

## Kapitel 14



# CSS-Layouts und Frameworks

*Worin Sie erfahren, was CSS-Frameworks überhaupt sind und warum sie bei der Erstellung mehrspaltiger Layouts sehr praktisch sein können.*

Bevor Sie in den folgenden Kapiteln erfahren, wie man mit dem CSS-Framework YAML ein mehrspaltiges Layout erstellt, möchte ich in diesem Kapitel zwei Dinge kurz erläutern:

- warum es auf Webseiten keine Spalten und somit eigentlich auch kein mehrspaltiges Layout gibt
- was ein CSS-Framework ist und warum es die tägliche Arbeit erleichtern kann

### 14.1 Mehrspaltigkeit ist eine Illusion

Laut Wikipedia bezeichnet eine Illusion im allgemeinen Sprachgebrauch »eine mit technischen Mitteln herbeigeführte Sinnestäuschung«. Es gibt keine Spalten, also lassen Sie uns welche bauen ...

#### Das »Nebeneinander« auf Webseiten

Statische Block-Boxen können nicht nebeneinanderstehen, denn sie haben einen integrierten Zeilenumbruch. Nebeneinander, das können nur Inline-Boxen. Für eine einfache horizontale Navigation ist



`display: inline` durchaus geeignet und wegen der geringen Risiken und Nebenwirkungen besonders für Einsteiger zu empfehlen, aber man kann damit definitiv keine mehrspaltigen Layouts erstellen.

Für das Nebeneinander von Block-Boxen kennt CSS nur zwei Möglichkeiten, die beide unterschiedliche Vor- und Nachteile haben:

1. **Absolute Positionierung.** Absolut positionierte Elemente wissen nichts vom Flow. HTML-Elemente auf Webseiten sind von Natur aus sehr rücksichtsvoll und gehen sich gegenseitig aus dem Weg, sodass zwei im Flow befindliche Elemente sich niemals überlappen. Für ein absolut positioniertes Element ist hingegen der Autor verantwortlich, nicht der Flow.
2. **Float.** Gefloatete Elemente liegen zwar auch nicht im Flow, sehen sich aber wenigstens ein bisschen. Beim Layouten kommt ein Float selten allein und verschachtelte Floats führen oft zu unliebsamen Überraschungen. Besonders in Browsern, die zu einer Zeit programmiert wurden, als man `float` für eine eher selten genutzte Eigenschaft zum Umfließen von Grafiken hielt.

Fazit: Immer wenn auf einer Webseite Block-Boxen nebeneinanderstehen, wurde das entweder mit absoluter Positionierung oder `float` erreicht. Was anderes gibt es nicht.

### Tipp **Röntgenblick mit der *Web Developer Toolbar***

Im Menü **HERVORHEBEN** können Sie mit dem Befehl **POSITIONIERTE ELEMENTE HERVORHEBEN ...** ohne langwierige Quelltextanalyse herausfinden, welche Methode zur Positionierung eines Elementes verwendet wurde.

### **In CSS gibt es keine gleich langen Spalten**

Visuelle Designer benutzen zum Entwerfen von Layouts gern Gestaltungsraster: Waagerechte und senkrechte Linien, die den zu gestaltenden Raum in Spalten und Zeilen aufteilen. Grids. Tabellen. Aus Designersicht ist das Fehlen eines solchen Rasters eine der großen Schwächen von CSS. Spalten in einem Gestaltungsraster sind zum Beispiel problemlos gleich lang, und diese Erwartungshaltung überträgt sich auch auf CSS-basierte Designs.

Mit CSS kann man zwar Boxen nebeneinanderstellen, aber es gibt keine Spalten mit gleicher Länge. Es ist zwar möglich, Flächen zu erstellen, die optisch wie eine Spalte wirken, aber eigentlich sind mehrspaltige CSS-Layouts eine Illusion:



Abbildung 14.1:  
Die »Spalte« geht nicht wirklich bis ganz unten

Um bei einem CSS-Layout zumindest scheinbar gleich lange Spalten zu ermöglichen, haben Webdesigner Tricks wie »Faux Columns« oder »Companion Columns« erfunden, aber es bleiben Tricks und sie sind zum Teil nur mit einem sehr hohen Aufwand zu realisieren.

