

Dave Shea
Molly E. Holzschlag

Zen und die Kunst des CSS-Designs

Inspiration und
Umsetzung



ADDISON-WESLEY

Ein Imprint von Pearson Education

München • Boston • San Francisco • Harlow, England
Don Mills, Ontario • Sydney • Mexico City
Madrid • Amsterdam



White Lily, Seite 120



Not So Minimal, Seite 144

Layout

DIE SEITENGESTALTUNG mit Hilfe von CSS erfordert eine Kombination aus technischem Wissen und Kreativität. Unabhängig davon, ob Sie ein festes oder ein fließendes Layout wählen und ob Sie Ihre Designs linksbündig oder zentriert ausrichten, müssen Sie eine feste Vorstellung von der Funktionsweise der CSS-Layouts haben.

In diesem Kapitel behandeln wir Layoutansätze, CSS-Positionierungsschemata, die Verwendung von Floats für das Layout und die Kombination dieser Verfahren zum Erstellen von nützlichen, attraktiven und – in einigen Fällen – überraschenden Layouts.

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.

Download the sample [html file](#) and [css file](#)

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WotP and the major browser creators.

The zen Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and intriguing fashion. Become one with the web.

SO WHAT IS THIS ABOUT?

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. To date, most examples of neat tricks and hacks have been demonstrated by annotations and coders. Designers have yet to make their mark. This needs to change.

PARTICIPATION

Graphic artists only please. You are modifying this page, so strong CSS skills are necessary, but the example files are commented well enough that even CSS novices can use them as starting points. Please see the [CSS Resource Guide](#) for advanced tutorials and tips on working with CSS.

You may modify the style sheet in any way you wish, but not the **HTML**. This may seem daunting at first if you've never worked this way before, but follow the listed links to learn more, and use the sample files as a guide.

Download the sample [html file](#) and [css file](#) to work on a copy locally. Once you have completed your masterpiece (and please, don't submit half-finished work) upload your .css file to a web server under your control. [Send us a link](#) to the file and if we choose to use it, we will spider the associated images. Final submissions will be placed on our server.

BENEFITS

Why participate? For recognition, inspiration, and a resource we can all refer to when making the case for CSS-based design. This is sorely needed, even today. More and more major sites are taking the leap, but not enough have. One day this gallery will be a historical curiosity; that day is not today.

REQUIREMENTS

We would like to see as much CSS as possible. CSS2 should be limited to widely-supported elements only. The zen Zen Garden is about functional, practical CSS and not the latest bleeding-edge tricks viewable by 2% of the browsing public. The only real requirement we have is that your CSS validates.

Unfortunately, designing this way highlights the flaws in the various implementations of CSS. Different browsers display differently, even completely valid CSS at times, and this becomes maddening when a fix for one leads to breakage in another. View the [Resources](#) page for information on some of the fixes available. Full browser compliance is still sometimes a pipe dream, and we do not expect you to come up with pixel-perfect code across every platform. But do test in as many as you can. If your design doesn't work in at least IE6+/Win and Mozilla (run by over 90% of the population), chances are we won't accept it.

We ask that you submit original artwork. Please respect copyright laws. Please keep objectionable material to a minimum; tasteful nudity is acceptable, outright pornography will be rejected.

This is a learning exercise as well as a demonstration. You retain full copyright on your graphics (with limited exceptions, see [submission guidelines](#)), but we ask you release your CSS under a Creative Commons license identical to the [one on this site](#) so that others may learn from your work.

Bandwidth graciously donated by [DreamFire Studios](#).

SELECT A DESIGN

- [contemporary.novena](#) by David Novena
- [El Collar de Tomas](#) by Maria Sotir
- [Order&Zen](#) by Steve Smith
- [Bonsai](#) by Martin Plozdra
- [Type.Think](#) by Michal Makrycki
- [Pseudo Sahara](#) by John Barwick
- [Dreazen](#) by Matthew Buchanan
- [Vivacity](#) by Sozo

ARCHIVES

- [next designs »](#)
- [View All Designs](#)

RESOURCES

- [View This Designer's CSS](#)
- [CSS Resources](#)
- [FAQ](#)
- [Submit A Design](#)
- [Translations](#)

Ray Henry, Designer

www.csszengarden.com/029

Backyard

Absolute Positionierung und Floats verstehen

MANCHMAL IST ES SCHWER, an einen Aufhänger für ein neues Design zu gelangen, aber Inspirationen finden sich überall. In einem Café im Park zu sitzen, den Nachmittag bei einer großen Tasse Café Latte zu verträdeln und eine anregende Unterhaltung zu führen, hat schon zu vielen revolutionären Ideen geführt. Beim Aufbau von **Backyard** wurde **Ray Henry** in zwei Gärten von der Muse geküsst – dem Zen-Garten und seinem eigenen. Die Farben von Backyard stammen natürlich von dem lebendigen Foto einer Blume aus Henrys Garten.

So einfach ein dreispaltiges Layout auch sein mag, für Ray Henry stellten sich beim Kodieren grundlegende Fragen, zum Beispiel, ob er die vielen auf Tabellen beruhenden Layouteffekte, die ihm bereits vertraut waren, mit Hilfe von CSS nachbauen könnte. Könnte er die Spalte mit den Links auf die rechte Seite setzen? Wenn ja, wie? Musste er sie anders gestalten, damit sie funktionierte? Dabei lernte Henry durch Ausprobieren und löste ein Layoutproblem nach dem anderen.

Layoutgrundlagen

Spaltenlayouts in CSS erfordern einige Vorkenntnisse. Die Eigenschaft `left-column: 200px`, mit der Sie wie von Zauberhand Spalten erstellen, existiert nicht, so dass verschiedene Tricks notwendig sind, um diese Einschränkungen zu umgehen.

Es gibt zwei gebräuchliche Methoden zum Erstellen von Spaltenlayouts, beide haben Vor- und Nachteile. Von den vier CSS-Positionierungsschemata (absolut, statisch, relativ und fest) stellt die absolute Positionierung das Verfahren dar, mit dem Sie Inhalte aus dem regulären Dokumentfluss entfernen und an beliebiger Stelle wieder einsetzen können. Obwohl es sich im technischen Sinne nicht um ein Positionierungsschema handelt, können CSS-Floats andererseits Inhalten einen Platz zuweisen und den Rest des Dokuments fließend um diese Inhalte anordnen.

Die absolute Positionierung oder Floats erlauben die notwendige Steuerung der Elemente zum Erstellen eines mehrspaltigen Layouts. Sowohl Floats als auch die absolute Positionierung können einzeln oder gemeinsam eingesetzt werden.

Hinweis

Obwohl es im Grunde richtig ist, dass CSS-Spaltenlayouts nicht explizit zur Verfügung stehen, gestatten CSS-Tabellen genau diese Art von Layout. Sie wurden zwar in CSS2 eingeführt, werden allerdings vom Internet Explorer für Windows nicht unterstützt, so dass sie derzeit unbrauchbar sind. Schade. Weitere Informationen über CSS-Tabellen finden Sie unter www.w3.org/TR/CSS21/tables.html.

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

ABBILDUNG 1 Elemente innerhalb des normalen Dokumentflusses

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

The Road to Enlightenment

ABBILDUNG 2 Die absolute Positionierung entfernt die Kopfzeile aus dem Dokument und ermöglicht es dabei, dass der Absatz an ihre frühere Position rückt.

ABSOLUTE POSITIONIERUNG

Von diesen beiden Methoden ist die absolute Positionierung leichter zu verstehen. Wenn Sie wie im folgenden HTML-Code eine Reihe von Elementen nacheinander im Dokument platzieren (was als *normaler Dokumentfluss* bezeichnet wird), entsteht das von Ihnen erwartete, in **ABBILDUNG 1** dargestellte lineare Layout, in dem die Elemente nacheinander angeordnet sind.

```
<h1>This is a header</h1>
<p>This is some text</p>
<p>And even more text</p>
```

Bei Anwendung der absoluten Positionierung auf eines der Elemente wird dieses (in diesem Fall die Überschrift) jedoch an die von Ihnen genannte Stelle verschoben und die dadurch entstandene Lücke geschlossen (**ABBILDUNG 2**). Das absolut positionierte Element hat keinen Einfluss auf seine benachbarten Elemente und verhält sich beinahe so, als würde es im Dokument nicht existieren (obwohl Sie es deutlich sehen können). Die Syntax könnte in etwa wie folgt aussehen:

```
h3 {position: absolute; top: 100px; }
```

Der Vorteil der absoluten Positionierung besteht darin, dass die Position eines beliebigen Elements genau gesteuert werden kann; es gibt kein Rätselraten. Da das Element aus dem normalen Dokumentfluss vollständig entfernt wird, ist den anderen Elementen nicht bewusst, welchen Einfluss es ausübt. Dies erweist sich als Problem, was wir später in diesem Abschnitt noch zeigen werden.

FLOAT-OBJEKTE

Float-Objekte sind etwas schwieriger zu erkennen, so dass wir einige optische Hilfsmittel einsetzen. **ABBILDUNG 3** zeigt einen langen Textabschnitt; ein in diesen Text gesetztes Foto beansprucht naturgemäß einen bestimmten Platz und verschiebt damit den restlichen Text (**ABBILDUNG 4**). Um die Unterbrechung zwischen dem über und unter dem Foto liegenden Text zu vermeiden, muss der Text auf irgendeine Weise in den weißen Zwischenraum neben dem Foto übertragen werden, um diesen Bereich effektiver auszufüllen (**ABBILDUNG 5**).



ABBILDUNG 3 Regulärer Text

Das ist der Kern dessen, was CSS-Floats bieten: Das Foto wird in den Textkörper eingefügt und vom Text umflossen, damit keine hässliche Lücke bestehen bleibt.

Die Syntax ist einfach, ihre Beherrschung jedoch nicht. Der CSS-Codeabschnitt zur Beschreibung dieses Szenarios könnte in etwa wie folgt aussehen:

```
#photo {float: left; margin: 10px 10px 10px 0;}
```

Für Floats ist dies so grundlegend wie möglich und in hohem Maße vorhersehbar.

Der Vorteil floatbasierter Layouts besteht darin, dass ein Float-Element nicht wirklich aus dem Dokumentfluss entfernt wird; die benachbarten Elemente kennen seine Position. Wegen der etwas uneinheitlichen Browser-Unterstützung sind Float-Objekte allerdings nicht immer die ideale Lösung für das Layout und es erfordert einige praktische Kenntnisse, um sich für ein Layoutverfahren entscheiden zu können.

Layoutverfahren

Das Markup auf der gegenüberliegenden Seite umfasst die grundlegenden Bausteine für drei Spalten und eine Fußzeile. Für Suchmaschinen und Benutzer, deren Browser kein CSS unterstützen, wird der Haupttextkörper im Quellcode vor den beiden seitlichen Themenkästen angeordnet; je früher der Text im Quellcode erscheint, desto geringer ist die Anzahl der Navigations- und Linkobjekte, die übersprungen werden müssen, bevor der Inhalt der Seite angezeigt wird.

```
<div id="wrapper">
  <div id="contentArea">
    <p>(headers and text here)</p>
  </div>
  <div id="leftPanel">
    <p>(links and various text items here)</p>
  </div>
  <div id="rightPanel">
    <p>(links and various text items here)</p>
  </div>
  <div id="footer">
    <p>(Copyright statement, links, etc.)</p>
  </div>
</div>
```



ABBILDUNG 4 Wenn ein Foto hinzugefügt wird, erscheint eine Lücke im Text.



ABBILDUNG 5 Der Text wird fließend um das Foto angeordnet, um die Lücke zu schließen.

Tip

Merkwürdigerweise verdoppelt der Internet Explorer für Windows die horizontalen Ränder einiger Float-Elemente, was zu Verwunderung und Frustration führen kann. Wenn Ihre Werte für die Ränder auffälliger ausfallen als etwa 10 Pixel, wenden Sie zur Umgehung dieses Problems `display: inline` auf die Float-Elemente an. Weitere Informationen über diesen Fehler finden Sie bei [Position is Everything \(www.positioniseverything.net/explorer/doubled-margin.html\)](http://Position is Everything (www.positioniseverything.net/explorer/doubled-margin.html)).



