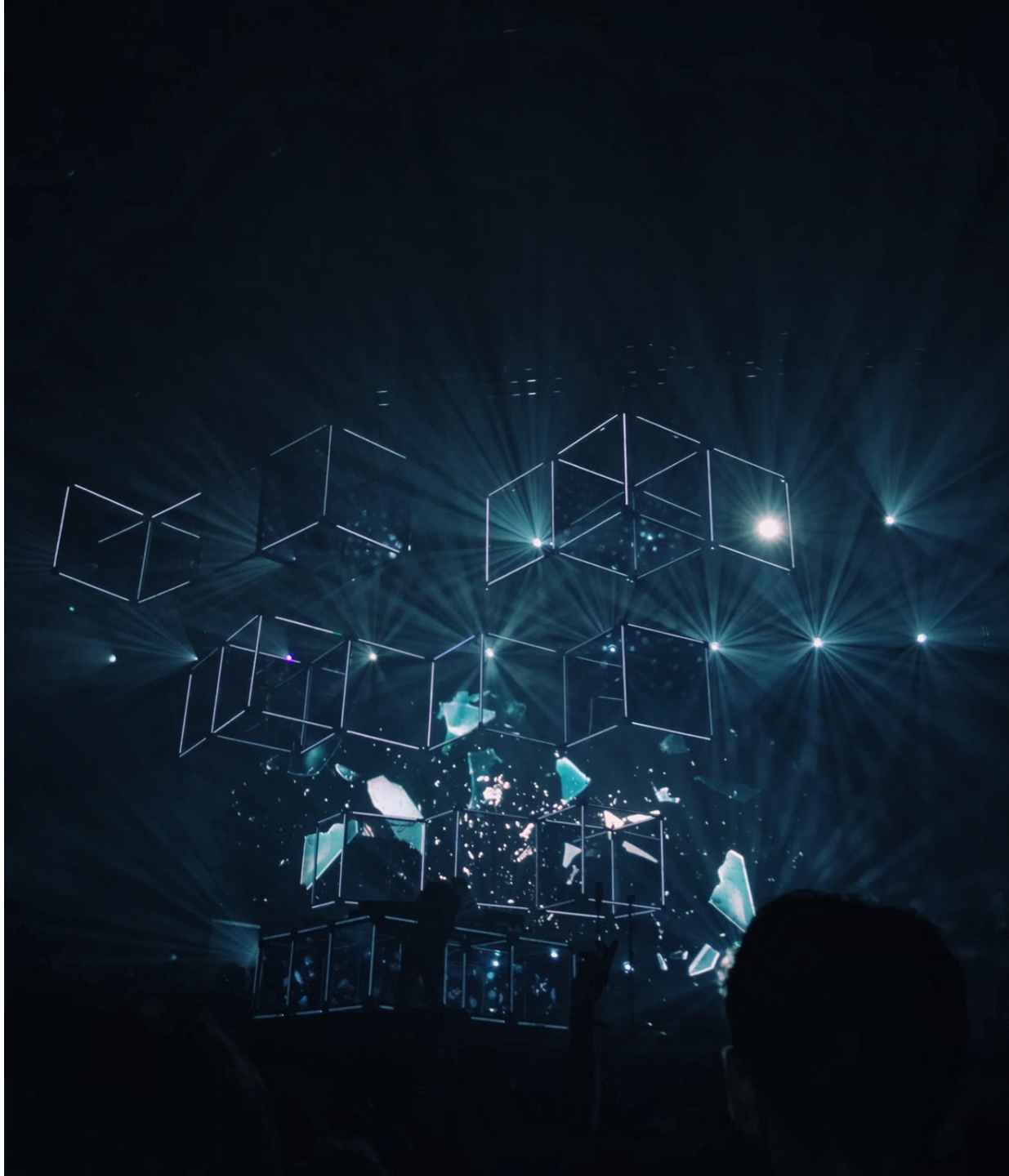
Das DOM

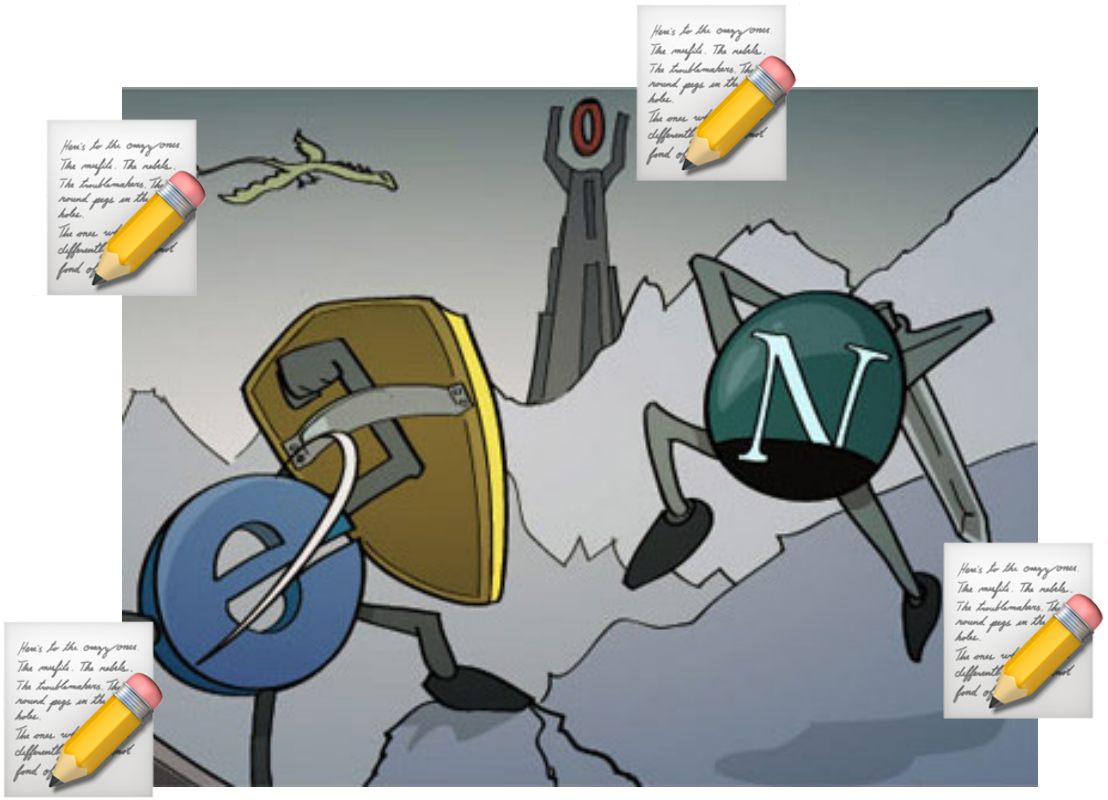
Ist der Club.

Und alles was sich im Club befindet.

