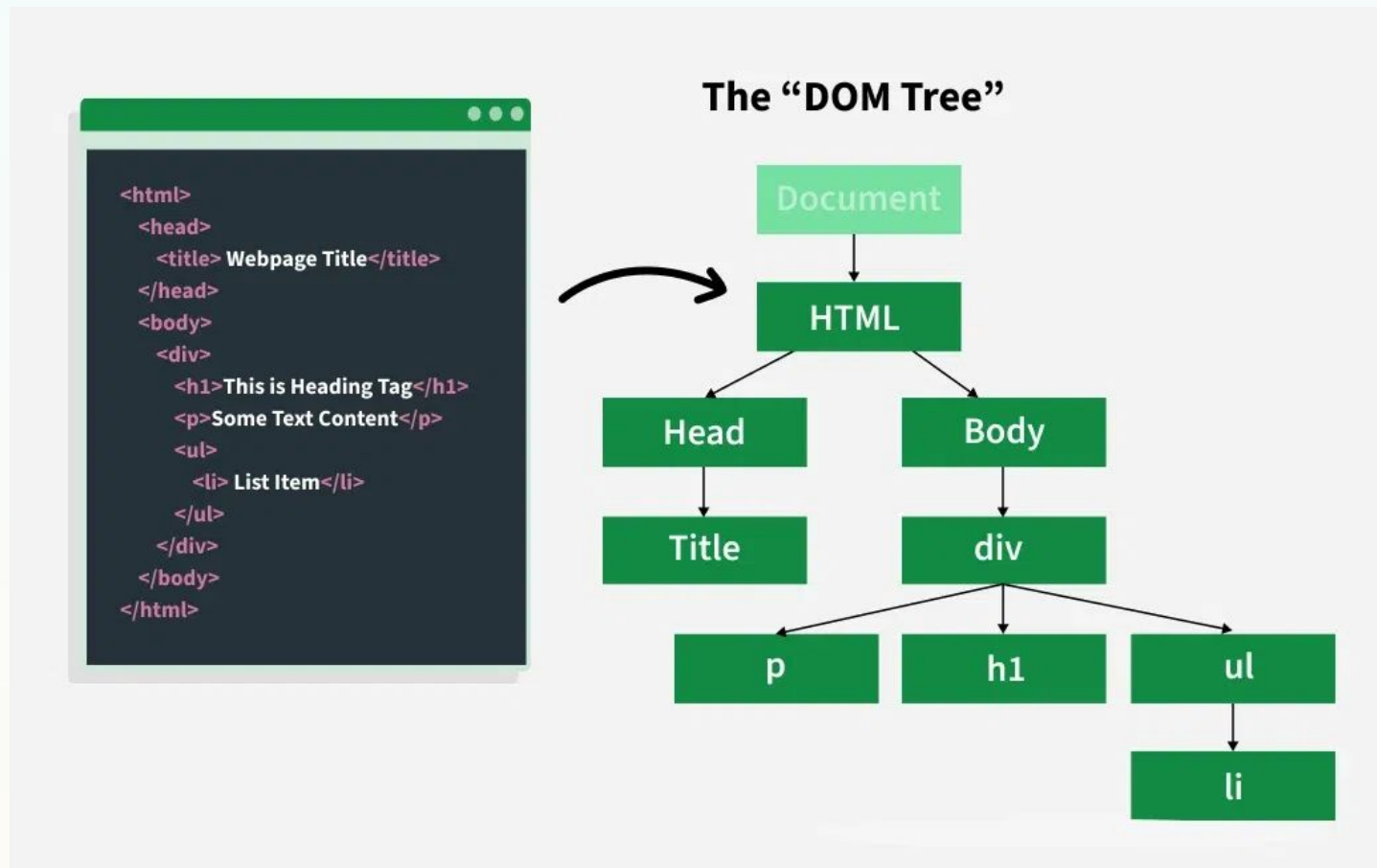


# DOM

- DOM (Document Object Model) adalah representasi struktur dokumen HTML dalam bentuk pohon objek yang dapat dimanipulasi menggunakan JavaScript.
- Memungkinkan interaksi dinamis dengan halaman web.