Vor dem Erstellen mehrspaltiger CSS-Layouts sollte man also zunächst die grundlegenden Techniken wie `position: absolute` und `float` verstehen, denn wer glaubwürdige Illusionen erzeugen will, muss sein Handwerk beherrschen. Das weiß jeder Magier. Die zauberhafte Assistentin wird nicht wirklich zersägt. Alles Illusion. Genau wie Spalten in CSS-Layouts.

### Tricks zur Erstellung gleich langer Spalten

Die folgenden Artikel erklären zwei Techniken zum Erzielen von Spalten mit gleicher Länge:

- *FauxColumns*: <http://alistapart.com/articles/fauxcolumns/>
- *Companion Columns*: <http://satzansatz.de/cssd/companions.html>

### Tipp

## Gestaltungsraster in CSS sind Zukunftsmusik

In den 90ern wurden zur Simulation von Gestaltungsrastern auf Webseiten HTML-Tabellen eingesetzt, in diesem Jahrtausend absolute Positionierung und `float`, aber keine dieser Techniken ist in erster Linie zum Erstellen von mehrspaltigen Layouts erfunden worden:

- HTML-Tabellen sollen tabellarische Daten darstellen.
- Absolute Positionierung nimmt Elemente aus dem Flow.
- `float` lässt Text um eine Grafik fließen.

Von Gestaltungsrastern oder mehrspaltigen Layouts hat keiner was gesagt. So, wie die Layout-Tabellen der 90er-Jahre heute als veraltet gelten, werden zukünftige Webdesigner wahrscheinlich eines Tages auf heutige CSS-Layout-Techniken zurückblicken und dabei leicht kopfschüttelnd vor sich hin murmeln: »Nee, wat'n Klamdudel ...«

Trotzdem sind CSS-Layouts gegenüber Tabellen-Layouts ein Schritt in die richtige Richtung, denn sie ermöglichen immerhin eine weitgehende Trennung von Inhalt und Gestaltung. Das war die Geschichte mit Shrek: Oger bestehen aus Schichten, Zwiebeln bestehen aus Schichten, Webseiten auch (»Little Boxes«, Band 1, Seite 30).

Aber es besteht durchaus Hoffnung, denn in CSS3 sind gleich mehrere Möglichkeiten zur Gestaltung mit Rastern vorgesehen. Und während das W3C früher Standards wie HTML oder CSS in einem akademischen Elfenbeinturm zu entwickeln schien und Webdesigner in der weiten Welt anschließend fehlende Praxistauglichkeit bemängelten, scheinen heute beide Seiten zu begreifen, dass sie einander brauchen.

So haben sich elf weltbekannte Webentwickler unter dem Namen »CSS Eleven« zusammengetan, um der CSS Working Group beim W3C dabei zu helfen, dass CSS3 real existierende Designprobleme löst und kein Papiertiger wird:

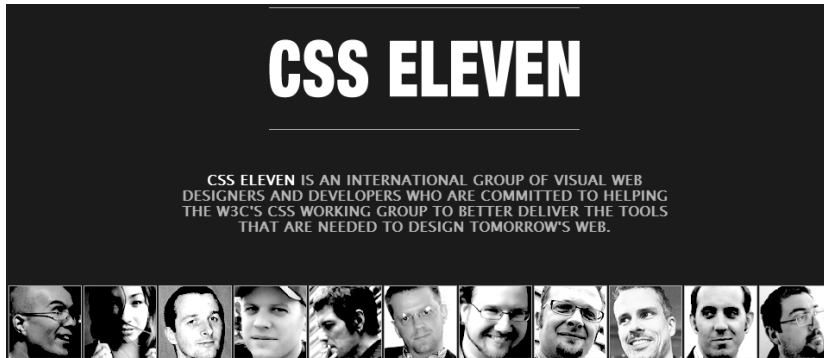


Abbildung 14.2:  
csseleven.com –  
elf visuelle Web-  
designer unter-  
stützen das W3C  
bei CSS3

Bis CSS3 Realität wird, kann es aber noch ein paar Jahre dauern. Wer also *heute* mehrspaltige Layouts erstellen möchte, muss weiterhin entweder absolut positionieren oder einen Float-Zirkus bändigen.

