

Nama : Tantri Romadhoni Siswining Ndaru
NIM : 202210370311240
Kelas : Pemrograman Web- G

1. Mengakses Elemen DOM

- **getElementById("content")**: Digunakan untuk mengakses elemen berdasarkan ID-nya. ID harus unik di dalam dokumen.

kode javascript :

```
<div id="demo">Ini adalah elemen dengan ID 'demo'.</div>  
<script>  
    const element = document.getElementById("demo");  
    console.log(element); // Output: <div id="demo">Ini adalah elemen  
dengan ID 'demo'.</div>  
</script>
```

Kegunaan: Memudahkan pengambilan elemen secara langsung dengan ID yang spesifik. Mengambil elemen berdasarkan ID. Metode ini sangat efisien karena ID bersifat unik dalam halaman HTML.

- **querySelector("content")**: Mengambil elemen pertama yang sesuai dengan selector CSS yang diberikan.

kode javascript :

```
<div class="content">Ini elemen pertama.</div>  
<div id="unique">Elemen dengan ID 'unique'.</div>  
<script>  
    const element = document.querySelector(".content");  
    console.log(element); // Output: <div class="content">Ini elemen  
pertama.</div>  
</script>
```

Kegunaan: Dapat menggunakan selektor CSS apapun, termasuk class, ID, tag atau bahkan selektor kompleks(flaksibel). Mengambil elemen pertama yang sesuai dengan selektor CSS.

2. Mengubah Konten Elemen

- **textContent = "content"**: Sama seperti **innerText**, tetapi juga mengambil teks tersembunyi (misalnya, teks di dalam elemen yang disembunyikan dengan CSS).

kode javascript:

```
<div id="text-hidden">
```

```

    <span style="display: none;">Teks tersembunyi.</span>
    Ini teks yang terlihat.
</div>
<script>
    const element = document.getElementById("text-hidden");
    console.log(element.textContent); // Output: "Teks tersembunyi. Ini
    teks yang terlihat."
</script>

```

Kegunaan: Lebih cepat dari `innerText` dan juga mengambil/mengubah teks yang tersembunyi. Sama seperti `innerText`, namun juga mengambil teks tersembunyi di dalam elemen, misalnya teks yang tersembunyi oleh CSS.

3. Mengubah Atribut Elemen

- `setAttribute("attribute", "value")`: Menambahkan atau mengubah atribut elemen.

kode javascript :

```


<script>
    const image = document.getElementById("image");
    image.setAttribute("src", "new-image.jpg");
</script>

```

Kegunaan: Mengubah atribut elemen seperti `class`, `src`, `href`, dan lainnya. Atau menambahkan atribut HTML seperti `src`, `href`, atau `class`.

- `getAttribute("attribute")`: Mengambil nilai dari atribut tertentu.

kode javascript :

```

<a id="link" href="https://example.com">Go to Example</a>
<script>
    const link = document.getElementById("link");
    const hrefValue = link.getAttribute("href");
    console.log(hrefValue); // Output: "https://example.com"
</script>

```

Kegunaan: Mengambil nilai dari atribut elemen. Mengambil nilai dari atribut tertentu, seperti `href`, `src`, atau `id`.

- `removeAttribute("attribute")`: Menghapus atribut dari elemen.

kode javascript :

```
<button id="btn" disabled>Click Me</button>
<script>
  const button = document.getElementById("btn");
  button.removeAttribute("disabled");
</script>
```

Kegunaan: Menghapus atribut tertentu dari elemen. Misalnya, bisa menghapus atribut `disabled` dari sebuah tombol agar bisa diklik.

4. Event Handling (Penanganan Peristiwa)

- **`addEventListener("event", function)`:** Menambahkan event listener ke elemen untuk menangani event tertentu, seperti klik, hover, dll.

kode javascript :

```
<button id="click-btn">Klik saya!</button>
<script>
  const button = document.getElementById("click-btn");
  button.addEventListener("click", function() {
    alert("Tombol diklik!");
  });
</script>
```

Kegunaan: Memungkinkan elemen untuk merespons aksi pengguna seperti klik, mouse hover, atau penekanan tombol. Menambahkan event listener ke elemen. Dalam contoh ini, ketika tombol diklik, fungsi yang terkait akan dipanggil.