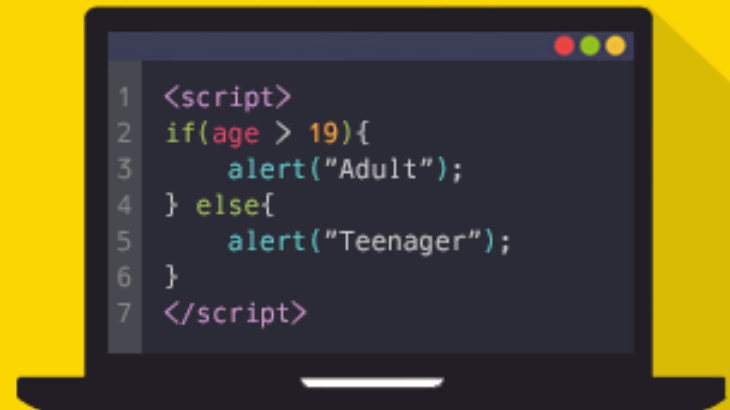




Assalamualaikum



JavaScript





Apa yang akan anda pelajari?

- Cara mengubah konten elemen HTML
- Cara mengubah gaya (CSS) elemen HTML
- Bagaimana bereaksi terhadap peristiwa-peristiwa DOM HTML
- Cara menambah dan menghapus elemen HTML



JavaScript HTML DOM

Model **objek** standar dan **antarmuka pemrograman** untuk HTML.

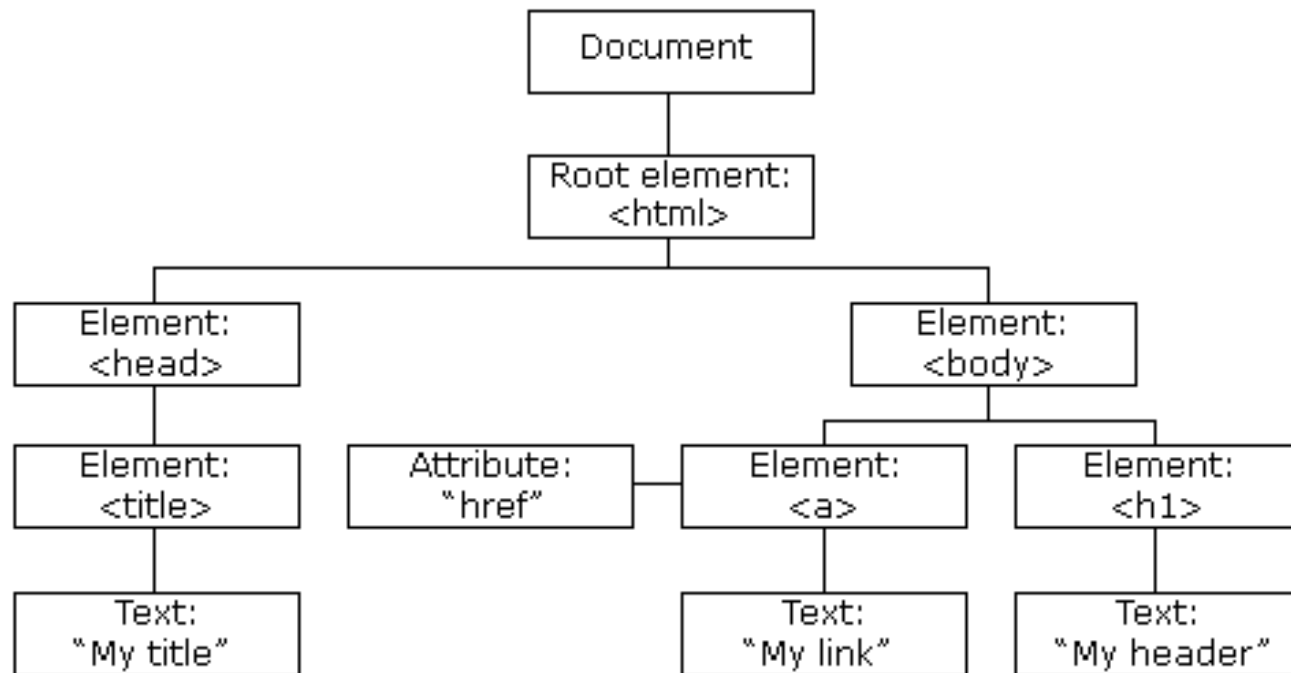
Ini mendefinisikan:

