

13.4 Javascript

Javascript ist eine Programmiersprache (und nicht zu verwechseln mit der Programmiersprache Java). Sie wurde entwickelt, um eine Möglichkeit zu haben, das HTML einer Webseite dynamisch (also nach dem Laden und Darstellen im Browser) zu verändern. Deshalb verstehen alle modernen Browser Javascript und du kannst die Möglichkeiten, die dir Javascript bietet, sofort ausprobieren.

Mittels Javascript lassen sich Webseiten «animieren» – Image-Slider sind Beispiele dafür, ein- oder ausblendbare Elemente, Dropdownmenüs, die einen Einfluss auf die Inhalte der Webseite haben, bewegte Elemente oder Spiele.

Das vorliegende Kapitel geht nur auf einige ausgewählte Aspekte von Javascript ein, für eine detaillierte Behandlung der Materie verweisen wir auf die folgenden Referenzen.

Referenzen & weiterführende Webseiten

<https://developer.mozilla.org/de/docs/Web/JavaScript>

<https://wiki.selfhtml.org/wiki/JavaScript>

<https://jquery.com/> sowie <https://jqueryui.com/>

Buch

Javascript, das umfassende Referenzwerk, David Flanagan, O'Reilly (in der Mediothek erhältlich)

Demos

<http://webdesign.kanti-baden.ch/demos/js>

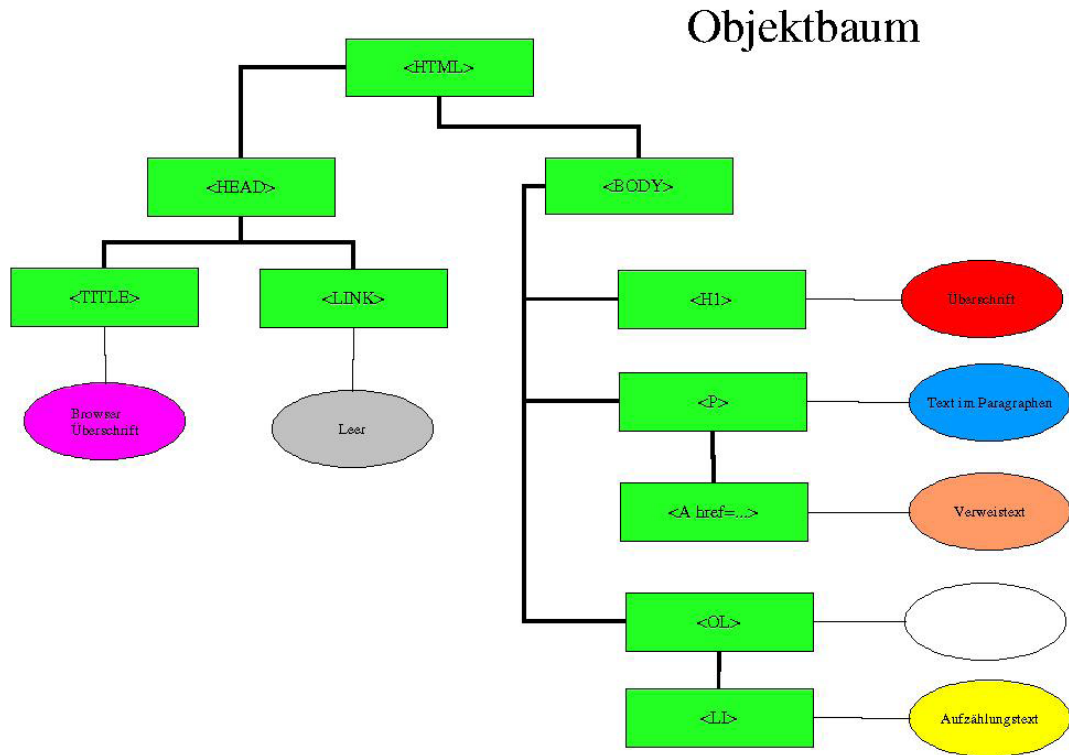


Abbildung 13.4.2: Der Objektbaum (DOM) nach der Modifikation durch CSS.

