HTML5, CSS3 und JavaScript 1.8

Fortgeschrittene Entwicklung von Webseiten

Isolde Kommer

1. Ausgabe, Februar 2013

HTML5F

HERDT

Impressum

Matchcode: HTML5F

Autorin: Isolde Kommer

Redaktion: Andreas Dittfurth, Andrea Weikert

Produziert im HERDT-Digitaldruck

1. Ausgabe, Februar 2013

HERDT-Verlag für Bildungsmedien GmbH
Am Kümmerling 21-25
55294 Bodenheim
Internet: www.herdt.com
E-Mail: info@herdt.com

2 Bereiche definieren und positionieren

In diesem Kapitel erfahren Sie

- wie Sie `div`-Container erstellen
- wie Sie `div`-Container positionieren
- wie Sie Bereiche mit semantischen Tags definieren

Voraussetzungen

- Grundlegende HTML- und CSS-Kenntnisse

2.1 Zeitgemäßes Webseitenlayout

Nachteile von Tabellen als Layoutwerkzeug

Bis vor wenigen Jahren war es in der Welt des Webdesigns üblich, Seitenelemente mithilfe unsichtbarer Tabellen an bestimmten Stellen auf der Seite zu fixieren. In rein visueller Hinsicht sind Tabellen für das Seitenlayout auch recht gut geeignet, sie haben aber schwerwiegende Nachteile.

- Um ein stabiles Seitenlayout zu erzielen, mussten transparente GIF-Bilder verwendet werden, um die Breiten und Höhen der unsichtbaren Tabellen, und damit die Abstände der Seitenlayoutelemente, zu fixieren.
- Bilder müssen häufig im Bildbearbeitungsprogramm zerstückelt (in "Slices" unterteilt) und in HTML wieder wie ein Puzzle zusammengesetzt werden.
- Für ältere Browser sind teilweise einander widersprechende Tabellenformatierungstags notwendig.
- Durch ständig wiederholte `table`- und `td`-Tags wird der Code unnötig aufgebläht.
- Tief greifende Umgestaltungen des Seitenlayouts sind schwierig und aufwendig.
- Struktur und Darstellung der Seite befinden sich komplett im HTML-Code.

Vorteile von `div`-Containern als Layoutwerkzeug

Aus diesem Grund hat das W3C die Empfehlung ausgesprochen, HTML-Tabellen nicht für Layoutzwecke zu verwenden, sondern ausschließlich für die Strukturierung von Tabellendaten.

Für das Seitenlayout hingegen verwenden Sie am besten mit CSS formatierte **`div`-Container** und gegebenenfalls die neuen HTML5-Tags für Kopf- und Fußzeilen, Artikel usw. Daraus ergeben sich folgende Vorteile:

- vollständige Trennung von Struktur und Darstellung,
- bessere Kontrolle über das Seitenlayout,
- schnelle und einfache Aktualisierung oder Umgestaltung.

Diese Herangehensweise bedeutet auch, dass Sie verschiedene Stylesheets für verschiedene Zwecke spezifizieren können, ohne mehrere Versionen Ihrer Site verwalten zu müssen. Damit können Sie z. B. eine abgespeckte Version für PDAs, eine Hochkontrast- oder eine Nur-Text-Version für Sehbehinderte anbieten. Sogar eine spezielle Druckversion lässt sich ohne allzu großen Aufwand realisieren - Sie benötigen dazu nur ein Stylesheet mit geeigneten Farben, Seitenrändern usw.

Vorgehensweise beim Gestalten des Seitenlayouts mit `div`-Containern

Um ein Seitenlayout mit `div`-Containern aufzubauen, können Sie beispielsweise folgendermaßen vorgehen. Nach dieser Reihenfolge richtet sich auch der Aufbau dieses Kapitels.

Container erstellen	Zunächst fertigen Sie eine Skizze des gewünschten Seitenlayouts an. Dazu können Sie auch ein Blatt Papier verwenden. Erstellen Sie anschließend die benötigten `div`-Container. Für jeden Seitenbereich - etwa Hauptnavigation, Inhaltsbereich usw. - erstellen Sie einen eigenen `div`-Container.
Container formatieren	Weisen Sie den `div`-Containern per CSS die gewünschten Formatierungsmerkmale wie Hintergrundfarbe, Rahmen, Schriftmerkmale usw. zu. Dabei definieren Sie für jeden `div`-Container die jeweils benötigte Breite und eventuell eine Höhe.
Container positionieren	Positionieren Sie die einzelnen `div`-Container an der gewünschten Stelle der Seite.

2.2 `div`-Container erstellen

Bereiche definieren

Trennen Sie einzelne Gestaltungseinheiten bzw. Layout-Bereiche, die Sie an verschiedenen Stellen der Seite verteilen möchten, durch `div`-Elemente, auch `div`-Container genannt.

`div`	`div` leitet sich vom englischen "division" (Bereich) ab und bezeichnet einen gemeinsamen Bereich, in den Sie mehrere Inhalte aller Art einschließen können, beispielsweise Texte, Grafiken, Datentabellen usw. Wenn Sie einen `div`-Container nicht mit Cascading Style Sheets formatieren, beginnen alle in den Container eingeschlossenen Elemente in einer neuen Zeile.

