

LU08a - HTML-DOM

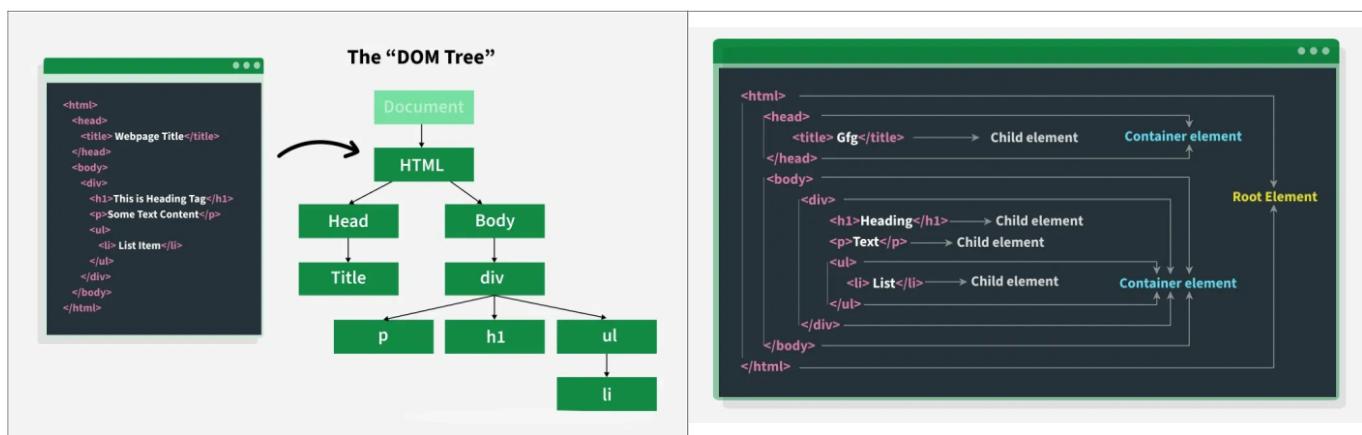
Lernziele

1. Den Begriff **HTML-DOM** in eigenen Worten beschreiben
2. Zusammenhang zwischen JavaScript und HTML-DOM darlegen können.
3. Den Zugriff JS-Zugriff auf HTML-DOM Elemente beschreiben können.

Einleitung

Der Begriff **HTML-DOM** (Document Object Model) bezeichnet die strukturierte, hierarchische Repräsentation eines HTML-Dokuments in einer Form, die von Programmiersprachen verarbeitet werden kann. Sobald ein Webbrowser eine HTML-Datei lädt, erzeugt er aus dem Dokument eine baumartige Datenstruktur, in der jedes HTML-Element (z. B. <body>,

, <p>) als eigener Knoten dargestellt wird.



Attribute, Textinhalte und Beziehungen zwischen Elementen werden ebenfalls in dieser Struktur abgebildet.

Zusammenhang mit JavaScript

JavaScript nutzt das DOM, um Inhalte einer Webseite dynamisch zu verändern, ohne die Seite neu zu laden. Der DOM-Baum dient hierbei als Schnittstelle: JavaScript kann auf jedes Element zugreifen, Eigenschaften auslesen, Werte ändern, neue Elemente erzeugen oder bestehende entfernen. Dadurch wird die statische HTML-Struktur in ein interaktives und reagierendes System überführt.

Zugriff auf DOM-Elemente

JavaScript stellt verschiedene Methoden bereit, um auf Knoten im DOM zuzugreifen. Beispiele sind:

Codebeispiel

```
<!DOCTYPE html>
```

```
<html lang="de">
<head>
    <meta charset="UTF-8">
    <title>Demo – DOM Manipulation</title>
    <style>
        #box {
            width: 200px;
            height: 100px;
            background-color: salmon;
            padding: 10px;
            margin-top: 10px;
        }
    </style>
</head>
<body>
    <h1 id="titel">Alter Titel</h1>
    <div id="box">
        Dies ist eine Box.
    </div>
    <button onclick="machWas()">Klick mich</button>
    <script>
        function machWas() {
            document.getElementById("titel").textContent = "Neuer Titel";
            document.getElementById("box").style.backgroundColor =
"lightgray";
        }
    </script>
```

Im obigen Code wird beim Drücken des Buttons die Farbe in der Box geändert und der Titel von *Alter Titel* in *Neuer Titel* geändert, wobei dafür die HTML-DOM-methode *getElementById* verwendet wird.

Änderung von DOM-Inhalten

Durch DOM-Manipulation lassen sich Struktur, Inhalt und Darstellung beeinflussen.

Beispiele:

```
document.getElementById("titel").textContent = "Neuer Titel";
document.getElementById("box").style.backgroundColor = "lightgray";
//
const neu = document.createElement("p");
neu.textContent = "Dynamisch erzeugter Absatz";
document.body.appendChild(neu);
```

In diesem Beispiel wird Text ersetzt, ein Stil angepasst und ein neues Element erzeugt und in das Dokument eingefügt.

Ereignisverarbeitung (Event Handling)

Das DOM ermöglicht die Verarbeitung von Benutzerinteraktionen. JavaScript kann sogenannte „Event

Listener“ registrieren, die auf Ereignisse reagieren, beispielsweise Mausklicks oder Tastatureingaben.

```
document.getElementById("button").addEventListener("click", function () {  
    alert("Schaltfläche wurde gedrückt.");  
});
```

Lernvideos

[HTML-DOM Explained](#)

Zusatzmaterial

- [W3School - HTML-DOM](#)
- [SelfHTML- HTML DOM](#)



Volkan Demir

From:

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

Permanent link:

<https://wiki.bzz.ch/de/modul/m288/learningunits/lu08/01?rev=1765200803>



Last update: **2025/12/08 14:33**