ABBILDUNG 6 Einfaches dreispaltiges Layout



ABBILDUNG 7 Dasselbe Layout in zentrierter Form

Ein auf die Schnelle erstelltes Layout unter Einsatz der absoluten Positionierung erfordert kaum Planung: Sie weisen allen drei Spalten eine feste Breite zu, beginnen in der oberen linken Ecke des Dokuments mit der Positionierung der einzelnen Spalten, ignorieren die Fußzeile erst einmal und schon sind Sie fertig (**ABBILDUNG 6**).

```
#contentArea {width: 450px; position: absolute; top: 0;
  left: 150px;}
#leftPanel {width: 150px; position: absolute; top: 0;
  left: 0;}
#rightPanel {width: 150px; position: absolute; top: 0;
  left: 600px;}
#footer {display: none;}
```

Das Layout ist linksbündig ausgerichtet, aber was geschieht, wenn Sie es, wie in **ABBILDUNG 7** dargestellt, zentrieren wollen? Nichts leichter als das: Wenn Sie Elemente absolut in bereits positionierten Elementen einfügen wollen, errechnet die Funktion die Werte für die Position. Dabei berücksichtigt sie, dass top und left von der oberen linken Ecke des Containerelements berechnet werden müssen und nicht vom html-Stammelement (also der oberen linken Ecke des Browserfensters).

Klingt das etwas verwirrend? Das Containerelement fungiert als neuer Ausgangspunkt für die absolute Positionierung aller darin befindlichen Elemente. Wird #wrapper zentriert und positioniert, verbleiben daher die darin enthaltenen Spalten innerhalb dieses Elements. Der unten angegebene Stil zeigt uns, wie alles miteinander verbunden wird (die zweimalige Verwendung der Eigenschaft text-align dient zur Umgehung eines Fehlers im Internet Explorer 5 für Windows, der das Zentrieren eines Elements nicht erlaubt, wenn der Wert für die Ränder auf auto gesetzt wird).

```
body {
  text-align: center;
}
#wrapper {
  width: 750px;
  margin: 0 auto;
  text-align: left;
  position: relative;
}
#contentArea {
  width: 450px;
```

```

position: absolute;
top: 0;
left: 150px;
}
#leftPanel {
width: 150px;
position: absolute;
top: 0;
left: 0;
}
#rightPanel {
width: 150px;
position: absolute;
top: 0;
left: 600px;
}

```

Was geschieht aber, wenn die Fußzeile benötigt wird? Da die absolut positionierten Spalten die umgebenden Elemente nicht beeinflussen, werden sie von der Fußzeile ignoriert, so dass diese dort positioniert wird, wo sie normalerweise platziert würde, wenn diese Elemente nicht vorhanden wären, d.h. in diesem Fall als erstes Element auf der Seite.

Um dies unter Verwendung der absoluten Positionierung zu umgehen, muss die Höhe der einzelnen Spalten vorab bekannt sein und die Position der Fußzeile jeweils unterhalb des unteren Rands liegen. Ohne auf JavaScript zurückzugreifen, können Sie das nicht erreichen, wenn die Textlänge einer der Spalten unbekannt ist, es sei denn, Sie wechseln von der absoluten Positionierung zum Einsatz von Floats.

In **ABBILDUNG 8** gleicht die Fußzeile dank der Verwendung von Floats die drei Spalten aus und positioniert sich selbst darunter.

```

body {
text-align: center;
}
#wrapper {
width: 750px;
margin: 0 auto;
text-align: left;
position: relative;
}

```



ABBILDUNG 8 Dasselbe Layout mit Floats und einer Fußzeile

Tipp

Weitere Informationen über eine Lösung mit negativen Rändern finden Sie in der Demo von Ryan Brill (www.alistapart.com/articles/negativemargins).

```
#contentArea {  
  width: 450px;  
  float: left;  
}  
#leftPanel {  
  width: 150px;  
  float: left;  
}  
#rightPanel {  
  width: 150px;  
  float: left;  
}  
#footer {  
  clear: both;  
}
```

Ein offensichtliches Problem besteht darin, dass die linke und die mittlere Spalte nun ihre Plätze getauscht haben. Da Floats innerhalb des regulären Dokumentflusses bleiben, ist die Reihenfolge innerhalb des HTML-Quellcodes viel wichtiger als bei der absoluten Positionierung. Der einfachste Weg zur Lösung dieses Problems besteht darin, die Reihenfolge der linken und der mittleren Spalten im HTML-Code zu vertauschen. Es gibt kompliziertere Maßnahmen zur Lösung dieses Problems, die die Verwendung von negativen Rändern und eine sehr unorthodoxe Denkweise umfassen; begnügen wir uns damit, dass das Ändern der Reihenfolge im Quellcode in neun von zehn Fällen Abhilfe schafft.

Komplementäre Lösungen

Wie viele andere, die sich in CSS-Layouts einarbeiten, verwendete Ray Henry schließlich die absolute Positionierung für die Fertigstellung seines dreispaltigen Backyard-Layouts. Das ist in vielen Fällen (und sicherlich auch bei Backyard) eine gute Entscheidung, denn die absolute Positionierung ist leicht zu erlernen und wirkungsvoll in der Anwendung.

Dennoch ist sie nicht für alle Szenarios die perfekte Lösung und es können Probleme bei der Anwendung auftreten. In Fällen, in denen entweder eine Fußzeile erforderlich ist oder die Seitenelemente den ihnen benachbarten Elementen mehr Beachtung schenken müssen, dienen Floats dazu, die Schwachstellen aufzufangen. Kein Verfahren allein ist perfekt, aber zusammen bieten sie einen praktischen Satz komplementärer Layoutwerkzeuge. Verwenden Sie eine oder beide Methoden; die Möglichkeiten, die Ihnen eine Kombination der beiden Werkzeuge bietet, werden nur durch Ihre Vorstellungsgabe begrenzt.

css Zen Garden
the beauty of CSS design



(select a design:)
CONTEMPORARY NOUVEAU
by David Helling
EL OOLLAR DE TOMAS
by Maria Shultz
ORDEDEEEN
by Steve Smith
BONZAI
by Martin Flaxotta
TYPE THING
by Michal Mohrzycski
PSEUDO SAHARA
by John Barnick
DRAGON
by Matthew Buchanan
VINACITY
by Siso



(archives:)
[next designs >](#)
[View All Designs](#)



(resources:)
[View This Design's CSS](#)
[CSS Resources](#)
[FAQ](#)
[Submit a Design](#)
[Promotions](#)



A DEMONSTRATION OF WHAT CAN BE ACCOMPLISHED visually through CSS-based design. Select any style sheet from the list to load it into this page.
 Download the sample [html file](#) and [css file](#)

THE ROAD TO ENLIGHTENMENT
 Littering a dark and dreary road by the faint relics of browser-specific tags, incompatible DOM, and broken CSS support.
 Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WAP, and the major browser creators.
 The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the eye to lay time-honored techniques in new and invigorating fashion. Become one with the web.

SO WHAT IS THIS ABOUT?
 There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.
 CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. To date, most examples of neat tricks and hacks have been demonstrated by structuralists and coders. Designers have yet to make their mark. This needs to change.

PARTICIPATION
 Graphic artists only please. You are modifying this page, so strong CSS skills are necessary, but the example files are commented well enough that even CSS novices can use them as starting points. Please see the [css Resource Guide](#) for advanced tutorials and tips on working with CSS.
 You may modify the style sheet in any way you wish, but not the HTML. This may seem daunting at first if you've never worked this way before, but follow the listed links to learn more, and use the sample files as a guide.
 Download the sample [html file](#) and [css file](#) to work on a copy locally. Once you have completed your masterpiece (and please, don't submit half-finished work) upload your .css file to a web server under your control. [Send us a link](#) to the file and if we choose to use it, we will spider the associated images. Final submissions will be placed on our server.

BENEFITS
 Why participate? For recognition, inspiration, and a resource we can all refer to when making the case for CSS-based design. This is sorely needed, even today. More and more major sites are taking the leap, but not enough have. One day this gallery will be a historical curiosity; that day is not today.

REQUIREMENTS
 We would like to see as much CSS2 as possible. CSS2 should be limited to widely-supported elements only. The css Zen Garden is about functional, practical CSS and not the latest bleeding edge tricks viewable by 2% of the browsing public. The only real requirement we have is that your CSS validates.
 Unfortunately, designing this way highlights the flaws in the various implementations of CSS. Different browsers display differently, even completely valid CSS at times, and this becomes maddening when a fix for one leads to breakage in another. View the [Resources](#) page for information on some of the fixes available. Full browser compliance is still sometimes a pipe dream, and we do not expect you to come up with pixel-perfect code across every platform. But do test in as many as you can. If your design doesn't work in at least IE5+Win and Mozilla (run by over 50% of the population, chance as we won't accept it).
 We ask that you submit original artwork. Please respect copyright laws. Please keep objectionable material to a minimum; tasteful nudity is acceptable, outright pornography will be rejected.
 This is a learning exercise as well as a demonstration. You retain full copyright on your graphics (with limited exceptions, see [submission guidelines](#)), but we ask you release your CSS under a Creative Commons license identical to the one on [this site](#) so that others may learn from your work.

Bandwidth graciously donated by [DreamFire Studios](#). Coming soon: *Zen Garden, the book*.

{ xhtml } { css } { cc } { 508 } { aaa }

Jon Hicks, Designer

www.csszengarden.com/030

Entomology

Zentrierte Layouts im heutigen Webdesign

ES IST INTERESSANT, wo Ideen entstehen. Im Falle von **Entomology** stammt das Designkonzept aus dem Musikvideo der Single »Pin« von den Yeah Yeah Yeahs. **John Hicks** wollte einen optischen Aufhänger schaffen, der noch einen Bezug zum Gartenthema aufweisen sollte, ohne selbst einen Garten darzustellen. Nun, das Video von »Pin« zeigt präparierte Schmetterlinge ...

Nach dem Entwurf eines groben Layouts begann Hicks direkt mit der CSS-Programmierung, ohne zuvor ein Modell in Adobe Photoshop zu erstellen. Da er beruflich stark eingespannt war, hatte er wenig Zeit, den Standarddesignprozess zu durchlaufen. Außerdem behauptet er, dass er dabei einen Tropenhelm getragen und ein Schmetterlingsnetz griffbereit gehalten habe, weil es wichtig sei, die richtige Atmosphäre zu schaffen!

Festgelegte und fließende Layouts

Es gibt zwei herkömmliche Arten von Weblayouts: *fest* oder *fließend*. Ein *festes* Design weist eine bestimmte Breite auf. Feste Designs können zwar linksbündig, rechtsbündig (obwohl dies selten der Fall ist) oder zentriert ausgerichtet werden, ihr Inhalt geht jedoch nicht über die angegebene Breite hinaus. Entomology verfügt beispielsweise über ein festes Layout (**ABBILDUNG 1**).

Bei einem *fließenden* Design, das auch als dynamisches Design bezeichnet wird, handelt es sich um ein Design, bei dem von festgelegten Breitenangaben prozentuale Werte verwendet werden, so dass das Layout fließt und das gesamte zur Verfügung stehende Darstellungsfeld ausfüllt (**ABBILDUNG 2**).

Kombinationseffekte sind ebenfalls möglich. Stellen Sie sich ein dreispaltiges Layout vor, in dem die angrenzenden Spalten zwar auf eine bestimmte Breite gesetzt werden, für die mittlere Spalte jedoch ein prozentualer Wert (bzw. keine festgelegte Breite) verwendet wird. Dies führt dazu, dass das Design stets das gesamte Darstellungsfeld ausfüllt, die mittlere Spalte jedoch breiter oder schmaler wird, wenn sich die Größe des Browserfensters ändert oder die Seite bei unterschiedlichen Auflösungen betrachtet wird.

Wie bei vielen Dingen kommen und gehen die Trends und Vorlieben auch hier. In den Anfangszeiten des Web war es der letzte Schrei, feste Designs zu erstellen. Nach einer Weile entstanden Diskussionen über die Vorteile fließender Designs und für einige Jahre waren diese dann der letzte Schrei.

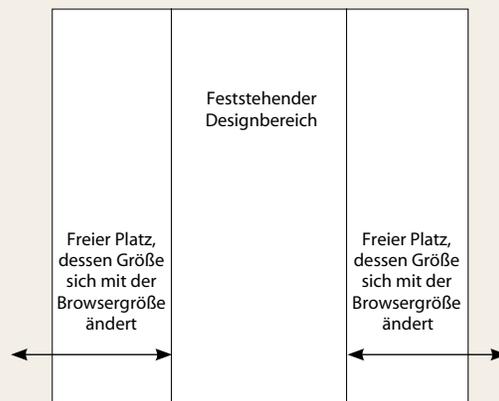


ABBILDUNG 1 Darstellung des Verhaltens bei fester Breite

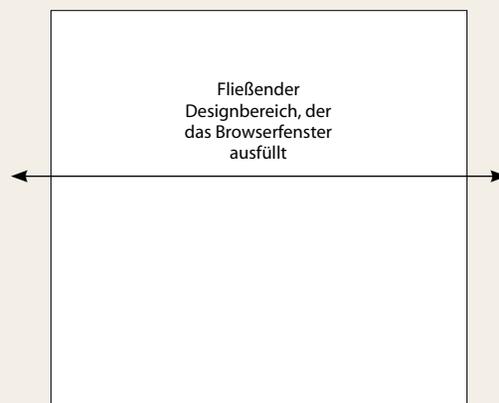


ABBILDUNG 2 Darstellung des Verhaltens bei einem fließenden Design

In den letzten zwei Jahren ist die Beliebtheit fester Designs wieder gestiegen, insbesondere unter den auf Standards zurückgreifenden Designern. Zum Zeitpunkt der Drucklegung dieses Buchs verwendeten nahezu alle führenden Fachleute feste Designs für ihre eigenen Sites und im beruflichen Bereich. Dies ist ein interessanter Trend und bringt die Gründe dafür ans Tageslicht, warum Designer möglicherweise ein festes anstelle eines fließenden Designs wählen.