HTML, CSS, JavaScript und DOM

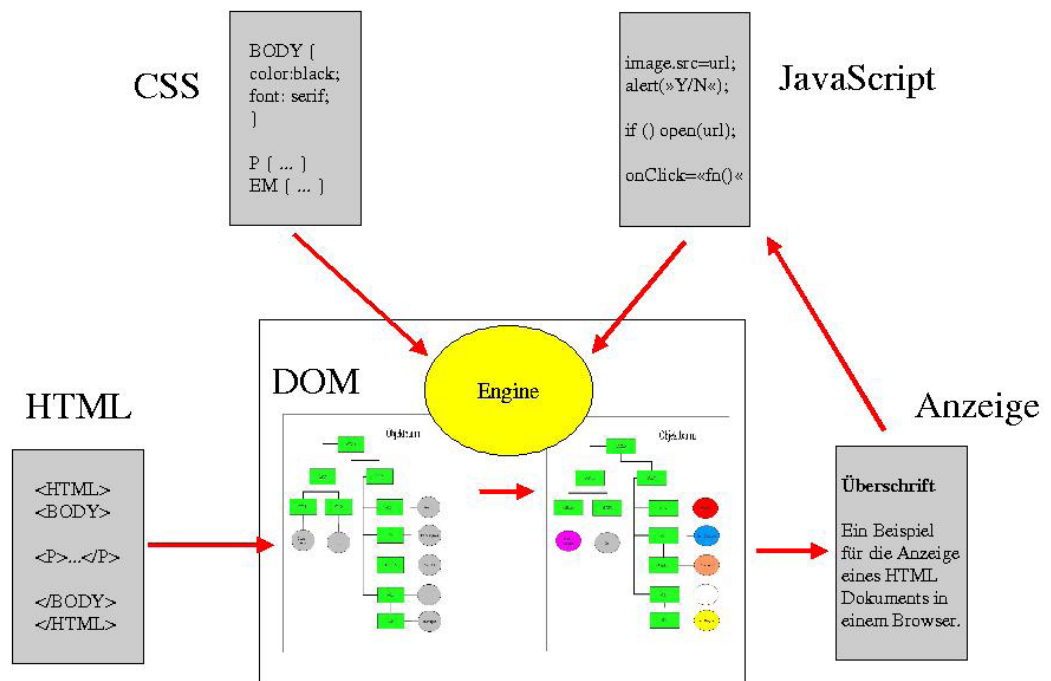


Abbildung 13.4.3: Das Zusammenspiel aller Komponenten einer Webseite..