# DOM Tree



## **Kenapa kita butuh DOM?**

- Dynamic Content Updates
- User Interaction
- Flexibility
- Cross-Platform Compatibility

## Node pada DOM

- Elemen (Element Node): **<div>, <p>, <h1>**, dll.
- Atribut (Attribute Node): **id, class, src**, dll.
- Teks (Text Node): **Isi teks** dalam elemen.
- Komentar (Comment Node): **<!-- Ini komentar -->**
- Dokumen (Document Node): **Akar dari seluruh DOM.**

## Mengubah konten di dalam DOM

```
<html>
  <body>
    <p id="demo"></p>

    <script>
      document.getElementById("demo").innerHTML = "Hello World!";
    </script>
  </body>
</html>
```

## Menagkses element di dalam DOM

- `document.getElementById("myId");`
- `document.getElementsByClassName("myClass");`
- `document.getElementsByTagName("p");`
- **`document.querySelector(".myClass");`**
- `document.querySelectorAll("p");`

## Menagkses element di dalam DOM

- **getElementById**: Cepat tetapi hanya satu elemen.
- **getElementsByClassName** & **getElementsByTagName**:  
Mengembalikan koleksi.
- **querySelector**: Fleksibel tetapi lebih lambat dibanding  
getElementById.
- **querySelectorAll**: Mengembalikan NodeList yang bisa di-loop  
dengan forEach.



## Menagkses element di dalam DOM

- `let parent = document.getElementById("child").parentNode;`
- `let firstChild = document.getElementById("parent").firstChild;`
- `let lastChild = document.getElementById("parent").lastChild;`
- `let nextSibling = document.getElementById("child").nextSibling;`
- `let prevSibling = document.getElementById("child").previousSibling;`

# Mengubah Konten dan Atribut

## Mengubah teks dalam element:

```
document.getElementById("demo").textContent = "Hello World!";
```

## Mengubah atribut dalam element:

```
document.getElementById("img").setAttribute("src", "image.jpg");
```

## Menambahkan element

```
let newDiv = document.createElement("div");  
newDiv.textContent = "Elemen baru!";  
document.body.appendChild(newDiv);
```

## Menghapus element

```
let parent = document.getElementById("container");  
let child = document.getElementById("item");  
parent.removeChild(child);
```

## DOM Event

Kode javascript biasanya akan dieksekusi saat sebuah event terjadi, seperti saat user menekan suatu element HTML.

Untuk mengeksekusinya, silahkan tambahkan kode javascript pada suatu event tertentu seperti di bawah ini:

**onclick=JavaScript**

## DOM Event Example

- Saat user click mouse
- Saat halaman web di load
- Saat suatu gambar di load
- Saat kursor mengarah ke suatu element
- Saat input field berubah
- Saat form di submit
- Saat user menekan suatu tombol di keyboard

## Event Bubbling

- Ketika suatu event terjadi pada elemen anak, event tersebut akan naik ke elemen induknya sampai mencapai document (melewati semua ancestor).
- Event akan diproses dari elemen terdalam ke elemen luar

```
<div id="parent" style="padding: 20px; background: lightblue;">
  <button id="child">Klik Saya</button>
</div>

<script>
  document.getElementById("parent").addEventListener("click", function() {
    alert("Event pada Parent!");
  });

  document.getElementById("child").addEventListener("click", function() {
    alert("Event pada Child!");
  });
</script>
```

## Mari latihan

- Buatlah sebuah aplikasi todo app dengan fitur sebagai berikut:

- ✓ Menambahkan tugas
- ✓ Menghapus tugas
- ✓ Menandai tugas sebagai selesai

Silahkan akses templatanya melalui: <https://pastebin.com/EKkSqNC4>



**Thank You**

# JS DOM

## Document Object Model