Zuallererst gibt ein festes Design dem Designer bessere Steuerungsmöglichkeiten, da er über einen geschlossenen Arbeitsbereich verfügt. Die Länge der Textzeilen ist leichter zu steuern wie auch die festgelegte Platzierung und die Umgebung von Text und Bildern. Diese Gründe sind sehr wahrscheinlich die Ursache für das Wiederauftauchen des Trends zu festen Layouts.

Fließende Designs haben ebenfalls ihre Stärken. Das Ausfüllen des Darstellungsfelds im Browser bedeutet, dass der gesamte zur Verfügung stehende Platz genutzt und Probleme mit der optischen Ausgewogenheit vermieden werden, die entstehen, wenn ein Design linksbündig ausgerichtet und fest ist, so dass es nur einen Teil des verfügbaren Raums einnimmt. Die Kehrseite ist jedoch, dass die Steuerung der Zeilenlänge, des Textflusses und der Positionierung und der Umgebung von Seitenelementen entfällt. Für geschulte Designer stellt dies ein erhebliches Problem dar, für das nur sehr begrenzte Lösungsmöglichkeiten zur Verfügung stehen.

Ein von Designern verwendeter Ansatz, um die von einem festen Design bereitgestellten Steuerungsmöglichkeiten zu nutzen und Probleme mit der optischen Ausgewogenheit zu lösen, ist das horizontale Zentrieren ihrer festen Designs. Dies erlaubt das Hinzufügen von Leerraum um das Design, so dass es zur rechten und linken Seite gleichmäßig ausgerichtet werden kann, wenn sich die Größe des Browser-Fensters ändert, und somit das Erscheinen eines großen Nichts auf der rechten Seite verhindert wird.

Ab in die Mitte

Das Zentrieren in CSS stellt eine gewisse Herausforderung dar, da der korrekte Weg zum horizontalen Zentrieren eines Designs mit CSS nicht in allen Browsern konsistent funktioniert. Hicks stieß während seiner Arbeit an Entomology auf dieses Problem und beschloss, zur Lösung des Problems Abhilfemaßnahmen zu treffen. Bevor wir uns diesem Ansatz widmen, ist jedoch eine Prüfung der unterschiedlichen Möglichkeiten zum horizontalen Zentrieren CSS-basierter Layouts an der Reihe.

ZENTRIERTE GESTALTUNG MIT AUTOMATISCHEN RÄNDERN

Der bevorzugte Weg zum horizontalen Zentrieren eines beliebigen Elements ist, die Eigenschaft `margin` zu verwenden und die Werte für `left` und `right` auf `auto` zu setzen. Damit dies bei Layouts funktioniert, erstellen Sie einen `div`-Container. Für diesen Container *müssen* Sie eine Breite angeben:

```
div#container {  
    margin-left: auto;  
    margin-right: auto;  
    width: 168px;  
}
```

ABBILDUNG 3 zeigt ein Beispiel für diesen Effekt unter Verwendung eines Schmetterlingsbildes aus Hicks' Design. Das Bild wurde in ein `div`-Element platziert und mit Hilfe der automatischen Ränder zentriert.

Dieses Verfahren kann in nahezu allen aktuellen Browsern sehr gut angewendet werden, sogar im Microsoft Internet Explorer 6 für Windows, sofern Sie sich im Compliance-Modus befinden. Leider funktioniert es in früheren Versionen des Internet Explorer für Windows nicht, so dass sein Einsatz auf Situationen begrenzt ist, in denen die Anforderungen der Browserbasis nicht unter der Version 6.0 liegen.

TABELLE 1 beschreibt die wichtigsten Browser, die dieses Verfahren unterstützen.

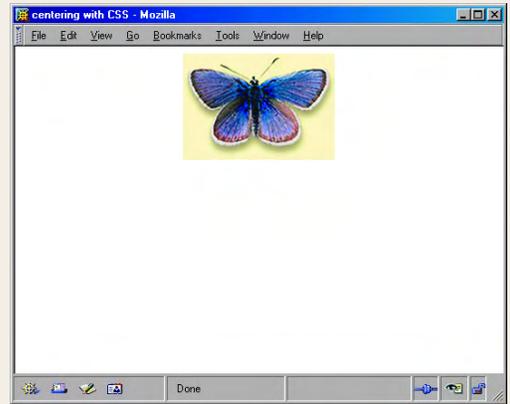


ABBILDUNG 3 Zentrieren eines Elements mit automatischen Rändern

Hinweis

Der Compliance-Modus ist Teil einer Browser-Technologie, die als DOCTYPE-Switching bezeichnet wird. Der XHTML-Code des CSS-Zen-Gartens nutzt den richtigen DOCTYPE, um den Compliance-Modus aufzurufen. Mehr über dieses wichtige Konzept erfahren Sie in dem Artikel »Doctype switching and standards compliance: An overview« von Matthias Gutfeldt (<http://gutfeldt.ch/matthias/articles/doctypeswitch.html>).

TABELLE 1 Browser-Unterstützung für automatische Ränder

Browser	Version	Unterstützung
Internet Explorer	6.0, Compliance-Modus	Ja
Internet Explorer	6.0, Quirks-Modus	Nein
Internet Explorer	5.5 Windows	Nein
Internet Explorer	5.0 Windows	Nein
Internet Explorer	5.2 Macintosh	Ja
Mozilla	Alle aktuellen Versionen	Ja
Mozilla Firefox	Alle Versionen	Ja
Netscape	4.x	Nein
Netscape	6.x+	Ja
Opera	6.0, 7.0, 8.0 Macintosh und Windows	Ja
Safari	1.2	Ja

Trotz der Probleme bei der Unterstützung plädieren die meisten Designer dafür, dieses Verfahren in allen möglichen Fällen einzusetzen, da sie es als die richtige Methode zum horizontalen Zentrieren von Elementen in CSS betrachten.

ZENTRIERTE GESTALTUNG DURCH TEXTAUSRICHTUNG

Eine weitere Zentriermethode erfordert die Verwendung der Eigenschaft `text-align`, die auf den Wert `center` gesetzt und auf das Element `body` angewendet wird. Dabei handelt es sich durch und durch um einen Hack, der aber in einer recht großen Anzahl von Browsern funktioniert, so dass er sehr gelegen kommt.

Die Methode ist ein Hack, weil eine Texteneigenschaft auf ein Containerelement und nicht auf Text angewendet wird. Dadurch entsteht wiederum zusätzliche Arbeit für Sie.

Nach Erstellen der für Ihr Layout erforderlichen `div`-Container wenden Sie die Eigenschaft `text-align` wie folgt an:

```
body {  
    text-align: center;  
}
```

Und welches Problem muss jetzt bewältigt werden? Alle Nachkommen von `body` werden zentriert (**ABBILDUNG 4**).

Das bedeutet, dass Sie zusätzliche Regeln festlegen müssen, um das Problem zu korrigieren und den Text auf seine Standardwerte zurückzusetzen, z.B.

```
p {  
    text-align: left;  
}
```

Wie Sie sich vorstellen können, ergibt sich daraus ein gewisser Nachteil. Außerdem wird von den wirklich standardkonformen Browsern nicht der Container, sondern nur der Text ausgerichtet.

RÄNDER UND TEXTAUSRICHTUNG KOMBINIERT EINSETZEN

Aufgrund der für die Textausrichtung bestehenden Abwärtskompatibilität und der heute weit verbreiteten Unterstützung automatischer Ränder kombinieren viele Designer die beiden Methoden zu einem ultimativen browserübergreifenden Hack zum horizontalen Zentrieren:

```
body {  
    text-align: center;  
}  
#container {  
    margin-left: auto;  
    margin-right: auto;  
    border: 1px solid red;  
    width: 168px;  
    text-align: left  
}
```

Leider immer noch nicht perfekt und immer noch ein Hack. Sie müssen zusätzliche Regeln festlegen, um die Textausrichtung zu steuern. Sie erzielen aber jetzt recht gute browserübergreifende Ergebnisse.

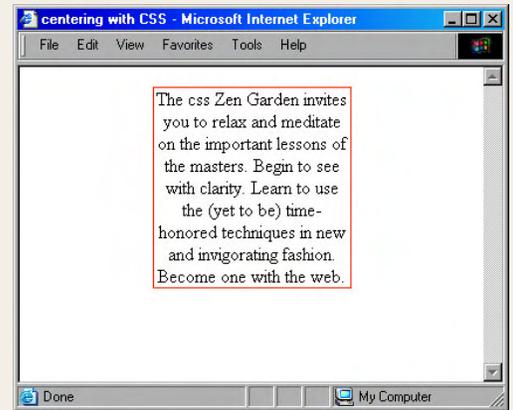


ABBILDUNG 4 Zentrieren mit Hilfe der Eigenschaft `text-align` im Internet Explorer. Beachten Sie, wie der Text des gesamten Absatzes nun innerhalb des `div`-Containers zentriert ist.

Hinweis

Es ist wichtig zu erwähnen, dass Ränder mit negativen Werten eine völlig legitime Verwendungsmöglichkeit von CSS darstellen. Der Einsatz negativer Ränder kann eine Reihe von Designanforderungen erfüllen, z.B. das Überlappen bestimmter Elementfelder.

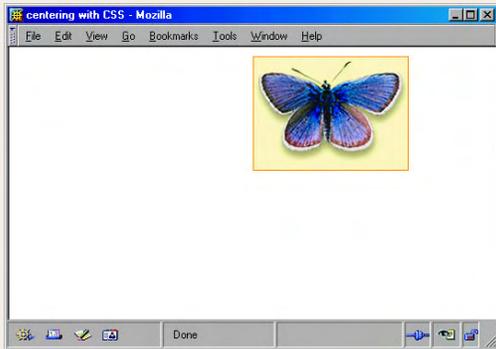


ABBILDUNG 5 Absolute Positionierung des Containerblocks mit einem Abstand von 50 Prozent vom linken Rand

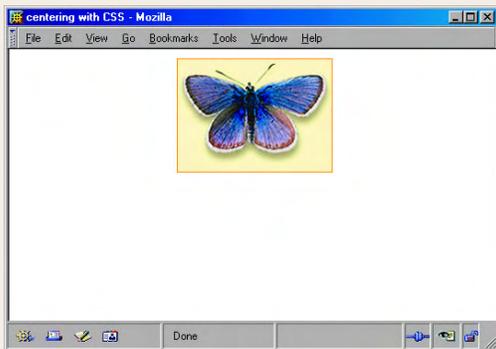


ABBILDUNG 6 Fügt man einen negativen Rand hinzu, der halb so breit ist wie die gesamte Breite des Elements, wird das Feld horizontal zentriert.

Hinweis

Seien Sie vorsichtig bei Ihren Berechnungen! Padding oder Rahmen müssen in die Breitenberechnungen einbezogen werden; andernfalls erhalten Sie nicht die gewünschten Ergebnisse.

DER KNIFF MIT DEM NEGATIVEN RAND

Angesichts der Herausforderung, Entomology zu zentrieren, beschloss Hicks, sich mit einer weiteren Zentrierungsmethode zu beschäftigen: mit der Verwendung negativer Ränder.

Die Lösung, negative Ränder zum Zentrieren einzusetzen, umfasst jedoch etwas mehr als die bloße Verwendung negativer Ränder. Es handelt sich um eine Kombination aus absoluter Positionierung und einem negativen Randwert.

Das funktioniert wie folgt: Zuerst wird ein Container erstellt, absolut positioniert und ein Abstand von 50 Prozent zum linken Rand festgelegt. Dies allein führt dazu, dass der linke Rand des Containerblocks bei 50 Prozent der Seitenbreite beginnt (**ABBILDUNG 5**).