Zum Beispiel die Bühne, die Bar, Schweinwerfer, Gäste, Gläser …



DOM Geschichte



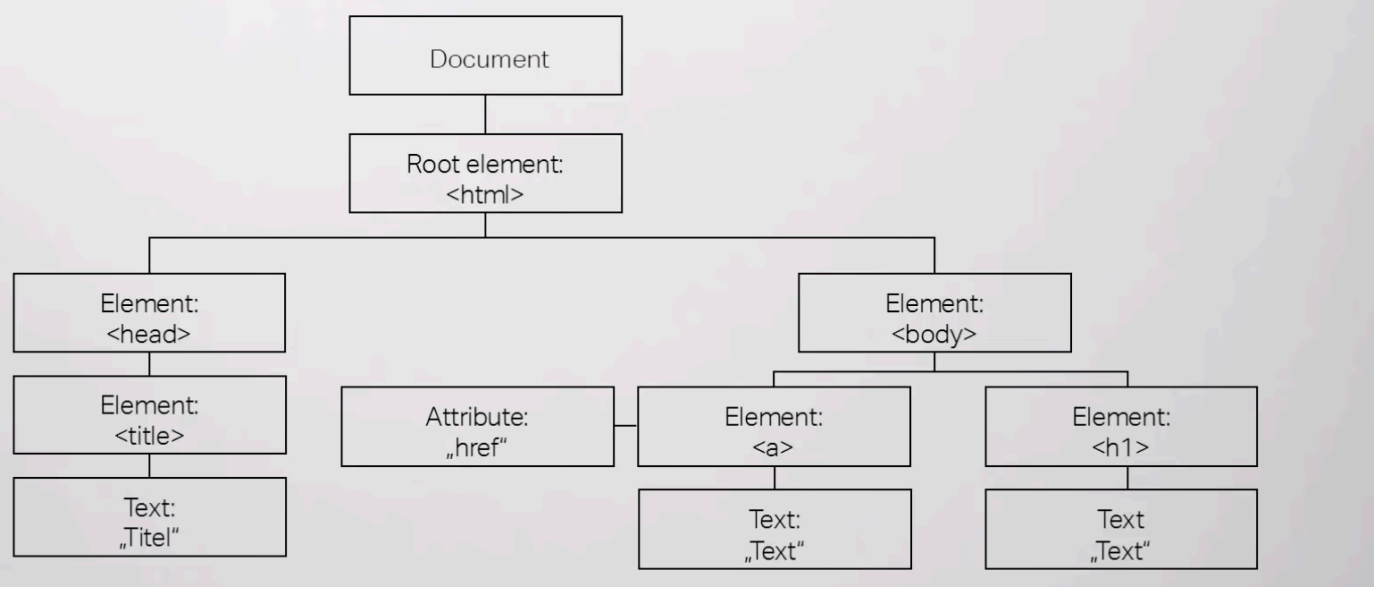
Der Browserkrieg (1995 – 1998)