### Gestaltungsraster in CSS3

Zur Erstellung von mehrspaltigen Layouts und Gestaltungsrastern gibt es in CSS3 gleich mehrere Ansätze:

- CSS3 module: Multi-column layout (Editor Hakon Wium Lie)  
<http://w3.org/TR/2007/WD-css3-multicol-20070606/>
- CSS Advanced Layout Module (Editor Bert Bos)  
<http://w3.org/TR/2007/WD-css3-layout-20070809/>
- CSS Grid Positioning Module Level 3  
<http://w3.org/TR/css3-grid/>
- Die CSS-Roadmap des W3C gibt einen Überblick über die Fortschritte:  
<http://www.w3.org/Style/CSS/current-work#table>

### Tipp

## 14.2 Frameworks für CSS

Da es eigentlich keine gleich langen Spalten gibt, müssen Webdesigner jeden Tag viele Klippen umschiffen, um mehrspaltige Layouts zu erzeugen. Viele malen dazu eigene Landkarten, andere benutzen vorgefertigte.

### Frameworks sind wie Waschmaschinen

Jeder Profi hat im Laufe der Zeit ein Repertoire von Techniken, Tricks und Templates entwickelt, das als Fundament für neue Projekte dient und den Rahmen für seine tägliche Arbeit bildet. Den Rahmen für die Arbeit oder, auf Englisch, das »*Frame-Work*«.

Die meisten Entwickler benutzen ein solches Fundament, einen Maßanzug, der für die eigenen Bedürfnisse geschneidert wurde und nicht veröffentlicht wird, weil er anderen Leuten sowieso nicht passt.

Der Designer Jeff Croft hat bei »A List Apart« einen Artikel mit dem Titel »Frameworks for Designers« geschrieben. Er beschreibt ein *Framework* (frei übersetzt) als ...

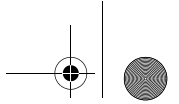
*»... eine Sammlung von Werkzeugen, Bibliotheken, Konventionen und bewährten Praktiken, die regelmäßig anfallende Aufgaben in allgemeine, wiederverwendbare Module zu abstrahieren versucht.«*

*<http://alistapart.com/articles/frameworksfordesigners>*

Frameworks sollen also die tägliche Arbeit erleichtern, sodass man nicht mehr die Hälfte seiner Zeit damit zubringt, aus jeder Webseite immer gleiche Patch-Flecken und Browserbugs rauszuwaschen. Frameworks sind also wie Waschmaschinen.

Ein Beispiel dafür ist die Kalibrierung der unterschiedlichen Abstände aus den Browser-Stylesheets. Das ab Seite 50 vorgestellte CSS-Fundament vereinheitlicht die Abstände für häufig benutzte Elemente, auch wenn sie auf der Seite (noch) gar nicht vorkommen. Zusätzlich erleichtert die Verteilung der Styles auf verschiedene Dateien die Wiederverwendbarkeit in verschiedenen Projekten. Eine solche Herangehensweise ist bereits der erste Schritt in Richtung Framework.

Ein anderes Beispiel sind mehrspaltige Layouts: Das klassische Webseitenlayout hat zwei bis drei Spalten mit einer durchgehenden Kopf- und Fußzeile. Zur Umsetzung eines solchen Layouts gibt es zahl-



reiche Methoden und nicht nur für Einsteiger ist die Vielzahl der Möglichkeiten eher verwirrend, zumal die Nachteile oft erst viel später in Form gestalterischer Einschränkungen oder nerviger Browserbugs ersichtlich werden. Beides hat schon so manchen Webautor an den Rand der Verzweiflung und darüber hinaus getrieben.

Ein gutes CSS-Framework schnürt also ein Gesamtpaket zur Erstellung mehrspaltiger Layouts und bietet so ein solides Fundament für verschiedenste Layoutwünsche.

### CSS-Frameworks: Vor- und Nachteile

Ein CSS-Framework bietet dem Benutzer einige Vorteile:

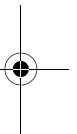
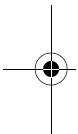
- Ein solides und vielfach bewährtes Fundament spart Zeit bei der Umsetzung neuer Projekte.
- Die Pflege des Quelltextes wird durch konsistente Konventionen bei der Namensgebung für IDs und Klassen erleichtert. Gerade bei der Arbeit im Team ist das ein wichtiger Faktor.
- Patchwork und Browserfehler kosten weniger Nerven, denn die Bug-Prophylaxe ist Teil des Frameworks.