Anschließend wird der linke Rand des Containers auf einen negativen Wert gesetzt, und zwar auf die *Hälfte* der Breite des Containerelements. Damit wird der Containerblock horizontal zentriert (**ABBILDUNG 6**).

Im Folgenden sehen Sie den CSS-Code von Entomology:

```
#container {
    background: #ffc url(mid.jpg) repeat-y center;
    position: absolute;
    left: 50%;
    width: 760px;
    margin-left: -380px;
}
```

Gemerkt ? Keine Hacks! Obwohl dies nicht die bevorzugte Methode ist, handelt es sich doch um eine großartige Hilfslösung, die breite Unterstützung findet. Ziemlich beeindruckend – sogar Netscape Navigator 4.x unterstützt dieses Verfahren, so dass es äußerst nützlich ist, wenn Sie eine große Anzahl von Browsern unterstützen müssen.

Stilvoll geht die Welt zugrunde

Trends kommen und gehen, aber feste Designs werden uns noch lange Zeit begleiten. Sie stellen bestimmte Mittel zur Gestaltung von Webseiten bereit und sind Teil des Werkzeugkastens eines jeden Designers.

Welchen Ansatz Sie wählen, hängt von Ihrer Browser-Basis und persönlichen Vorlieben ab, aber wie bei Entomology ist die Verwendung negativer Ränder und der absoluten Positionierung zum Zentrieren eine stilvolle Möglichkeit, die in einer Vielzahl von Browsern funktioniert – sogar in solchen, von denen man es nicht erwartet.



Jens Kristensen, Designer

www.csszengarden.com/036

White Lily

Prinzipien und Prozesse zum Gestalten wirkungsvoller Layouts

Select a Design

- [The Final Endings](#) by Ray Henry
- [contemporary nouveau](#) by David Helling
- [El Collar de Tomas](#) by Maria Stultz
- [OrderedZen](#) by Steve Smith
- [Bansai](#) by Martin Plazotta
- [Twe Thing](#) by Michal Makrzycki
- [Escudo Sahara](#) by John Barrick
- [Orasen](#) by Matthew Buchanan

Archives

- [next designs >](#)
- [View All Designs](#)

Resources

- [View This Design's CSS](#)
- [CSS Resources](#)
- [FAQ](#)
- [Submit a Design](#)
- [Translations](#)

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WAP, and the major browser creators.

The css Zen Garden invites you to relax, and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

So What Is This About?

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. To date, most examples of neat tricks and hacks have been demonstrated by structuralists and coders. Designers have yet to make their mark. This needs to change.

Participation

Graphic artists only please. You are modifying this page, so strong CSS skills are necessary, but the example files are commented well enough that even CSS novices can use them as starting points. Please see the [CSS Resource Guide](#) for advanced tutorials and tips on working with CSS.

You may modify the style sheet in any way you wish, but not the HTML. This may seem daunting at first if you've never worked this way before, but follow the listed links to learn more, and use the sample files as a guide.

Download the sample [HTML file](#) and [CSS file](#) to work on a copy locally. Once you have completed your masterpiece (and please, don't submit half-finished work) upload your .css file to a web server under your control. [Send us a link](#) to the file and if we choose to use it, we will spider the associated images. Final submissions will be placed on our server.

Benefits

Why participate? For recognition, inspiration, and a resource we can all refer to when making the case for CSS-based design. This is sorely needed, even today. More and more major sites are taking the lead, but not enough have. One day this gallery will be a historical curiosity; that day is not today.

Requirements

We would like to see as much CSS1 as possible. CSS2 should be limited to widely-supported elements only. The css Zen Garden is about functional, practical CSS and not the latest bleeding-edge tricks viewable by 2% of the browsing public. The only real requirement we have is that your CSS validates.

Unfortunately, designing this way highlights the flaws in the various implementations of CSS. Different browsers

OBWOHL ES KEINE bestimmten optischen Anforderungen an die Vorlagen für Zen Garden gibt, stellte **Jens Kristensen** fest, dass es vier klare Schlüsselwörter gibt, die Aufmerksamkeit erfordern: Visualität, CSS, Zen und Garten. Diese Begriffe gaben ihm den Rahmen für seine Arbeit, die bei dem wirkungsvollen botanischen Thema entspannend und beruhigend wirken sollte.

Nachdem er das Thema festgelegt hatte, prüfte er den Inhalt und entwarf für die definierten Ziele geeignete Layouts. Aus dieser systematischen Vorgehensweise entstand White Lily mit einem praktischen, nützlichen Layout, das weltweit vielen Sites als Inspiration diente.

Die Layoutprinzipien des Web

Wir sprechen oft über das Wort Layout und meinen dabei einfach die Anzahl der Spalten, die wir auf einer Seite anordnen wollen. Viele, die CSS lernen, gehen auf diese Weise an das Layout heran und fragen: »Wie erstelle ich ein dreispaltiges Layout?« oder »Wie erstelle ich ein Layout mit einer Kopf- und einer Fußzeile?« Das sind gute und wichtige Fragen, aber bevor wir zu CSS kommen, sollten wir die der Entscheidung für ein bestimmtes Layout zugrunde liegenden Prinzipien sorgfältig studieren.

Verfahren, die beim Entwerfen wirkungsvoller Layouts helfen, wurzeln in Prinzipien, die es schon lange vor der Zeit des Web gab. Obwohl sie sicherlich für das Web verändert wurden, bilden diese Prinzipien die notwendige Grundlage für das Erstellen von Layouts, die nicht nur großartig aussehen, sondern auch gut funktionieren.

Layouts in allen grafischen Formaten dienen in erster Linie der wirkungsvollen Anordnung von Inhalten. Wie sich dieses optisch gestaltet, ist unterschiedlich und hängt von Ihrem Medium, dem Inhalt und der Zielgruppe ab. Ein weiterer wichtiger Aspekt beim Weblayout ergibt sich aus der Tatsache, dass das Layout auch Funktionen unterstützen muss. Links, Menüsysteme und Formulare sind Beispiele für die wirksam in ein erfolgreiches Layout zu integrierende Funktionalität. Letztlich bildet das Layout den Grundstein für die Konsistenz einer Website, die für die Anwenderfreundlichkeit und das menschliche Fassungsvermögen von großer Bedeutung ist.

Erst denken, dann...

Eine der nützlichsten Vorgehensweisen, die Kristensen später beim Erstellen von White Lily half, war sein Ansatz, zunächst den Inhalt zusammenzustellen, mit dem er anschließend arbeiten würde.

Hinweis

Leser, die mit anderen Formen grafischer Gestaltung vertraut sind, z.B. bei Druckerzeugnissen und Verpackungen, werden schnell feststellen, dass das Erstellen von Layouts am Bildschirm eine einzigartige Herausforderung ist. Neulinge im Studium des Webdesigns lernen schnell, dass die Gestaltung einer Seite entscheidende Unterschiede zur Druckgestaltung aufweist, obwohl dabei einige gemeinsame Verfahren genutzt werden.

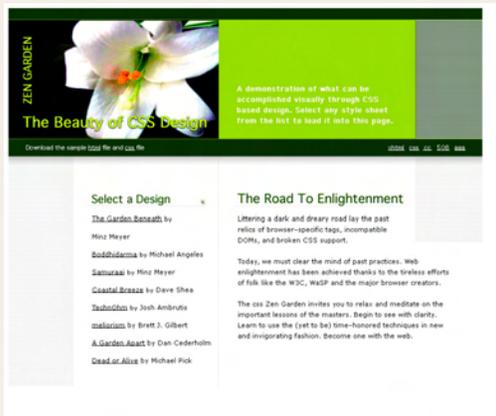


ABBILDUNG 1 Ein alternativer Stil für White Lily



ABBILDUNG 2 Drei bildhafte Blumensymbole, die innerhalb von White Lily verwendet werden, um die botanische Thematik zu unterstreichen

Die Zusammenstellung des Inhalts dient mehreren Zwecken. Sie bekommen sofort einen klaren Überblick über die zu bewältigenden Arbeiten, was natürlich zum Festlegen von Prioritäten für das Material führt. Dies regt zu weiteren Gedanken über das Layout und verschiedene Ansätze an, die zur Erreichung der Form- und Funktionsziele geeignet sein könnten.

Kristensen unterteilte den Inhalt von CSS Zen Garden zunächst in drei Kategorien:

- Allgemeine Informationen über Zen Garden, z.B. Templates, Validierungslinks und eine Übersicht über den Garten
- Links und Navigation von Zen Garden
- Hauptinhalt

Unter Verwendung dieser besonderen Informationen begann er, an der Zusammenstellung der Seite zu arbeiten. Er verfolgte nun zwei ganz spezielle Ziele: seinen thematischen Vorstellungen zu entsprechen und den Inhalt in drei verschiedenen Abschnitten zu präsentieren. Mehrere Schritte waren erforderlich, um zu einer Entscheidung über den endgültigen Stil zu gelangen (**ABBILDUNG 1**).

Schließlich entschied er sich dann für den Stil, den er dann als fertiges White-Lily-Projekt vorlegte. Es ist wichtig, dass die Auswahl Ihres Layouts auf dem Inhalt und den optischen Anforderungen beruht und nicht zufällig getroffen wird oder deshalb, weil ein bestimmtes Layout gerade „trendy“ ist.

Die Tricks im Layout von White Lily

Mit einer klaren Vorstellung über die weitere Vorgehensweise traf Kristensen seine Wahl über die Verarbeitung des Themas und des Inhalts.

Kristensen stellte fest, dass das Foto der Lilie die Grundlage für den botanischen Eindruck bot, den er herstellen wollte. Aus dem Foto zog er die Inspiration für sein Farbschema. Außerdem wählte er sorgfältig zusätzliche Ausschmückungen aus, die das botanische Konzept erweitern (**ABBILDUNG 2**).

Tatsächlich glänzt Kristensens Arbeit, wenn es um den Inhalt geht. Mit einem so klar definierten Ziel war es einfach, die Prioritäten für die Inhalte und deren effektive Behandlung über das Layout zu bestimmen.

Indem er die allgemeinen Informationen über den Garten in der Kopfzeile platzierte, erreichte er zwei wichtige Ziele: die Darstellung der »Marke« (in diesem Falle der Blume und des Seitentitels »CSS Zen Garden«) und die Verdeutlichung des Zwecks der Seite. Dieser Abschnitt umfasst auch die HTML- und CSS-Musterdateien mit den Validierungslinks (**ABBILDUNG 3**).

Download the sample [html file](#) and [css file](#)

[xhtml](#) [css](#) [cc](#) [508](#) [aaa](#)

ABBILDUNG 3 Die Behandlung der expliziten Anforderungen: die effektive Zusammenstellung der Inhalte

Dadurch wird das restliche Layout freigegeben, um zwei wesentliche Punkte zu bearbeiten: die Navigation und den Hauptinhalt. Zur Navigation nutzte Kristensen die linke Spalte, was sehr gut funktioniert, da sie sich damit an einer Stelle befindet, an der man sie auch erwartet. Die rechte Spalte wird für den Hauptinhalt genutzt. Beide Abschnitte sind hell gestaltet und bleiben frisch und sauber. Alles in allem sind die Kopfzeile und das zweiseitige Layout für die Ziele und Bedürfnisse gut geeignet und ergeben ein unkompliziertes, einfach funktionierendes Design.

Inspiration und Ratschläge

Wenn Sie erst einmal eine Vorstellung von Ihrem Gesamtziel haben, können Sie die Vorteile hilfreicher Informationen nutzen, die Ihnen eine Anleitung zum Verfeinern des Layouts bieten. Dieser Abschnitt erkundet einige der Hauptgesichtspunkte bei der Arbeit an Weblayouts. Zwei wichtige Regeln bieten Richtlinien für die Gestaltung großartiger Weblayouts – das Design und die Anwenderfreundlichkeit der Benutzerschnittstelle.



ABBILDUNG 4 Die bildliche Darstellung ist ein wichtiger Aspekt des Schnittstellendesigns. Ein Einkaufswagensymbol wie dieses von amazon.com ist ein allgemein bekanntes Symbol, das für die Verwendung im Onlinehandel übernommen wurde.



ABBILDUNG 5 Die Navigationslinks von White Lily

DIE BENUTZERSCHNITTSTELLE GESTALTEN

Im Gegensatz zur Druckgestaltung bietet eine Webseite Interaktion, d.h., sie enthält nicht nur eine passive Nachricht. Die Gestaltung der Benutzerschnittstelle ist aufgrund der vielen Aspekte der Computertechnik, für die interaktive Schnittstellen erforderlich sind, ein recht ausgereiftes Gebiet. Beispiele dafür sind sowohl Betriebssysteme als auch Softwareanwendungen.