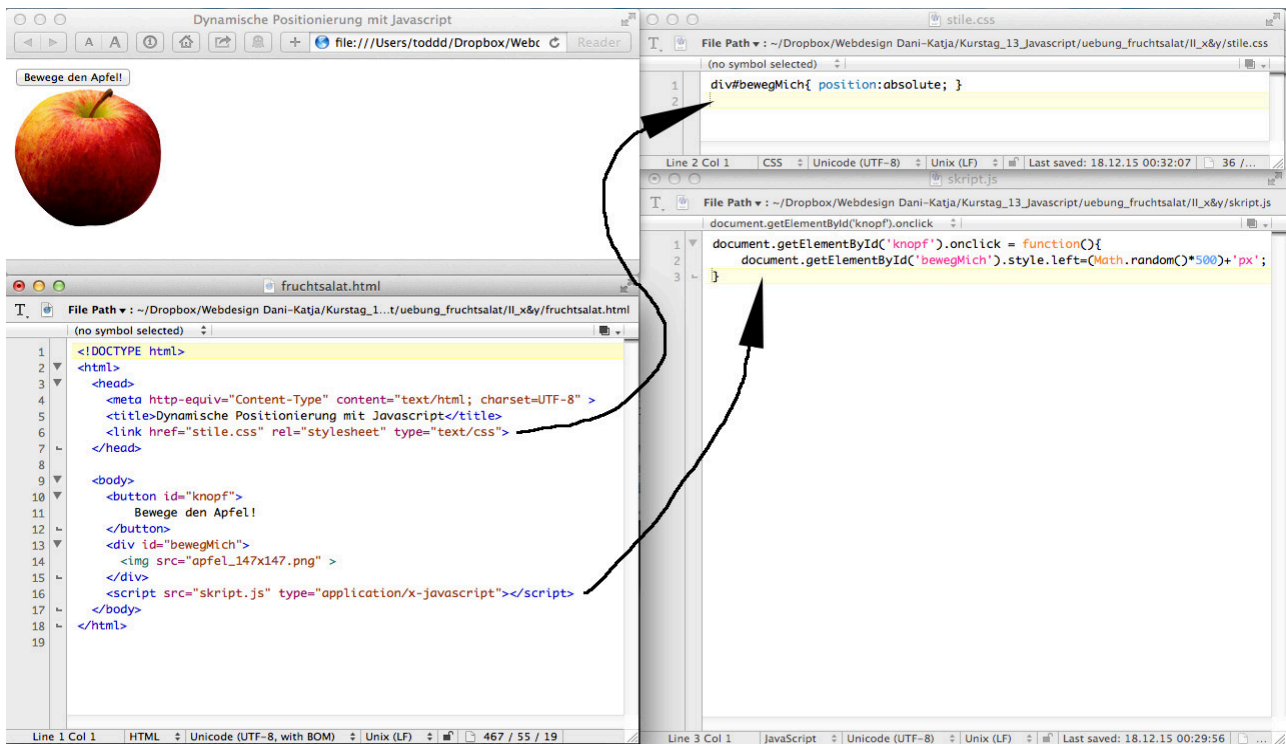


Abbildung 13.4.4c: Zusammenspiel der Dateien einer gängigen Webseite. Für die korrekte Browser-Ansicht braucht es eine HTML-Datei (unten links) welche über das `<link>`-Tag die CSS-Datei (oben rechts) und über das `<script>`-Tag eine Javascript-Datei (rechts unten) einbindet.

Mit Javascript auf ein Element der Webseite zugreifen.

Das Selektieren von Elementen des DOMs mit Javascript geht am einfachsten über die ID des Elements: Mit der Funktion `getElementById('id')` kann das Element mit der ID «id» selektiert werden (Abbildung 13.4.4a, Zeile 9).

Mittels der Funktion `alert('text')` kann ein Popup-Fenster geöffnet werden, das «text» anzeigt (Abbildung 13.4.4b):

Fortgeschritten:

Über die Funktion `getElementsByClassName('class')` erhält man einen Array mit allen Elementen des vorliegenden HTML-Dokuments, die der Klasse 'class' zugehörig sind.

Und auch noch: Es gibt auch noch die Funktion `getElementsByTagName('tag')`, welche einen Array mit allen Elementen des Tags 'tag' zurück liefert.

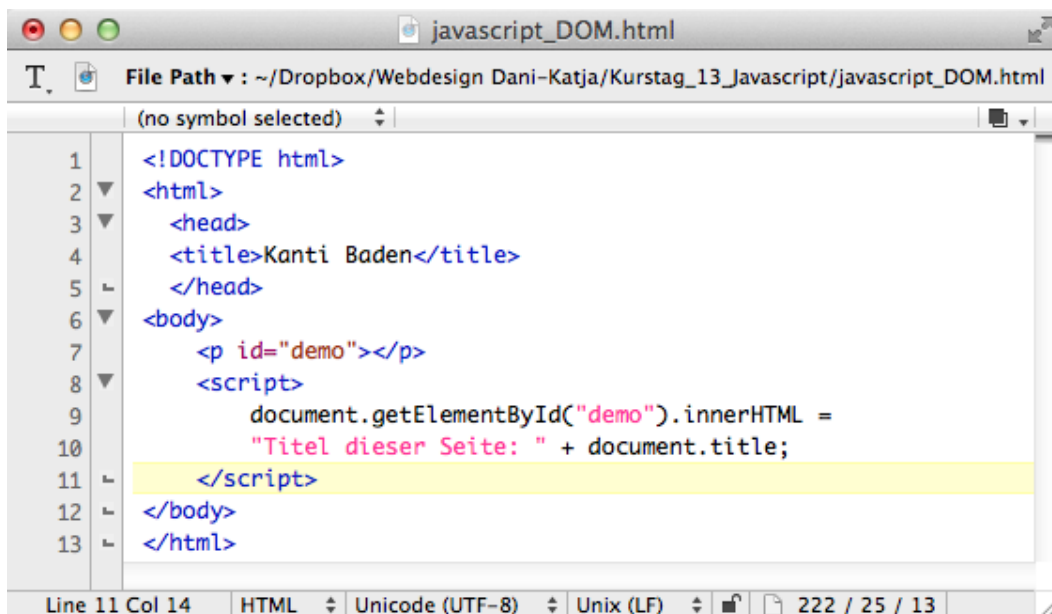
Elemente des DOMs verändern

Inhalt-Veränderung des Elements

Das Beispiel 13.4.5 zeigt, wie man mit Javascript dem `<P ID="demo">`-Tag einen Inhalt gibt: Den Titel der aktuellen Seite findet man im DOM über Javascript mittels `document.title` (Screenshot Abb. 13.4.5a, Zeile 9); er wird der Eigenschaft `<innerHTML>` des P-Tags zugewiesen.