Klingt verlockend? Aber Frameworks haben auch Nachteile:

- Die Einarbeitung in das Framework kostet Zeit. Sie müssen sich vor dem Gebrauch mit der Struktur des Frameworks vertraut machen und lernen, wie man damit zum Ziel kommt.
- Sie übernehmen die Konventionen des Framework-Entwicklers und damit ungefragt dessen persönliche Vorlieben, zum Beispiel bei Namensgebung und Syntax.
- Frameworks erzeugen mehr Code als notwendig, weil sie für viele Situationen optimiert wurden und nicht für eine. Je komplexer das Layout und je mehr Inhalte auf einer Site stehen, desto weniger fällt dieser Overhead allerdings ins Gewicht.

Die Arbeit mit einem Framework ist also ein Kompromiss. Genau wie mehrspaltige CSS-Layouts. Vielleicht passen die beiden darum so gut zusammen. Man muss halt nur zu leben wissen, mit Kompromissen.

Zur Erstellung einer kleinen persönlichen Homepage mit vier oder fünf Seiten wäre die vorherige Einarbeitung in ein Framework über-



trieben. Das ist, als ob Sie erst Architektur studieren, um dann von Hand ein Gartenhäuschen zu bauen, das es im Baumarkt als Bausatz gibt.

Je öfter Sie aber Webseiten mit mehrspaltigen Layouts erstellen, desto eher lohnt sich die Beschäftigung mit einem CSS-Framework.

## Tipp

### Vor- und Nachteile von Frameworks

Die Listenmacher vom »Smashing Magazine« haben den Stand der Diskussion in einem lesenswerten Artikel zusammengefasst:

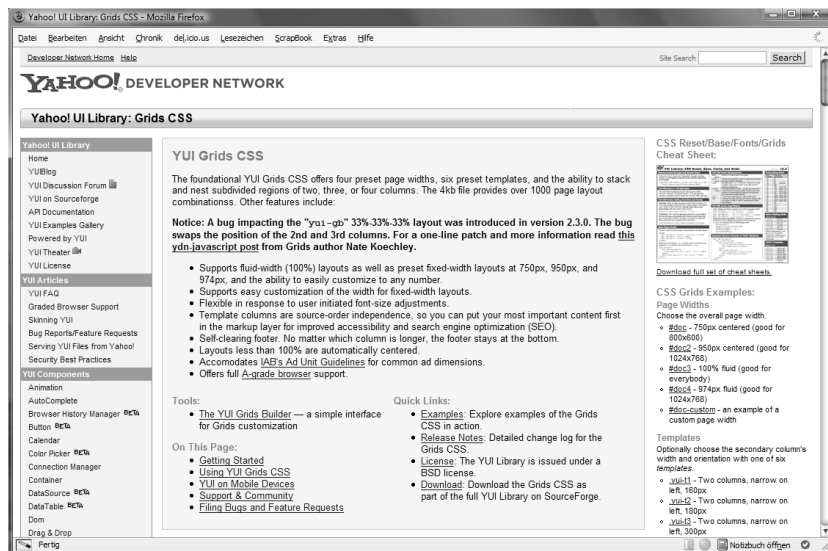
- <http://smashingmagazine.com/2007/09/21/css-frameworks-css-reset-design-from-scratch/>

## CSS-Frameworks – ein kurzer Überblick

In diesem Abschnitt möchte ich Ihnen drei der momentan bekanntesten CSS-Frameworks kurz vorstellen:

- *YUI Grids CSS*: <http://developer.yahoo.com/yui/grids/>

Abbildung 14.3:  
Die Startseite von  
Yahoo Grids CSS



*YUI Grids* (Grids heißt »Raster«) ist auf Layouts mit fester Breite spezialisiert und auf die Zusammenarbeit mit den anderen Komponenten des Yahoo-UI ausgelegt. English only.