Einige der grundlegenden Prinzipien für die Gestaltung der Benutzerschnittstelle sind im Folgenden aufgeführt:

- Lernen Sie Ihre Benutzer kennen, da sie diejenigen sind, für die Sie Ihr Design erstellen.
- Sorgen Sie dafür, dass sich der Benutzer innerhalb der Seite und der Site zurechtfindet.
- Verwenden Sie vertraute bildliche Darstellungen. Ein großartiges Beispiel dafür ist das Einkaufswagensymbol auf einer E-Commerce-Site. Sogar Webneulinge verstehen, was dieses Symbol bedeutet (**ABBILDUNG 4**).
- Stellen Sie sicher, dass funktionsrelevante Eigenschaften deutlich sichtbar sind. Dies begegnet uns bei White Lily in Form von Linklisten, die ihren Zweck präzise wiedergeben (**ABBILDUNG 5**).
- Sorgen Sie für konsistente Designelemente. Wenn White Lily als Entwurf für eine vollständige Website verwendet würde, müsste die Navigation konsistent bleiben, wie es auch bei der Position des Fotos und des Titellogos der Fall sein sollte. Ein Wechsel der Position der Marke und der Funktionselemente kann bei den Benutzern zu Verwirrungen führen, so dass Sie solche radikalen Änderungen zwischen den einzelnen Seiten vermeiden sollten.
- Machen Sie sich den visuellen Schwerpunkt klar. Es wird immer einige Objekte in einem Design geben, die die Blicke mehr auf sich ziehen als andere Objekte. Bei White Lily bildet die Blume selbst diesen Schwerpunkt. Wenn der Designer die Bedeutung des Bilds erkennt, kann er die anderen Seitenelemente so gestalten, dass

sie für eine entsprechende Ausgewogenheit im Design sorgen.

- Schaffen Sie Klarheit darüber, um welchen Inhalt es auf der Seite geht. Menschen haben eine sehr kurze Aufmerksamkeitsspanne; normalerweise warten sie *höchstens* neun Sekunden, um sich zu informieren, worum es geht – finden sie es innerhalb dieser Zeit nicht heraus, verlieren Sie möglicherweise vollständig ihre Aufmerksamkeit. Kristensen weckt die Aufmerksamkeit des Betrachters, indem er die wichtigste Mitteilung in einem eigenen Bereich innerhalb des Layouts platziert (**ABBILDUNG 6**).

Schnittstellendesigner sind ebenfalls sehr interessiert an der Ästhetik, sofern sie sich auf die Funktionen bezieht. Beachten Sie die Pfeile neben den Links in den Kopfzeilen von White Lily. Alle weisen dieselbe Größe auf und sollen den Besuchern eine ästhetisch ansprechende und hilfreiche Orientierungsmöglichkeit bieten (**ABBILDUNG 7**).

ANWENDERFREUNDLICHKEIT

Obwohl sie konzeptionell mit dem Design der Benutzerschnittstelle verwandt ist (und über ähnliche historische Wurzeln verfügt), hat die Anwenderfreundlichkeit einen anderen Schwerpunkt. Die Gestaltung der Benutzerschnittstelle befasst sich natürlich mit den Benutzern – und dabei geht es vor allem um die Art und Weise, in der die Arbeit von Entwicklern und Konstrukteuren den Bedürfnissen der Benutzer gerecht wird.

Bei den Forschungen zur Anwenderfreundlichkeit („Usability“) liegt der Schwerpunkt jedoch auf der endgültigen Erlebniswelt des Benutzers und der Art und Weise, in der der Benutzer mit einer ausgefeilten Benutzerschnittstelle arbeitet. Beide Forschungen werden in dem Sinne integriert, dass die Anwenderfreundlichkeit die Einarbeitung in die Methoden und Prozesse des Benutzerschnittstellendesigns betrifft und dass Benutzerschnittstellen den Anwendern zu Testzwecken und letztlich zur Durchführung von Verbesserungen und Änderungen präsentiert werden.



ABBILDUNG 6 Die Aufgabe der Seite in einem eigenen Bereich innerhalb des Layouts zu verdeutlichen, trägt zur leichten Verständlichkeit des Zwecks der Seite bei.



ABBILDUNG 7 Die bei den Hauptüberschriften in den Linklisten verwendeten Pfeile sind sowohl ästhetisch ansprechend als auch funktionell und lenken das Auge des Betrachters auf den darunter stehenden Inhalt.

Hinweis

Eine großartige Ressource für die Gestaltung von Benutzerschnittstellen und interaktive Designs finden Sie bei Boxes and Arrows (www.boxesandarrows.com).



ABBILDUNG 8 Die Kopfzeile mit wichtigen Informationen über die Seite befindet sich im oberen Bildschirmabschnitt.

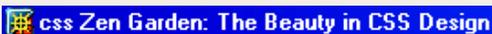


ABBILDUNG 9 Auch wenn es sich nicht um einen speziellen Teil des Layouts handelt, hilft ein Titel in der Titelleiste dem Benutzer, sich innerhalb einer Site zu orientieren.

This is a learning exercise as well as a demonstration. You retain full copyright on your graphics (with limited exceptions, see [submission guidelines](#)), but we ask you release your [CSS](#) under a Creative Commons license identical to the [one on this site](#) so that others may learn from your work.

ABBILDUNG 10 Die Angabe des Copyrights oder der Lizenzbedingungen ist ein wichtiger Teil des Inhalts, der innerhalb des Layouts wirkungsvoll erscheinen sollte.

Hinweis

In vielen Layouts werden Fußzeilen für den Hinweis auf Copyrights, den Datenschutz, die Lizenzbedingungen und damit zusammenhängende Informationen verwendet. Dies ist eine gebräuchliche Konvention und nützlich, wenn Sie Sites gestalten, bei denen diese Art von Informationen wesentlich für die Organisation oder Firma sind.

Ebenso wie zum Design der Benutzerschnittstelle gibt es bereits seit langem Forschungen zur Anwenderfreundlichkeit, die ebenfalls aus anderen Bereichen der Computertechnologie hervorgegangen ist. Bei der Anwenderfreundlichkeit im Web handelt es sich um eine Untergruppe der Usability, die über eigene Vorschläge und Richtlinien verfügt.

Im Folgenden sind einige mit dem Layoutdesign verbundene Aspekte der Anwenderfreundlichkeit aufgeführt:

- Wichtige Informationen sollten so platziert werden, dass sie wie in einer Zeitung auf der gefalteten Titelseite sichtbar sind. Der Sinn ist eindeutig: Stellen Sie sicher, dass alle Informationen, die dem Benutzer sofort zur Verfügung stehen müssen, auf der Seite möglichst weit oben platziert werden. Bei White Lily wird dies sichergestellt, indem die »Marke« und der Seitentitel sofort sichtbar sind (**ABBILDUNG 8**).

- Verwenden Sie stets eine Beschreibung innerhalb des Titelements:

```
<title>css Zen Garden: The Beauty in CSS Design</title>
```

Damit wird ein Titel in die Browserleiste gesetzt, der bei der Orientierung hilft (**ABBILDUNG 9**).

- Stellen Sie sicher, dass die Links und Navigationssysteme innerhalb eines Layouts so konsistent wie möglich positioniert werden. Sorgen Sie außerdem dafür, dass Links zur Homepage eingefügt werden. Links zu Seiten mit Copyright Hinweisen, Datenschutzrichtlinien und Kontaktinformationen sind stets förderlich. Im Fall von CSS Zen Garden sind alle Designs durch eine Creative-Commons-Lizenz geschützt, die innerhalb des Texts deutlich kenntlich gemacht wurde (**ABBILDUNG 10**).
- Für die meisten mittelgroßen bis großen Sites ist eine Suchfunktion *äußerst empfehlenswert*. Offensichtlich handelt es sich bei CSS Zen Garden eher um die Demonstration von Designansätzen als um individuelle Sites; in der Praxis sind die Benutzer jedoch dankbar für eine Suchfunktion und nutzen sie regelmäßig. Das ist der Fall, wenn sie an intuitiver Stelle innerhalb des Layouts platziert wird. Normalerweise sollten Suchfelder an einer logischen Stelle erscheinen: Findet man sie in

der Nähe der Navigation, ist das eine gute Idee. Wenn Sie so tun müssten, als verfügte White Lily über eine Suchfunktion, könnte sie direkt über den Linklisten erscheinen (**ABBILDUNG 11**).

- Unterteilen Sie Spalten durch Absätze und Abstände. Dies kann äußerst hilfreich sein, um einen Gesamtaufbau zu erzielen und das Verständnis der Benutzer zu verbessern (**ABBILDUNG 12**).

Natürlich gibt es noch viele weitere Faktoren, die zur Anwenderfreundlichkeit beitragen; die Einbeziehung dieser Verfahren während der Arbeit an Ihrem Layout wird Ihnen aber dabei helfen, wichtige Einzelheiten im Hinblick auf die Funktion und das Format und letztlich darauf, wie die Benutzer Ihr Design erleben, auszuarbeiten.

Gehen Sie gezielt vor

White Lily zeigt, wie die geplante Vorgehensweise dazu führen kann, viele an ein Layout gestellte Anforderungen zu erfüllen. Natürlich verläuft der Designprozess bei den einzelnen Designern und Designerteams unterschiedlich und Ihr Prozess wird wahrscheinlich in einigen wesentlichen Punkten davon abweichen.

Egal wie Ihr persönlicher Prozess verläuft – eines ist allerdings klar: Jens Kristensen konnte in diesem Fall dadurch, dass er zuerst das Thema festgelegt und dann den Inhalt analysiert hat, einem sehr klaren Weg zur Erreichung des Endergebnisses folgen: ein attraktives und verständliches Layout, das die Benutzer erfreut.



ABBILDUNG 11 Wenn in diesem Layout eine Suchfunktion zur Verfügung gestellt werden sollte, könnte diese möglicherweise über den Link innerhalb des Navigationsbereichs des Layouts erscheinen.



ABBILDUNG 12 Interessanterweise kann das Einrücken oder Versetzen von Text innerhalb einer Spalte zum Verständnis der Funktion beitragen. In diesem Fall wird in der Hauptlinkliste ein Versatz vorgenommen, so dass die Spalte unterteilt wird und das Gesamtlayout offener wirkt.

prêt-à-porter

Horizontale Linien in einer vertikalen Welt erforschen

EINE DER VON DESIGNERN am häufigsten gestellten Fragen lautet: »Wo finde ich Inspiration?« Vielen Anfängern wird beigebracht, sich an verschiedenen Orten umzusehen. Die Natur bietet beispielsweise Landschaften voller Formen und Bewegung, die eine ungeheure Quelle der Inspiration und Erquickung sein können.

Eine weitere Inspirationsquelle sind das Wesen des Menschen und seine Gestalt. Studenten der bildenden Künste verbringen zahllose Stunden im Atelier, um Körper zu studieren – ihre einzigartigen Formen, die Gesichtszüge und ihr Wesen. **Prêt-à-porter** von **Minz Meyer** fasst Formen der Natur und des menschlichen Charakters zusammen, woraus sich ein Gesamtdesign ergibt, so einzigartig wie ein Gesicht.

the road to enlightenment

Litfiring a dark and dreary road by the past relics of browser-specific tags, incompatible CSS, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the [CSS wizard](#) and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Design to see with clarity. Learn to use the (yet to be) time-honored techniques in new and inspiring fashion. Become one with the web.

so on what is this about?

There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external `css` file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. To date, most examples of neat tricks and hacks have been demonstrated by structuralists and coders. Designers have yet to make their mark. This needs to change.

participation

Graphic artists only please. You are modifying the page, so strong CSS skills are necessary, but the example files are commented well enough that even CSS novices can use them as starting points. Please see the [CSS Resource Guide](#) for advanced tutorials and tips on working with CSS. You may modify the style sheet in any way you wish, but not the HTML. This may seem daunting at first if you've never worked this way before, but follow the listed links to learn more, and use the sample files as a guide.

Download the sample [html file](#) and [css file](#) to work on it copy locally. Once you have completed your masterpiece (and please, don't submit half-finished work) upload your `css` file to a web server under your control. [Send us a link](#) to the file and if we choose to use it, we will update the associated images. Final submissions will be placed on our server.

benefits

Why participate? For recognition, inspiration, and a resource we can all refer to when making the case for CSS-based design. This is sorely needed, even today. More and more major sites are taking the leap, but not enough have. One day this gallery will be a historical curiosity, that day is not today.

requirements

We would like to see as much CSS1 as possible. CSS2 should be limited to widely-supported elements only. The css Zen Garden is about functional, practical CSS and not the latest bleeding-edge tricks viewable by 2% of the browsing public. The only real requirement we have is that your CSS validates.