- Elemen HTML sebagai **objek**
- **Sifat** dari semua elemen HTML
- **Metode** untuk mengakses semua elemen HTML
- **Peristiwa** untuk semua elemen HTML

Dengan kata lain: **HTML DOM** adalah standar untuk cara **mendapatkan, mengubah, menambah, atau menghapus** elemen HTML.



Pohon Objek HTML DOM





Dengan model objek, JavaScript mendapatkan semua kekuatan yang diperlukan untuk membuat HTML dinamis:

- JavaScript dapat mengubah semua elemen HTML di halaman
- JavaScript dapat mengubah semua atribut HTML di halaman
- JavaScript dapat mengubah semua gaya CSS di halaman
- JavaScript dapat menghapus elemen dan atribut HTML yang ada
- JavaScript dapat menambahkan elemen dan atribut HTML baru
- JavaScript dapat bereaksi terhadap semua acara HTML yang ada di halaman
- JavaScript dapat membuat acara HTML baru di halaman



Antarmuka Pemrograman DOM

HTML DOM dapat diakses dengan JavaScript (dan dengan bahasa pemrograman lain).

Di DOM, semua elemen HTML didefinisikan sebagai **objek**.

Antarmuka pemrograman adalah properti dan metode masing-masing objek.

Sebuah **properti** adalah nilai yang Anda bisa mendapatkan atau set (seperti mengubah isi dari elemen HTML).

Sebuah **metode** adalah tindakan yang dapat Anda lakukan (seperti add atau menghapus elemen HTML).



Contoh

```
<html>
```

```
<body>
```

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

```
<script>
```

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

```
</script>
```

```
</body>
```

```
</html>
```



Metode getElementById

Cara paling umum untuk mengakses elemen HTML adalah dengan menggunakan id elemen tersebut.

Pada contoh di atas getElementById metode yang digunakan id="demo" untuk menemukan elemen.

Properti innerHTML

Cara termudah untuk mendapatkan konten suatu elemen adalah dengan menggunakan innerHTML properti.

The innerHTML properti ini berguna untuk mendapatkan atau mengganti isi dari elemen HTML.



Objek Dokumen DOM HTML

Jika ingin mengakses elemen apa pun di halaman HTML, Anda selalu mulai dengan mengakses objek dokumen. Di bawah ini adalah beberapa contoh bagaimana Anda dapat menggunakan objek dokumen untuk mengakses dan memanipulasi HTML.



Mencari Elemen Objek Dokumen

Seringkali, dengan JavaScript, Anda ingin memanipulasi elemen HTML.

Untuk melakukannya, Anda harus menemukan elemen terlebih dahulu. Ada beberapa cara untuk melakukan ini:

- Menemukan elemen HTML melalui id
- Menemukan elemen HTML dengan nama tag
- Menemukan elemen HTML dengan nama kelas
- Menemukan elemen HTML oleh penyeleksi CSS
- Menemukan elemen HTML dengan koleksi objek HTML



Mencari Elemen Objek Dokumen

Metode mencari elemen akan mengembalikan nilai sesuai dengan metode yang digunakan. Ada 3 tipe yang dikembalikan :

- Elemen => Mengembalikan 1 elemen HTML.

```
<p class="intro" id="nama">Par 1</p>
```

- HTML Collection => kumpulan elemen HTML.

```
HTMLCollection(3) [p#nama.intro, p.intro, p.intro, nama: p#nama.intro]
```

- Node List => kumpulan elemen dan informasi lengkap.

```
NodeList [p#nama.intro]
```



Menemukan Elemen HTML oleh Id

Cara termudah untuk menemukan elemen HTML di DOM, adalah dengan menggunakan id elemen.

```
document.getElementById("intro");
```

Jika elemen ditemukan, metode akan mengembalikan elemen sebagai objek.