■ *Blueprint: <http://code.google.com/p/blueprintcss/>*

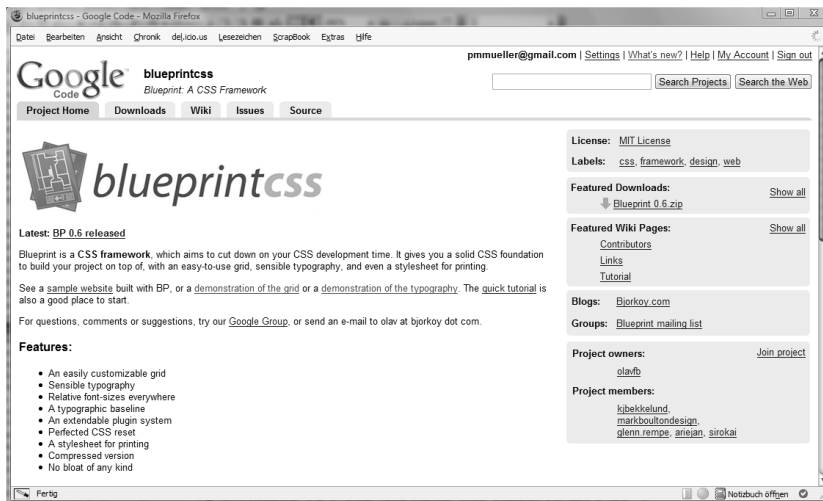


Abbildung 14.4:  
Die Startseite  
von BlueprintCSS

*Blueprint* basiert auf einem pixelbasierten Raster-Baukastensystem, das mit typographischen Gestaltungsvorgaben kombiniert wird. English only.

■ *YAML (»Yet Another Multicolumn Layout«): <http://yaml.de/>*

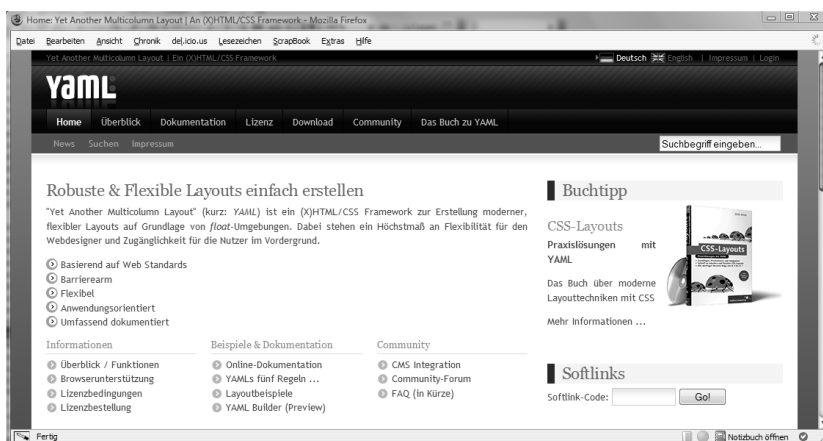


Abbildung 14.5:  
Die Startseite  
von YAML

YAML wurde am Ende des ersten Bandes bereits kurz vorgestellt (S. 290):

*Angeregt von Webkrauts-Gründer Jens Grochtdreis hat Dirk Jesse aus Dresden im Herbst 2005 begonnen, ein flexibles Layout-System zu kon-*

zipieren, mit dem man recht einfach alle möglichen verschiedenen Layouts realisieren kann, ohne bei jedem Projekt wieder bei null anfangen zu müssen.

YAML gibt ein Basislayout vor und kümmert sich um die Bugs, lässt dem Autor aber trotzdem alle Freiheiten.

Und YAML hat noch einen Vorteil, den man nicht unterschätzen darf: Es gibt keine Geheimnisse. Auf der Site finden Sie eine ausführliche Dokumentation (deutsch *und* englisch), die genau beschreibt, wie YAML funktioniert und welche Techniken zu welchem Zweck verwendet werden. Open Source. Wem das nicht reicht, der kann YAML unter dem Titel »CSS-Layouts« auch zwischen zwei Buchdeckeln erwerben, vom Meister selbst geschrieben.

### 14.3 Auf einen Blick

Hier noch einmal die wichtigsten Punkte dieses Kapitels im Überblick:

- Statische Block-Boxen auf Webseiten können nicht nebeneinanderstehen.
- Block-Boxen nebeneinanderzustellen geht nur mit `float` oder absoluter Positionierung.
- In CSS-Layouts gibt es keinen Spaltensatz. Es gibt nur Flächen, die optisch wie eine Spalte wirken.
- Gestaltungsraster in CSS werden erst in CSS3 verfügbar sein. Zukunftsmusik.
- Webdesigner entwickeln im Laufe der Zeit ein persönliches Repertoire von Techniken, Tricks und Templates, das den Rahmen für die tägliche Arbeit bildet.
- Frameworks erleichtern immer wiederkehrende Routinarbeiten, kosten aber Zeit bei der Einarbeitung.
- Die bekanntesten CSS-Frameworks sind derzeit YUI Grids, Blueprint und YAML.
- YAML bietet eine sehr flexible und gut dokumentierte Grundlage für mehrspaltige Layouts.