Unfortunately, designing this way highlights the flaws in the various implementations of CSS. Different browsers display differently, even completely valid CSS at times, and this becomes maddening when a fix for one leads to browserglitch in another. View the [Resources](#) page for information on some of the fixes available. Full browser compliance is still sometimes a pipe dream, and we do not expect you to come up with pixel-perfect code across every platform. But do test it as many as you can. If your design doesn't work in at least 85% with and Mozilla (run by over 80% of the population), chances are we won't accept it.

We ask that you submit original artwork. Please respect copyright laws. Please keep objectionable material to a minimum, tasteful nudity is acceptable, outright pornography will be rejected.

This is a learning exercise as well as a demonstration. You retain full copyright on your graphics (with limited exceptions, see [submission guidelines](#)), but we ask you release your CSS under a Creative Commons license identical to the [one on this site](#) so that others may learn from your work.

Bandwidth graciously donated by [DreamFire Studios](#).



the beauty of css design

SELECT A DESIGN

- note
- by [brian williams](#)
- by [becca](#)
- by [chris morrell](#)
- by [beatrice](#)
- by [emichela maria nardi](#)
- by [akyrada](#)
- by [axel hebernatik](#)
- by [evanfeld](#)
- by [john cotan](#)
- by [6ix lifestyle](#)
- by [emiliano peresini](#)
- by [medilovee](#)
- by [emiliano peresini](#)
- by [pleasant day](#)
- by [kyle jones](#)

next designs »
view all designs

RESOURCES

- view this design's css
- css resources
- faq
- submit a design
- translations



Printmedien als Inspiration

Webdesigner, die bereits eine Weile im Berufsleben stehen, bekamen schon oft von Meinungsführern, Vordenkern und Ausbildern zu hören: »Das Web ist kein Printmedium.« Obwohl diese Aussage offensichtlich zutrifft, lautet eines der überzeugendsten Argumente, die prêt-à-porter uns vorhält, dass es auch dann keinen Grund gibt, warum ein Printmedium nicht als Inspiration für das Webdesign dienen könnte, wenn wir uns das Web nicht wie eine gedruckte Seite vorstellen sollten.

In diesem Fall war es nicht nur ein Printmedium, das als Inspiration für prêt-à-porter diente, sondern sogar ein Printmedium zum Thema Drucken (**ABBILDUNG 1**).

In Meyers Originalbroschüre für eine Druckerei wurde eine Kombination aus Bildern und Wörtern eingesetzt, um optische Spannung zu erzeugen. Auf der linken Seite des ursprünglichen Designs zeigt eine vertikal angeordnete Fotostrecke – die für die Broschüre entworfen und an einem Tag aufgenommen wurde – Variationen eines Frauengesichts mit außergewöhnlicher Retusche und Kolorierung. Auf der rechten Seite werden den Bildern Wörter und Sätze entgegengesetzt, z.B. das Wort »verfärbt«. Die daraus entstandene Broschüre wirbt mit überzeugenden Bildern und deutlicher Terminologie für die Qualität der Druckereiarbeiten und führt zu der Schlussfolgerung, dass die Druckerei keine Fehler bei der Wiedergabe von Bildern macht.

Insbesondere liegt die Originalbroschüre nicht im Hoch-, sondern im Querformat vor. Darin lag einer der faszinierendsten – und herausforderndsten – Aspekte des Layouts, als es darum ging, es an das Web anzupassen.



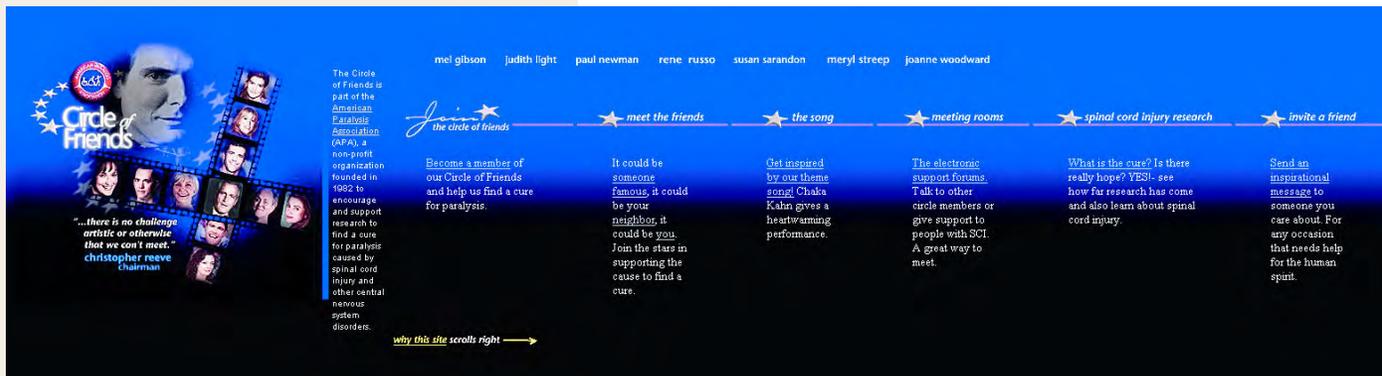
ABBILDUNG 1 Ein Ausschnitt aus einer gedruckten Broschüre diente als Inspiration für prêt-à-porter.

DIE REGELN BRECHEN

Obwohl sich das Querformat für Printmedien hervorragend eignen kann und daher häufig verwendet wird, fühlten sich Webdesigner lange darin bestärkt, derartige Designs zu meiden. Die Gründe dafür liegen vorwiegend in der Anwenderfreundlichkeit. Horizontale Designs zwingen den Browser normalerweise zur Anzeige einer Bildlaufleiste am unteren Rand des Dokuments, so dass die Besucher der Site horizontal hin- und herscrollen müssen. Dies wurde lange Zeit als problematisch angesehen, da die meisten Benutzer nicht geneigt sind, sich mehr als nötig zu bewegen, um an die Informationen zu gelangen. Außerdem ist es für Personen mit speziellen Behinderungen besonders schwer, eine horizontale Bildlaufleiste zu bewegen. Daher haben die meisten Webdesigner horizontale Designs vermieden, mit Ausnahme von Sonderfällen wie zum Beispiel experimentellen Anwendungen, oder sie erstellten derartige Designs leider versehentlich, wenn sie eine ungünstige Entscheidung im Hinblick auf die Bildschirmauflösung getroffen haben.

Eine seltene, frühe Ausnahme bildete eine Website, die 1998 für Circle of Friends veröffentlicht wurde – eine Gemeinschaft von unter Rückenmarksverletzungen Leidenden oder an diesem Thema Interessierten. Die Organisation, für die sich später übrigens unter anderem Christopher Reeves einsetzte, entschloss sich, eine Site im horizontalen Design zu entwerfen, weil viele Menschen mit schweren Rückenmarksverletzungen sich die Welt aus dieser Perspektive ansehen müssen. Damals war dieses Designkonzept revolutionär, es zählte zu denen, die mit größter Sicherheit von dem menschlichen Erleben inspiriert waren (**ABBILDUNG 2**).

ABBILDUNG 2 Die Website von Circle of Friends, die 1999 ins Leben gerufen wurde, mittlerweile jedoch nicht mehr existiert, ist ein frühes Beispiel für einen Regelbruch durch die Verwendung eines horizontalen Designs.



Herausgefordert durch die Vorstellung, dass ein horizontales Design starken Einfluss auf das Web nehmen könnte, entschloss sich Meyer, es zu versuchen und eine überzeugende Version des Printdesigns zu erstellen, die nicht nur im Web funktionieren, sondern auch CSS-Aspekte nutzen sollte, um das Design webwürdig zu gestalten.

Die Sache mit der Horizontalen

Während des Prozesses, die beeindruckende Botschaft der Originalbroschüre mit einem CSS-Layout zu verbinden, bestand eine der größten Hürden in dem unvermeidbaren Thema Browser-Unterstützung. Um ein für eine Vielzahl von Browser-Szenarien nützliches und ansprechendes Design zu entwerfen, waren Hilfslösungen erforderlich, um es browser- und betriebssystemübergreifend kompatibel zu gestalten.

AUSFLUG IN DIE FESTE POSITIONIERUNG

Das Bild der Frau inspirierte Meyer dazu, dafür zu sorgen, das Bild statisch im Design zu platzieren und den Text hinter das Bild laufen zu lassen. Dieser Effekt war ein erfrischender Ansatz, um das Design webfreundlich zu gestalten, was in der Printversion einfach nicht möglich ist.

Obwohl CSS dem Designer viele Möglichkeiten zur Implementierung von Hintergrundgrafiken bietet, hätte die Verwendung des Fotos der Frau als Hintergrundbild des Textkörpers zur Folge, dass der Text über das Bild läuft, was im Grunde die Kraft des Bilds zerstört und die Lesbarkeit des Texts auf null reduziert.

Tipp

Verwenden Sie `z-index`, um die Stapelreihenfolge der positionierten Felder festzulegen. Der höchste `z-index`-Wert erscheint im Stapel über den anderen Feldern.



ABBILDUNG 3 In Mozilla, das das feste Positionierungsschema unterstützt, bleibt das Bild an einer festen Position, während der Inhalt so gestaltet werden kann, dass er hinter das Bild läuft.

Hinweis

Obwohl er mit der CSS2-Spezifikation 1998 eingeführt wurde, fehlt in bestimmten Browsern – vor allem im Microsoft Internet Explorer – die Unterstützung des Werts `fixed` für die Eigenschaft `position`. Deshalb funktionieren die für dieses Szenario erstellten Regeln in kompatibleren Browsern zwar gut, beim Betrachten im Internet Explorer läuft das Bild jedoch aus der Seite heraus, was den Effekt zerstört.

Als Möglichkeit zur Problembehandlung wäre zu überlegen, ein `div`-Element mit fester Position einzusetzen und diesem ein Bild hinzuzufügen. Die feste Positionierung erlaubt es, dass ein beliebiges als feststehend definiertes Element völlig unbeweglich an seiner Position auf der Seite bleibt; daher läuft es beim Scrollen nicht mit, ganz unabhängig davon, was mit den anderen Seitenelementen geschieht. Somit ist es vorstellbar, dass Sie zu diesem Zweck einen Selektor mit den folgenden Deklarationen erstellen:

```
div#static-image {
    background-image: url(bg_face.jpg);
    background-repeat: no-repeat;
    background-position: left bottom;
    position: fixed;
    left: 0;
    bottom: 0;
    height: 594px;
    width: 205px;
    z-index: 2;
}
```

In einem wirklich CSS-kompatiblen Browser würde die feste Position dieses `div`-Elements erhalten bleiben, so dass andere Elemente auf der Seite dahinter positioniert werden können (**ABBILDUNG 3**).

AUF DEN BROWSER EINGEHEN

Wie es bei der Behandlung von browserübergreifenden Themen oft der Fall ist, ist es ratsam, ein Idealformat zu erstellen und dieses dann zur Verwendung in Browsern zurückzunehmen, die aus irgendeinem Grund einen bestimmten Effekt nicht unterstützen. Um das zu erreichen, können Sie verschiedene Designs erstellen und JavaScript zur Browsererkennung einsetzen; manche halten diese Vorgehensweise allerdings aus mehreren Gründen für ineffektiv, u.a. aufgrund der Tatsache, dass eine Reihe von Benutzern JavaScript in ihren Browsern deaktiviert.

Im Falle von prêt-à-porter bestand die Vorgehensweise darin, ein alternatives Design für den Internet Explorer zu erstellen, das gut aussehen und weich gerendert sein sollte, obwohl das Bild des Gesichts aus dem Anzeigebereich herausläuft.

Ebenso wie bei der festen Positionierung fehlt im Internet Explorer auch die Unterstützung untergeordneter Selektoren, die es uns erlauben, ein bestimmtes untergeordnetes Element eines vorgegebenen Elements auszuwählen. Untergeordnete Selektoren werden durch eine spitze Klammer (>) gekennzeichnet. Damit signalisiert der Selektor

```
p>strong
```

dass nur innerhalb von Absätzen vorkommende strong-Elemente durch die darauf folgende Deklaration gestaltet werden.

Aufgrund der Tatsache, dass Internet Explorer die feste Positionierung sowie untergeordnete Selektoren nicht unterstützt, wurden eine Reihe von Regeln erstellt, die es ermöglichen, in CSS-konformen Browsern das optimale Design und im Internet Explorer ein alternatives Design zur Anzeige zu bringen:

```
body#css-zen-garden>div#extraDiv2 {
  background-image: url(bg_face.jpg);
  background-repeat: no-repeat;
  background-position: left bottom;
  position: fixed;
  left: 0;
  bottom: 0;
  height: 594px;
  width: 205px;
  z-index: 2;
}
```



ABBILDUNG 4 Im Alternativlayout für den Internet Explorer 6.0 läuft die Abbildung aus dem Bildschirmbereich heraus, die Grundintegrität des Designs bleibt jedoch erhalten.



