

LU04a - Das DOM (Document Object Model)

Was ist das DOM?

Das DOM ist die Art und Weise, wie JavaScript Ihr HTML-Dokument «sieht». Es repräsentiert die gesamte Seite als einen **Baum aus Objekten**. Jedes HTML-Element wird zu einem Objekt, das Sie lesen und verändern können.

Einfach erklärt: HTML schreiben Sie als Text. Das DOM ist die «lebendige» Version davon – eine Datenstruktur, mit der JavaScript arbeiten kann. Ändern Sie etwas im DOM, sehen Sie es sofort im Browser.

Der DOM-Baum

Jede Verschachtelung in HTML wird im DOM zu einer Eltern-Kind-Beziehung:

```
document
├── <html>
│   ├── <head>
│   │   └── <title> "Meine Seite"
│   └── <body>
│       ├── <header>
│       │   └── <nav>
│       └── <main>
│           └── <section>
│               └── <p> "Hallo Welt"
```

Elemente auswählen

Um ein Element zu verändern, müssen Sie es zuerst «abholen»:

Methode	Beschreibung	Hinweis
<code>querySelector()</code>	Erstes Element, das dem CSS-Selektor entspricht	Flexibelste Methode
<code>querySelectorAll()</code>	Alle passenden Elemente als NodeList	Für mehrere Elemente mit Schleife
<code>getElementById()</code>	Element mit bestimmter ID	Kein # im Parameter nötig
<code>getElementsByClassName()</code>	Alle Elemente einer Klasse	Kein . im Parameter nötig

```
// Ein einzelnes Element holen
const btn = document.querySelector('#mein-button');
const logo = document.getElementById('logo');

// Mehrere Elemente greifen
const allePanels = document.querySelectorAll('.panel');
```

```
// Mit NodeList arbeiten (forEach)
allePanels.forEach(panel => {
  panel.classList.add('hidden');
});
```

Elemente verändern

classList

Die sauberste Methode für Style-Änderungen – Sie definieren den Stil im CSS und schalten die Klasse per JS:

```
const panel = document.querySelector('.panel');

panel.classList.add('open'); // Klasse hinzufügen
panel.classList.remove('open'); // Klasse entfernen
panel.classList.toggle('open'); // Umschalten (add ↔ remove)
panel.classList.contains('open'); // true oder false zurückgeben
```

Attribute setzen und lesen

```
// Attribut setzen
btn.setAttribute('aria-expanded', 'true');

// Attribut lesen
btn.getAttribute('aria-expanded'); // → 'true'
```

Elemente ein- und ausblenden

Die eleganteste Methode: den Zustand im CSS definieren und per JS die Klasse togglen.

```
/* CSS: standardmässig versteckt */
.panel {
  display: none;
}

/* CSS: sichtbar wenn Klasse 'open' gesetzt */
.panel.open {
  display: block;
}
```

```
// JS: Klasse togglen
const panel = document.querySelector('.panel');
```

```
panel.classList.toggle('open'); // ein ↔ aus
```

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

https://wiki.bzz.ch/de/modul/m291/learningunits/lu04/theorie/a_dom?rev=1772396423

Last update: **2026/03/01 21:20**