Jika elemen tidak ditemukan, myElement akan berisi null.



Menemukan Elemen HTML dengan Nama Tag

Selain menggunakan Id kita juga dapat menggunakan `document.getElementsByTagName()` dan akan menghasilkan HTML Collection.

```
document.getElementsByTagName("p");
```

Jika elemen ditemukan, metode akan mengembalikan elemen sebagai HTML Collection.

Jika elemen tidak ditemukan, `myElement` akan berisi null.



Menemukan Elemen HTML dengan Nama kelas

Jika Anda ingin menemukan semua elemen HTML dengan nama kelas yang sama, gunakan `getElementsByClassName()`.

```
document.getElementsByClassName("intro");
```

Jika elemen ditemukan, metode akan mengembalikan elemen sebagai HTML Collection.

Jika elemen tidak ditemukan, `myElement` akan berisi `null`.



Menemukan Elemen HTML dengan querySelector

querySelector memungkinkan anda memanggil element dengan id, class maupun tag dan akan menghasilkan 1 element pertama yang ditemukan.

```
document.querySelector(".intro");  
document.querySelector("#intro");  
document.querySelector("p");
```

Jika elemen ditemukan, metode akan mengembalikan elemen sebagai objek.

Jika elemen tidak ditemukan, myElement akan berisi null.



Menemukan Elemen HTML dengan `querySelectorAll`

`querySelectorAll` memungkinkan anda memanggil element dengan id, class maupun tag dan akan menghasilkan node list dari element tersebut.

```
document.querySelectorAll(".intro");  
document.querySelectorAll("#intro");  
document.querySelectorAll("p");
```

Jika elemen ditemukan, metode akan mengembalikan elemen sebagai `nodeList`.

Jika elemen tidak ditemukan, `myElement` akan berisi `null`.



Mengubah Gaya HTML

HTML DOM memungkinkan JavaScript untuk mengubah gaya elemen HTML.
HTML DOM memungkinkan Anda untuk mengeksekusi kode ketika suatu peristiwa terjadi.

Acara dihasilkan oleh browser saat "sesuatu terjadi" pada elemen HTML :

- Suatu elemen diklik
- Halaman telah dimuat
- Kolom input diubah

`document.getElementById(id).style.property = 'new style'`



Mengubah Gaya HTML

Mengubah style :

```
<p id="p2">Hello World!</p>
<p>The paragraph above was changed by a script.</p>

<script>
  document.getElementById("p2").style.color = "blue";
</script>
```

Mengubah style dengan event :

```
<h1 id="id1">My Heading 1</h1>

<button type="button" onclick="document.getElementById('id1').style.color = 'red'">
  Click Me!
</button>
```



DOM attribute HTML

HTML DOM memungkinkan JavaScript untuk mengubah, mengambil dan menghapus attribut HTML.

- Mengambil nilai dari attribut
`element.getAttribute('namaAttribut')`
- Membuat attribute
`element.setAttribute('namaAttribut','nilai')`
- Menghapus attribute
`element.removeAttribute('namaAttribut')`



DOM ClassList HTML

HTML DOM memungkinkan JavaScript untuk mengubah, mengambil, menghilangkan/menampilkan dan menghapus class HTML.

- *Add* = Menambahkan class
- *Remove* = Menghapus class
- *Toggle* = Menghilang/menampilkan class
- *Item* = Mengambil nilai class sesuai index
- *Contains* = Mengecek apakah ada class atau tidak
- *Replace* = Mengganti class lama dengan class baru



DOM ClassList HTML

- *Add*

`element.classList.add('namaKelas')`

- *Remove*

`element.classList.remove('namaKelas')`

- *Toggle*

`element.classList.toggle('namaKelas')`

- *Item*

`element.classList.item('noIndex')`

- *Contains*

`element.classList.contains('namaKelas')`

- *Replace*

`element.classList.replace('KelasLama','KelasBaru')`

Referensi :

<https://www.w3schools.com/js/>