ABBILDUNG 5 Prêt-à-porter in Opera 7.0. In den Versionen vor Opera 7.5 erschien keine Bildlaufleiste, so dass viele Funktionen des Designs nicht genutzt werden konnten.

Der folgende CSS-Code kommt nur in Browsern zum Einsatz, die die vorangegangene Regel nicht verstehen, in diesem Fall der Internet Explorer. Da die untergeordneten Selektoren eine größere Genauigkeit implizieren, hat die erste Regel Vorrang, allerdings nur, wenn der Browser sie versteht.

```
div#extraDiv2 {
    background-image: url(bg_face_ie.jpg);
    background-repeat: no-repeat;
    background-position: left bottom;
    position: absolute;
    left: 0;
    bottom: 0;
    height: 600px;
    width: 265px;
}
```

Obwohl das fest positionierte Bild im Internet Explorer 6.0 aus dem Bildschirmbereich hinausläuft, bleibt das Design attraktiv und annehmbar (**ABBILDUNG 4**).

OPERA MEISTERN

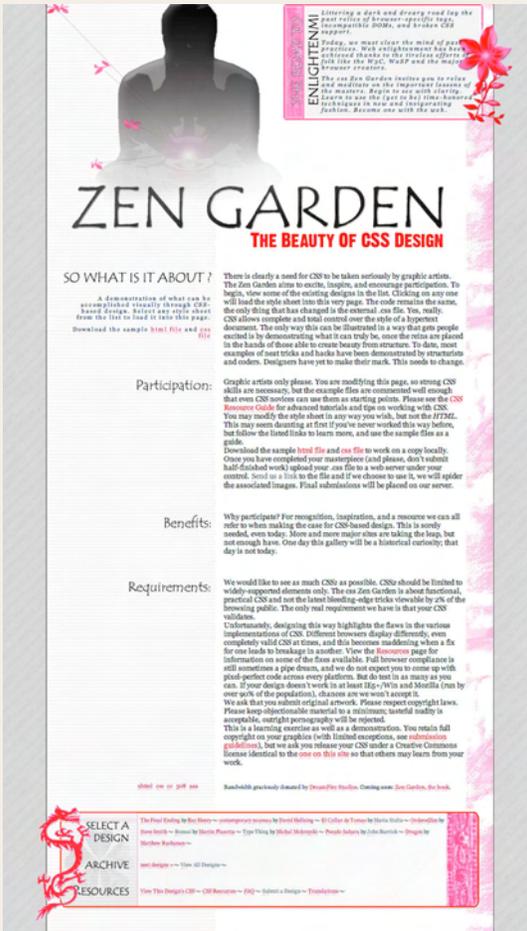
Ein weiteres Problem, das sich beim Erstellen von prêt-à-porter ergab, betraf die Unterstützung des Opera-Browsers von Opera Software. Selbst wenn sich die Browser-Entwickler intensiv mit der Implementierung von Standards befassen, wie es bei Opera der Fall ist, verläuft die Implementierung aus verschiedenen Gründen nicht immer perfekt.

In den Opera-Versionen vor der Version 7.5 erschien die horizontale Bildlaufleiste bei prêt-à-porter nicht. Natürlich wurde das Design dadurch in diesen Versionen völlig unbrauchbar (**ABBILDUNG 5**).

Glücklicherweise hat Opera das Problem gelöst und das Design funktioniert jetzt nicht einfach nur, sondern es funktioniert sogar in idealer Weise, da das fest positionierte Bild an Ort und Stelle bleibt, wie es in anderen, vollständig CSS-kompatiblen Browsern der Fall ist.

Erfahrungen

Zwischen dem Zeitpunkt, zu dem die Inspiration erfolgt, und dem Zeitpunkt, an dem sich aus dieser Inspiration ein vollständiges Design entwickelt hat, findet ein Prozess statt, der bei den einzelnen Designern so einzigartig ist wie das Ergebnis. Aus Meyers von einem horizontalen Printdesign inspiriertem Layoutansatz ergaben sich sowohl Möglichkeiten als auch Herausforderungen. Durch die Verwendung raffinierter Hilfslösungen erzielte Meyer ein bemerkenswert schönes Design, das uns nicht nur lehrt, in besonderer Weise darüber nachzudenken, welche einzigartigen Inspirationen den Layouts zugrunde liegen können, sondern auch zeigt, wie wirkungsvoll sie sich implementieren lassen.



Cedric Savarese, Designer

www.csszengarden.com/070

CS(S) Monk

Aussagekräftig positionieren und das Raster nutzen

WER SAGT, DASS SIE einen Gesamtplan brauchen? **Cedric Savarese** begann mit **CS(S) Monk**, indem er in Adobe Photoshop experimentierte und abwartete, was dabei herauskam. Der Vorteil liegt in der Flexibilität; wenn neue Ideen entstehen, können ältere Elemente verworfen oder so angepasst werden, dass sie sich in das überarbeitete Layout einfügen lassen.

Savarese entschied, die Kopfzeile als Schwerpunkt einzusetzen, und baute das restliche Layout darauf auf. Für Titel und Untertitel wählte er eine Schriftart aus, die zu dem in der Entwicklung befindlichen Thema passen sollte. Die Platzierung der Blumen verleiht dem Layout eine gewisse Tiefe, ein beruhigendes Farbschema in Rot- und Grautönen gestaltet das Design einladend.

Position und Bedeutung

Das dominierende Bild des Mönchs von CS(S) Monk nimmt den Platz ein, der den wichtigsten Schwerpunkt auf der Seite bildet, nämlich die obere linke Ecke. Das erste Element dieses Designs ist das wichtigste für das Gesamtwerk und befindet sich auch an der wichtigsten Stelle.

Wenn Sie an eine flache Leinwand mit einer festen Größe wie zum Beispiel ein Blatt Papier denken, so erhalten Sie durch horizontales und vertikales Falten vier Quadranten. In den westlichen Sprachen werden Sätze auf einer Seite von links nach rechts und von oben nach unten gelesen. Der Durchschnittsbetrachter hat daher unbewusst gelernt, jedem Quadranten eine unterschiedliche Bedeutung beizumessen.

Haben Sie sich jemals gefragt, warum auf so vielen Sites das Logo in der oberen linken Ecke zu finden ist? Das liegt daran, dass sich der Blick normalerweise als Erstes auf diese Stelle richtet. Das Platzieren eines Objekts im oberen linken Quadranten gewährleistet maximale Aufmerksamkeit, so dass er für die Markendarstellung der ideale Ort ist (**ABBILDUNG 1**).

Der obere rechte Quadrant ist ebenfalls von Bedeutung (**ABBILDUNG 2**). Auf einer Webseite ist der Bereich »über dem Falz« – der Bereich, der direkt zu sehen ist, ohne scrollen zu müssen – die am besten einsehbare Stelle, was für alle Seiten gilt. Mit der Platzierung eines Objekts in der oberen rechten Ecke wird suggeriert, dass dieses Objekt etwas unwichtiger ist als das in der oberen linken Ecke, aber immer noch bedeutend im Vergleich zu den restlichen Seitenelementen.

Die beiden unteren Quadranten haben auf den meisten Webseiten nicht viel zu bieten (**ABBILDUNG 3**). Einige Designer bringen dort eine redundante Navigation an, andere fügen nichts als einen einfachen Copyrighthinweis ein. In traditionellen Printmedien könnte das Platzieren eines Objekts am unteren Seitenrand den Eindruck vermitteln, dass die darüber befindlichen Elemente angehoben werden oder darüber schweben. Auf einer gedruckten Seite kann durch einfache Positionierung der Eindruck eines dreidimensionalen Raums geschaffen werden, im Web ist es jedoch wahrscheinlicher, dass ein Objekt am Seitenende unbeachtet bleibt oder völlig fehlt.

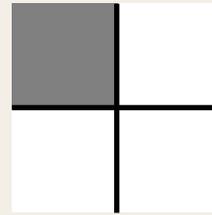


ABBILDUNG 1
Der Quadrant oben links

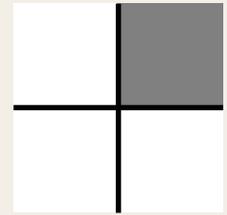


ABBILDUNG 2
Der Quadrant oben rechts

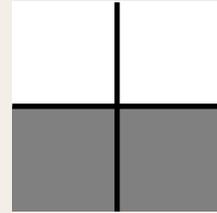


ABBILDUNG 3 Die Quadranten unten links und unten rechts

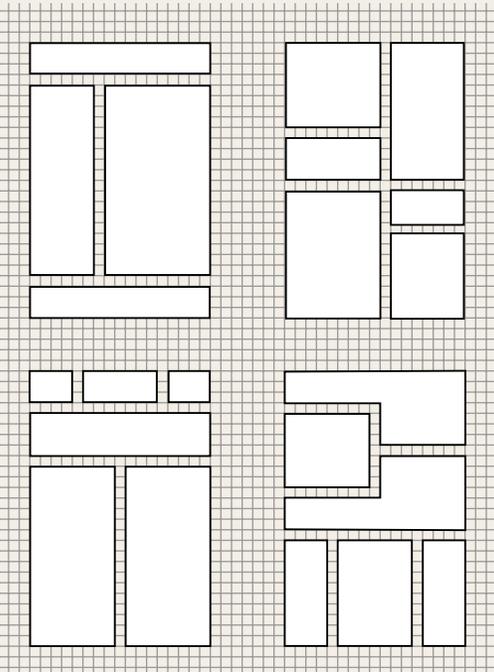


ABBILDUNG 4 Unterschiedliche Anordnung von Formen anhand eines Rasters

Eine interessante Vorgehensweise, die an Zugkraft gewinnt, ist die Positionierung unwesentlicher Extras am Ende einer Webseite. Allgemeine Links zu internen Inhalten oder Sitefunktionen können dort erscheinen, Links, die in der regulären Navigationshierarchie keinen Platz haben. Wenn die Benutzer sie übersehen, ist das kein Beinbruch – sie können den damit verknüpften Inhalt auf andere Weise finden. Für diejenigen, die diese zusätzlichen Links wahrnehmen, könnten sie wertvolle Zusatzfunktionen und wertvollen Kontext bereitstellen.

Das Raster

Ein Raster ist ein wichtiges Layoutwerkzeug, wird jedoch aufgrund seiner Starrheit häufig übersehen. Wie in der Mathematik handelt es sich bei einem Layoutraster einfach um eine Reihe gleichmäßig verteilter, sich kreuzender Linien, die eine Reihe von logischen, entsprechend konsistenten Feldern bilden. Diese Linien und Felder helfen dabei, die Voraussetzungen für die später hinzuzufügenden Designelemente zu schaffen. Halten Sie sich eng an das Raster, sind die Proportionen und die Abstände möglicherweise konsistenter, wenn auch vielleicht mit dem Nachteil, dass die Anordnung übermäßig ordentlich und vom kreativen Standpunkt aus langweilig wirken könnte (**ABBILDUNG 4**).

Das klassische tabellenbasierte Design zeigt die unfreiwillig einschränkende Beschaffenheit eines übermäßig strukturierten Rasters – ein in einer Tabellenzelle positioniertes Element hat keine andere Chance, als in dieser Zelle zu verbleiben. Wird ein Überschneidungseffekt gewünscht, müssen die benachbarten Zellen dazu herangezogen werden. Oft ist der einzige Weg, aus dem Raster auszubrechen, das Bild in winzige Teile zu zerschneiden und sie durch das Erstellen von Rastern innerhalb von Rastern oder den Einsatz komplexer verschachtelter Tabellen wieder zusammenzufügen.

CSS dagegen erlaubt eine wesentlich raffiniertere Steuerung der nicht an ein Raster gebundenen Elemente. Ein Layout kann auf einer Rasterstruktur beruhen, aber viele Elemente aufweisen, die sich fließender und spontaner über die Begrenzungen hinwegsetzen. Das CSS-Positionierungsmodell gewährt den Elementen viel mehr Freiheit. Es gestattet Ihnen, Elemente aus ihren Containern hinauszuschieben oder sie an beliebiger Stelle der Seite neu zu positionieren.

Das Raster konstruieren

Über das Raster hinauschieben bedeutet natürlich, dass ein solches zuerst erstellt werden muss. Legt man ein Raster über CS(S) Monk, scheint die Kopfzeile in drei separate Komponenten unterteilt zu sein und der Haupttextkörper erscheint in zwei Spalten.