👉

W3C®

DOM vs. HTML

"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."

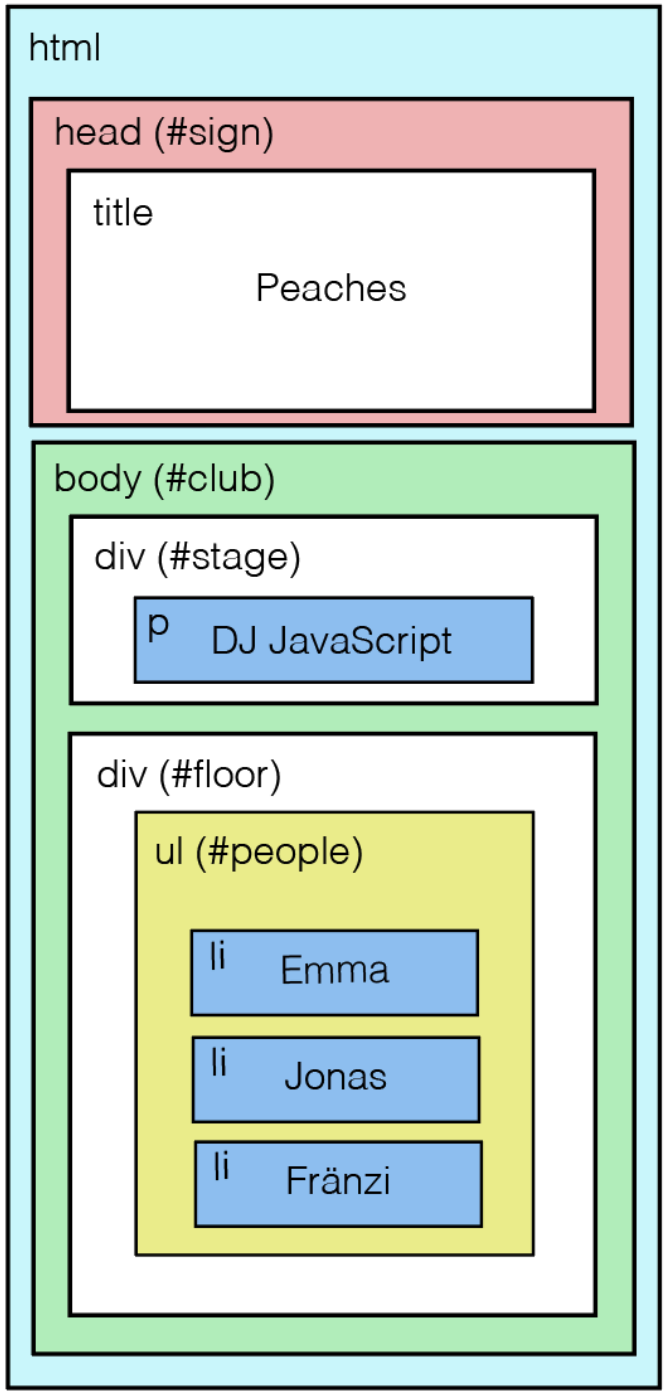


Document Object Model

Ist das Objektmodell (Object Model) und die Programmierschnittstelle für HTML

Das DOM definiert:

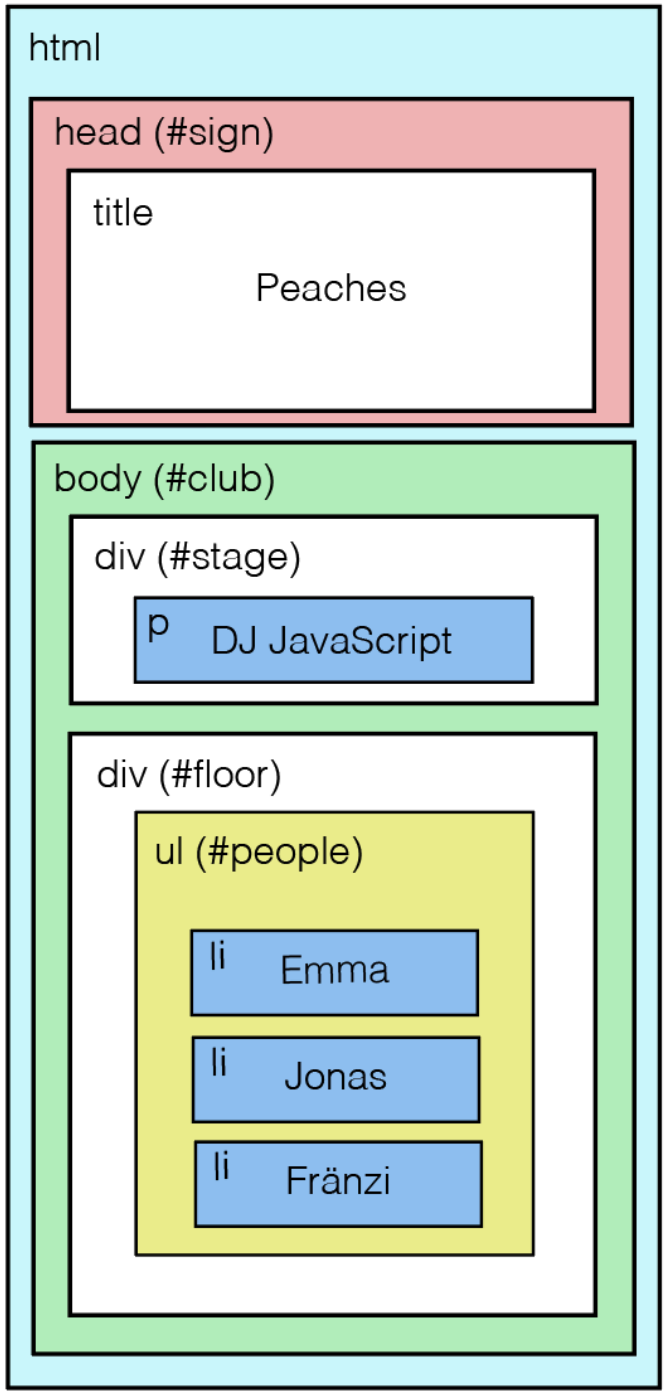
- HTML Elemente als Objekte
- Die Eigenschaften aller HTML Elemente
- Die Methoden, mit denen man auf Elemente zugreifen kann.
- Die Events aller HTML Elemente



Document Object Model

👉 siehe Beispiel 03.01.01 HTML

```
<html>

<head id="sign">

  <meta charset="utf-8">

  <title>Peaches</title>
</head>

<body id="club">

    <div class="room" id="stage">
      <p> DJ JavaScript </p>
    </div>

    <div class="room" id="floor">
      <ul id="people">
        <li> Emma </li>
        <li> Jonas </li>
        <li> Fränzi </li>
      </ul>
    </div>

<script src="js/control.js"></script>

</body>

</html>
```



DOM mit JS lesen

```
let stage = document.querySelector('#stage');

console.log(stage)
console.log(stage.id)
console.log(stage.tagName)
console.log(stage.firstElementChild);
```