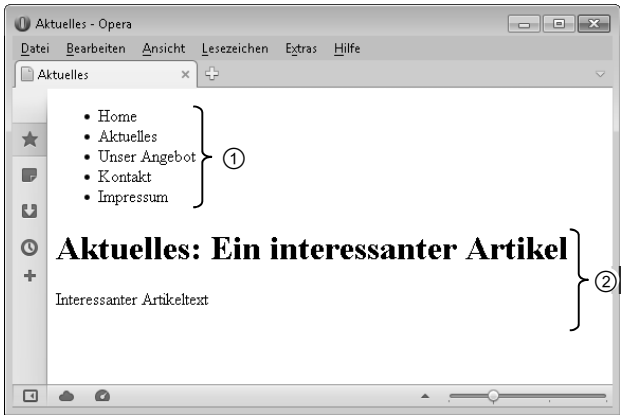
Beispiel: Zwei `div`-Container ohne CSS-Formatierung (*kap02|div-container.html*)

```
①  <div>
     <ul>
       <li>Home</li>
       <li>Aktuelles</li>
       <li>Unser Angebot</li>
       <li>Kontakt</li>
       <li>Impressum</li>
     </ul>
   </div>
②  <div>
     <h1>Aktuelles: Ein interessanter Artikel</h1>
     Interessanter Artikeltext
   </div>
```

Hier wurde ein `div`-Container für die Navigation ① und ein weiterer `div`-Container für den Seiteninhalt ② erstellt.

Da es keinerlei CSS-Formatierungen gibt, werden die beiden `div`-Container untereinander dargestellt, wie die nebenstehende Abbildung zeigt.

`div`-Bereiche können auch ineinander verschachtelt werden (vgl. Abschnitt 2.6).



Unformatierte `div`-Container

2.3 `div`-Container mit CSS formatieren

Die in Ihrem Dokument definierten `div`-Bereiche können Sie mit CSS gestalten.

Beispiel: Zwei `div`-Container mit CSS-Formatierung (*kap02\div-container-formatiert.html*)

- Öffnen Sie die Beispieldatei *div-container.html.*
- Definieren Sie innerhalb des Tagpaars `<head> ... </head>` die folgenden CSS-Stildefinitionen, um die Seite mit einem eingebetteten Stylesheet zu versehen:

```
<style type=text/css>
①  #Nav {
        font-family:Arial, Helvetica, sans-serif;
        font-size:small;
        background-color:#FFC;
        border:2px dotted #600;
   }
②  #Artikel {
        border:3px double #600;
        padding:3px;
   }
   </style>
```

① Erstellen Sie eine ID für die Navigation. Weisen Sie der ID die gewünschten CSS-Formatierungen zu.
② Erstellen Sie eine ID für den Seiteninhalt und weisen Sie der ID die gewünschten CSS-Formatierungen zu.

```
③  <div id="Nav">
     <ul>
       <li>Home</li>
       <li>Aktuelles</li>
       <li>Unser Angebot</li>
       <li>Kontakt</li>
       <li>Impressum</li>
     </ul>
   </div>
④  <div id="Artikel">
     <h1>Aktuelles: Ein interessanter Artikel</h1>
     <p>Interessanter Artikeltext</p>
   </div>
```

③ Im `body`-Bereich des Dokuments weisen Sie dem ersten `div`-Container die ID `Nav` zu.

④ Weisen Sie dem zweiten `div`-Container die ID `Artikel` zu.

Hier wurden für die Formatierung IDs verwendet, da die jeweiligen Elemente auf der Seite nur ein einziges Mal vorkommen. Wenn die Elemente auf der Seite mehrfach vorkommen, verwenden Sie stattdessen Klassen.

`div`-Container mit CSS-Formatierung

`div`-Containern eine Breite und eine Höhe zuweisen

Wenn Sie nichts anderes angeben, nehmen `div`-Container stets die gesamte Breite der Webseite oder des übergeordneten `div`-Containers ein. Dieses übergeordnete Element wird auch als **Eltern-Element** bezeichnet werden.

Damit Sie einen `div`-Container frei auf der Seite positionieren oder schweben lassen können (vgl. Abschnitt 2.4 und 2.5), müssen Sie per CSS auch seine Breite definieren.

`width``:Wert`	Mit der CSS-Eigenschaft `width` können Sie die Breite des Elements festlegen. Für die Angabe des Werts haben Sie folgende Möglichkeiten: ✔ Zahlenangaben mit entsprechender Maßeinheit, ✔ prozentuale Angaben relativ zur Breite des Eltern-Elements, ✔ `auto` = automatische Festlegung.
`height``: Wert`	Neben der Breite eines Elements können Sie auch dessen Höhe bestimmen. Die Wertangabe entspricht der für die Eigenschaft `width`.

2.4 `div`-Container schweben lassen

Der normale Elementfluss sieht vor, dass `div`-Container in der Reihenfolge, wie sie im Quelltext auftreten, aufeinanderfolgen. Das heißt, die Absätze werden untereinander dargestellt. Mit der CSS-Eigenschaft `float` können Sie `div`-Container jedoch von Text umfließen lassen.

`float``: Wert`	Die folgenden Werte können Sie für `float` angeben: ✔ `left`: Element steht links und wird rechts umflossen. ✔ `right`: Element steht rechts und wird links umflossen. ✔ `none`: Element wird nicht umflossen. Die Eigenschaft `float` lässt sich nicht auf absolut positionierte Elemente (vgl. Abschnitt 2.6) anwenden.

Damit die Eigenschaft `float` eine sichtbare Auswirkung hat, müssen Sie für den `div`-Container eine Breite festlegen (z. B. 50 %). Nur wenn neben dem Element Platz ist, können daneben weitere Inhalte stehen.

Wie die Abbildung zeigt, befindet sich das umfließende Element (der durch das Tag `p` definierte Absatz) nicht neben dem Float, sondern dahinter. Lediglich die Inhalte des Absatzes werden verschoben: Der Float gibt den ihm zustehenden Platz komplett frei, die nachfolgenden Elemente rutschen entsprechend weit nach oben, die darin enthaltenen Texte und Bilder werden in den sichtbaren Bereich gedrückt.