Dieses Beispiel ist digital zu finden unter: http://webdesign.kanti-baden.ch/demos/js/javascript_DOM_alert.html

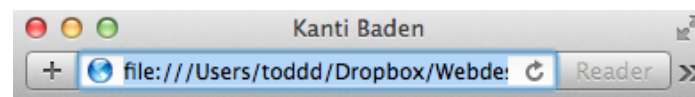
Ein weiteres, fertiges Beispiel, bei dem dynamisch die aktuelle Uhrzeit ins Dokument eingefügt wird, findest du unter http://webdesign.kanti-baden.ch/demos/js/innerHTML_zeit.



```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Kanti Baden</title>
5  </head>
6  <body>
7    <p id="demo"></p>
8    <script>
9      document.getElementById("demo").innerHTML =
10     "Titel dieser Seite: " + document.title;
11   </script>
12 </body>
13 </html>
```

Line 11 Col 14 HTML Unicode (UTF-8) Unix (LF) 222 / 25 / 13

Abbildung 13.4.5a: Der Inhalt des `<P>`-Tags wird dynamisch erstellt und mittels `innerHTML` eingefügt..



Titel dieser Seite: Kanti Baden

Display a menu

Abbildung 13.4.5b: Der Inhalt der Webseite wurde dynamisch mit Elementen aus dem DOM (`document.title`) erstellt.

Verändern der Stil-Angaben

Mittels der folgenden Javascript-Syntax kann auf die CSS-Stil-Angaben zugegriffen werden. Übrigens: Die Angaben können sowohl gelesen, als auch verändert werden:

```
document.getElementById("bewegmich").style.  
width = "250px";
```

Ein Beispiel findest du unter (Abbildung 13.4.4a, Zeile 9). Ausformulierte Beispiele für das Setzen einer Stil-Eigenschaft:
http://webdesign.kanti-baden.ch/demos/js/onclick_onmouseover
http://webdesign.kanti-baden.ch/demos/js/toggle_div

Verändern eines Bildes durch Anpassen seiner Quelle:

Möchte man ein Bild durch ein anderes ersetzen, kann man die Eigenschaft src des Bild-Objektes verändern:

```
document.getElementById("kaeferbild").src =  
'anderesbild.jpg';
```

Ein fertig programmiertes Beispiel dazu (wo ein Bildwechsel stattfindet, nachdem man mit der Maus über ein Bild fährt) findest du unter http://webdesign.kanti-baden.ch/demos/js/img_src_replacement.


```

1  <!DOCTYPE html>
2  <html dir="ltr" lang="de-ch">
3  <head>
4    <meta content="text/html; charset=utf-8" http-equiv="content-type">
5    <title>Javascript - onMouseOver - CSS</title>
6    <meta content="Daniel Süsstrunk" name="author">
7    <meta content="Javascript zur Demonstration des MouseOver-Eventhandlers"
8      name="description">
9    <link href="Javascript-onMouseOver-CSS.css" rel="stylesheet" type="text/css">
10 </head>
11 <body>
12   <div id="bewegmich">  </div>
13   <script src="Javascript-onMouseOver-CSS.js" type="application/x-javascript"></script>
14 </body>
15 </html>

```

Abbildung 13.4.7: Der Javascriptcode, welcher die Eventhandler registriert, wird direkt vor dem abschliessenden </BODY>-Tag in den Quellcode eingefügt (Zeile 13). Der Inhalt dieser Datei (Javascript-onMouseOver-CSS.js ist in der Abbildung 13.4.6 zu sehen.

es ist. Wird also das DIV vergrössert, wird auch das Bild grösser dargestellt. Der resultierende Effekt ist, dass das Bild grösser dargestellt wird, sobald die Maus über dem entsprechenden DIV-Element liegt.

Dieses Beispiel findest du fertig programmiert unter http://webdesign.kanti-baden.ch/demos/js/onclick_onmouseover.

Darin sind alle Javascript-Inhalte in der Datei Javascript-onMouseOver-CSS.js gespeichert. Diese Datei wird von der HTML-Datei aus vor dem abschliessenden </BODY>-Tag aufgerufen (siehe Abbildung 13.4.7).

Neben onclick sind weitere gängige Eventhandler die folgenden:

- onchange** (beim Verändern eines z.B. Input-Elements)
- ondblclick** (beim Doppelklicken)
- onfocus** (wenn ein Input-Element den Fokus erhält)
- onblur** (wenn ein Input-Element den Fokus verliert)
- onsubmit** (wenn ein Formular abgeschickt wird).

Eine vollständige Liste gibt es unter <https://wiki.selfhtml.org/wiki/JavaScript/DOM/Event/Übersicht>

jQuery

jQuery ist eine freie JavaScript-Bibliothek, die Funktionen zur DOM-Navigation und -Manipulation zur Verfügung stellt (<https://jquery.com/>). Die Benutzung von jQuery vereinfacht die Benutzung von Javascript, da jQuery Javascript-Browser-Unterschiede vor dem Programmierer versteckt. Dadurch laufen Javascript-Programme auf unterschiedlichen Browsern gleich, was den Radius einer Webseite beträchtlich erhöhen kann. Die