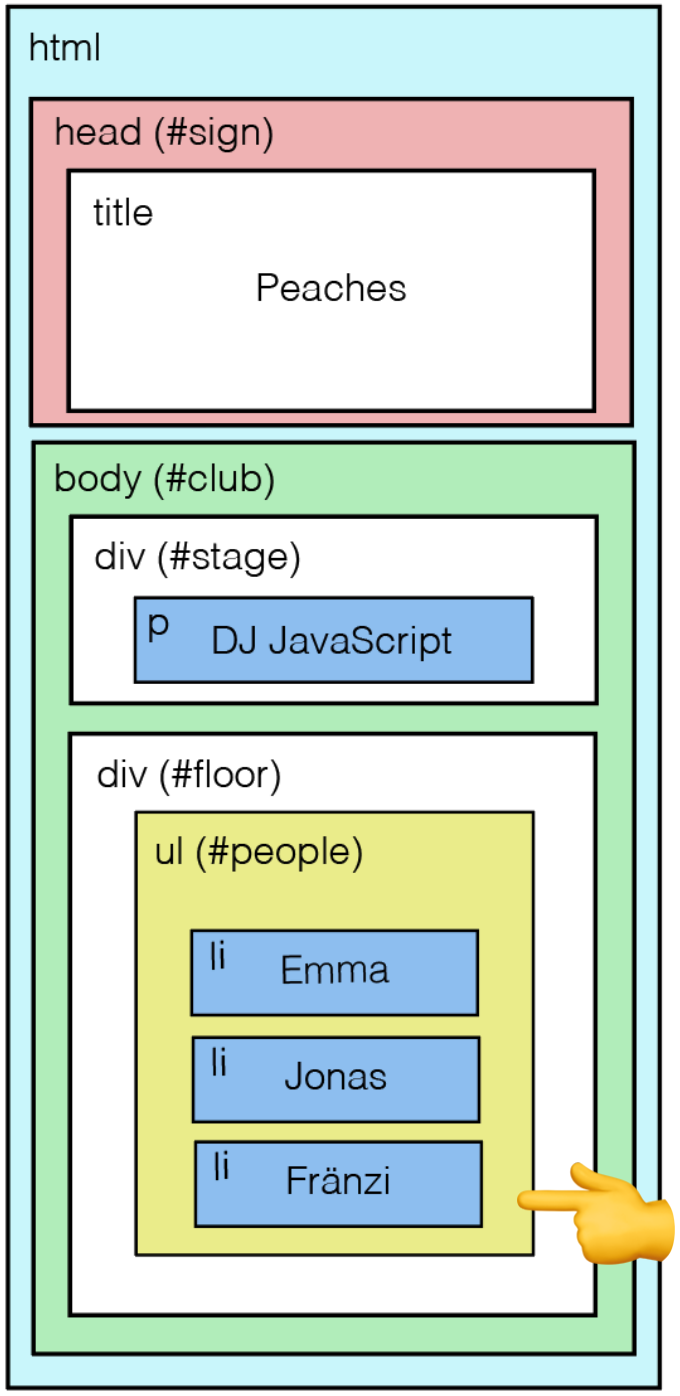
☝️ Das Element #stage als Objekt speichern. Mit stage.eigenschaft verschiedene Eigenschaften auslesen.

Möglichkeiten, DOM zu lesen

```
let club = document.getElementsByTagName("body");

let room = document.querySelector(.room');

let stage = document.querySelector('#stage');
```

👉 siehe Beispiel 03.01.01 JS

```
<html>

<head id="sign">

  <meta charset="utf-8">

  <title>Peaches</title>
</head>

<body id="club">

    <div class="room" id="stage">
      <p> DJ JavaScript </p>
    </div>

    <div class="room" id="floor">
      <ul id="people">
        <li> Emma </li>
        <li> Jonas </li>
        <li> Fränzi </li>
      </ul>
    </div>

<script src="js/control.js"></script>

</body>

</html>
```



DOM bearbeiten

Neues Element in Variable speichern

Inneres HTML dieses Elements setzen

Element mit appendChild dem DOM hinzufügen

```
let vladi = document.createElement("li");
vladi.innerHTML = 'Vladimir';

let people = document.querySelector('#people');
people.appendChild(vladi);
```

👉 siehe Beispiel 03.01.01

Der Tag

Fachhochschule Graubünden
University of Applied Sciences

⏰ 10 Minuten

```
<p> Meine Mutter hat <span id="farbe"> blaue </span> Augen.</p>
```

Das Element ist unsichtbar.

Das Element kann als Haken (hook) für Änderungen mit JS verwendet werden.

```
let farbe = document.querySelector('#farbe');
farbe.innerHTML = 'grüne';
```

👉 siehe Beispiel 03.01.02

Elemente stylen

```
<p> Meine Mutter hat <span id="farbe"> blaue </span> Augen.</p>

let farbe = document.querySelector('#farbe');
farbe.innerHTML = 'grüne';
farbe.style.color = 'green';
```

Style Object Properties:

https://www.w3schools.com/jsref/dom_obj_style.asp

👉 siehe Beispiel 03.01.02

Mit Klassen arbeiten

```
<p> Meine Mutter hat <span id="farbe"> blaue </span> Augen.</p>

let farbe = document.querySelector('#farbe');

farbe.classList.add('blue');
farbe.classList.remove('blue');
farbe.classList.toggle('blue');

.blue {
color: 'blue';
}
```

Klasse hinzufügen

Klasse entfernen

Schalter → Klasse hinzufügen oder entfernen.

👉 siehe Beispiel 03.01.03