KOPFZEILE

Auf der Basis des relativen Gewichts der einzelnen Elemente in der Kopfzeile existieren drei deutliche Bereiche: die Abbildung des Mönchs, der Titel »Zen Garden« und der Text innerhalb von #preamble, der aus dem regulären Dokumentfluss heraus nach oben in die Kopfzeile geschoben wurde.

Bei Überprüfung der einzelnen Bilder, aus denen sich die Kopfzeile von CS(S) Monk zusammensetzt, sieht es jedoch so aus, als ob zwei der Objekte, die Abbildung und der Titel, in einem Bild zusammengefasst wurden. Obwohl sie verschiedene Teile der optischen Struktur der Kopfzeile bilden, werden sie zu Kodierungszwecken als eine Einheit behandelt. Dies ist eine bekannte Eigenart des Webdesigns im Allgemeinen – die Bilder müssen vor dem Speichern zu einem einzigen Bild zusammengefügt werden, um das spätere Kodieren zu erleichtern (**ABBILDUNG 5**).

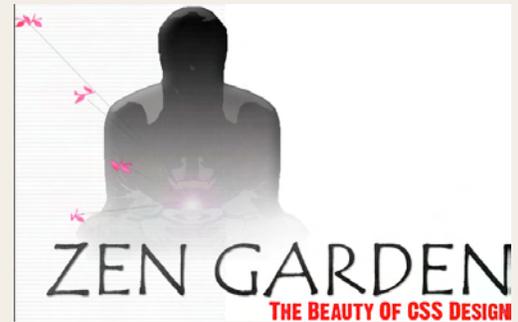


ABBILDUNG 5 Kombination der beiden Abbildungen und des Titels innerhalb ein und desselben Bildes

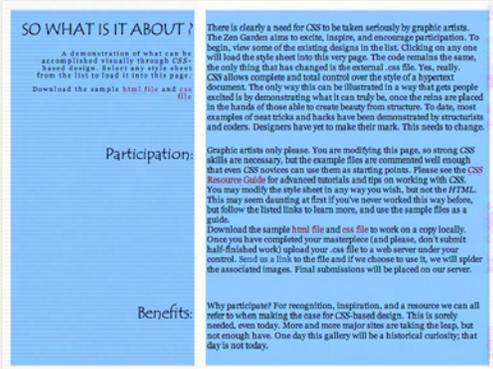


ABBILDUNG 6 Eine angedeutete zweite Spalte innerhalb des Textkörpers, die sich an die Rasterstruktur anlehnt

TEXTKÖRPER

Obwohl CS(S) Monk auf einer einzigen mittigen Textspalte beruht, wirken die am Rand platzierten Kopfzeilen wie eine angedeutete zweite Spalte. Ein einfaches blaues Overlay macht deutlich, wie sich die einzelnen Spalten im Verhältnis in die Seite einfügen (**ABBILDUNG 6**).

Die Spalte auf der linken Seite nimmt etwa zwei Drittel der Breite der rechten Spalte ein, so dass sich ein Verhältnis von 2:3 ergibt. Der Vorteil beim Erstellen einer solchen numerischen Verbindung zwischen Spalten liegt darin, dass sich die Proportionen automatisch ergeben; nachdem die Breite der ersten Spalte festgelegt wurde, wird die Breite der zweiten anhand des verbliebenen Platzes bestimmt. Bei dem 2:3-Verhältnis dieser beiden Spalten handelt es sich um ein klassisches, harmonisches Zusammenspiel, das einen natürlich und angenehm wirkenden typografischen Rhythmus schafft.

Weitere Informationen über Proportionen und Verhältnisse im Design finden Sie in *Grafik für Nicht-Grafiker. Ein Rezeptbuch für den sicheren Umgang mit Gestaltung. Ein Plädoyer für besseres Design* von Frank Koschembar (Westend, 2005).

Aus dem Raster ausbrechen

Für den Großteil der Seiten in CS(S) Monk erstellte Savarese zwar ein rasterbasiertes Layout, aber er hatte in ausreichendem Maße die Möglichkeit, Elemente zu verschieben, um die Struktur zu durchbrechen und zusätzliche, für dieses Layout interessante Hinweise anzubringen.

Das CSS-Positionierungsmodell bietet die erforderliche Flexibilität, und zwar unabhängig davon, ob Sie sich für eine absolute oder relative Positionierung entscheiden. Eine Analyse der Unterschiede zwischen den beiden Methoden hilft dabei zu ermitteln, in welchen Fällen Sie sich für die eine oder die andere Lösung entscheiden sollten.

ABSOLUTE POSITIONIERUNG

Die absolute Positionierung zu verstehen, bedeutet, das Konzept des Dokumentflusses zu verstehen. Bei der Kodierung ist HTML linear – ein h1-Element kann an erster Stelle stehen, gefolgt von einem p- und einem div-Element. Die Möglichkeit zum Verschachteln von Elementen besteht zwar, wird ein Dokument jedoch ohne Stil wiedergegeben, ist es klar, dass der HTML-Parser am Dokumentanfang beginnt und sich nach unten durcharbeitet (**ABBILDUNG 7**).

Die absolute Positionierung stellt die Möglichkeit bereit, ein Element nicht nur an eine beliebige Stelle der Seite zu verschieben, sondern es sogar aus dem Dokumentfluss zu entfernen. Ein absolut positionierter Block hat keinen Einfluss mehr auf die anderen Elemente des Dokuments und der lineare Fluss verläuft so, als wäre der Block nicht vorhanden (**ABBILDUNG 8**). Optisch erscheint er zwar an der Stelle, an der er positioniert wurde, für die anderen Elemente auf der Seite ist er aber nicht vorhanden. Die sich möglicherweise daraus ergebende Überschneidung zu bestimmen, ist Aufgabe des Designers.

Der einzige Fall, in dem ein absolut positionierter Block Einfluss auf ein anderes Seitenelement hat, tritt ein, wenn es sich bei dem zweiten um ein untergeordnetes Element des ersten handelt, das ebenfalls positioniert wurde. Das Positionierungsmodell schreibt vor, dass ein Element in Übereinstimmung mit seinem Containerblock positioniert wird; in den meisten Fällen handelt es sich bei dem Containerblock um das Element root `html`. Existiert aber zwischen dem betreffenden Element und dem Stammelement ein übergeordnetes Element, wird das untergeordnete stattdessen in Übereinstimmung mit dem übergeordneten Element positioniert.

Weitere Informationen über Containerblöcke finden Sie in der CSS-Spezifikation auf der Seite »10 Visual formatting model details« (www.w3.org/TR/REC-CSS2/visudet.html#containing-block-details).

css Zen Garden

The Beauty of CSS Design

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.

Download the sample [html file](#) and [css file](#)

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible *DOMs*, and broken *CSS* support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the *W3C*, *WaSP* and the major browser creators.

ABBILDUNG 7 Ein nicht mit einem Stil versehenes Markup liefert ein lineares, von oben nach unten verlaufendes Ergebnis.

css Zen Garden

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.

Download the sample [html file](#) and [css file](#)

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible *DOMs*, and broken *CSS* support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the *W3C*, *WaSP* and the major browser creators.

The *css Zen Garden* invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the *(not to be)* time-honored techniques in new and innovative

ABBILDUNG 8 Andere Elemente fließen so, als sei der positionierte Block nicht vorhanden.

css Zen Garden

The Beauty of CSS Design

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.

Download the sample [html file](#) and [css file](#).

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to

ABBILDUNG 9 Ein Element innerhalb des regulären Dokumentflusses

css Zen Garden

The Beauty of CSS Design

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.

Download the sample [html file](#) and [css file](#).

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to

ABBILDUNG 10 Ein relativ positioniertes Element verbleibt innerhalb des Dokumentflusses; seine Ausgangsposition hat weiterhin Einfluss auf andere Elemente, während seine neue Position vom Dokumentfluss ignoriert wird.

css Zen Garden

The Beauty of CSS Design

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.

Download the sample [html file](#) and [css file](#).

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

So What is This About?

ABBILDUNG 11 Dagegen übt ein absolut positioniertes Element keinen derartigen Einfluss auf den Dokumentfluss aus. Die Ausgangsposition des Elements wird vom Dokumentfluss ignoriert, das Gleiche gilt für seine neue Position.

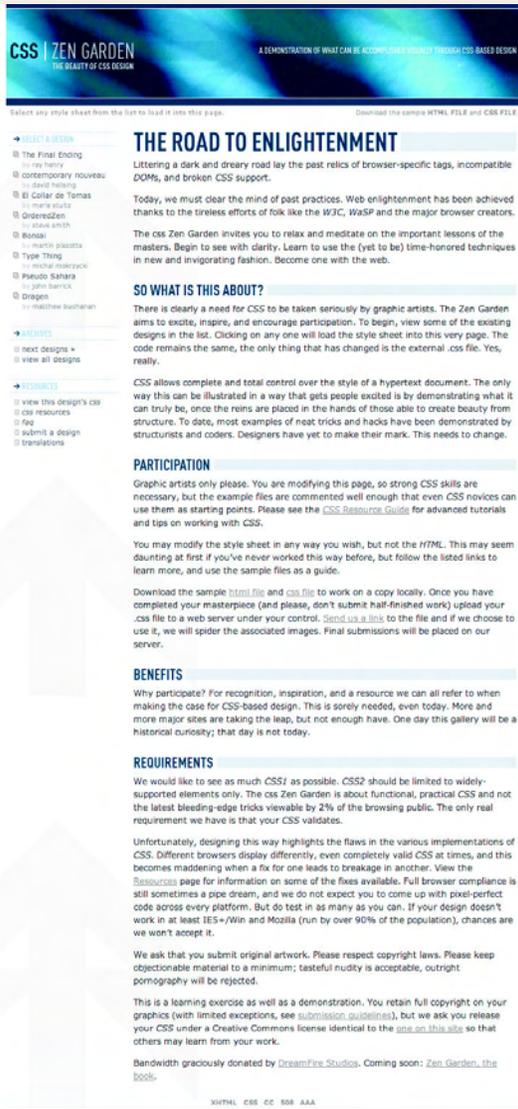
RELATIVE POSITIONIERUNG

Bei der relativen Positionierung wird dagegen kein Element aus dem Dokumentfluss entfernt. Ausgehend von seiner Anfangsposition wird das Element bei der relativen Positionierung versetzt und hinterlässt eine Lücke, hinter der die anderen Elemente so ausgehandelt werden müssen, als ob das Element immer noch dort positioniert wäre (**ABBILDUNGEN 9, 10 und 11**).

Die relative Positionierung ist meist zum Verschieben von Elementen nützlich; ein Element, dessen Ausgangsposition innerhalb eines herkömmlichen Rasters liegt, kann leicht aus der Rasterstruktur hinausgeschoben werden, und Elemente können fein ausgerichtet werden, wenn andere Positionierungsmöglichkeiten nicht gegeben sind. Savarese setzt die relative Positionierung beispielsweise dazu ein, den Drachen am unteren Ende des Designs von einem Ausgangspunkt unterhalb des weißen Inhaltsbereichs an seine endgültige Position auf der linken Seite des Fußes zu verschieben. Die Verwendung der absoluten Positionierung zur Platzierung von Elementen am unteren Ende eines Designs ist viel schwieriger als die Positionierung im oberen Bereich, so dass dieser relative Positionierungsvorgang das Ausrichten erleichtert.

Im Raster und doch flexibel

Ein evolutionärer Designprozess wie der von Cedric Savarese angewendete kann zu chaotischen Ergebnissen führen, wenn die Proportionen nicht mit Sorgfalt festgelegt werden. Das Studieren von Positionierungen und Verhältnissen hilft dabei, Probleme zu verhindern, und der Einsatz eines Rasters während des Designprozesses bietet eine einfache Möglichkeit, ein harmonisches Ergebnis zu erzielen.



Dan Rubin, Designer

www.csszengarden.com/024

Not So Minimal

Allgemeine Überlaufprobleme behandeln

DER TITEL **Not So Minimal** wirft die Frage auf: Nicht so minimal wie was? Hinter diesem Design steht eine Geschichte: **Dan Rubins** erster Beitrag zu Zen Garden mit dem Namen *Grazey* wurde mit dem Ziel maximaler Schlichtheit erstellt.

Rubin legte *Grazey* (**ABBILDUNG 1**) mit einem absichtlich hell gehaltenen Farbschema und entsprechender Schriftartauswahl (FF DIN Condensed und Verdana) vor. Obwohl er die Bedeutung eines guten minimalistischen Designs erkannte, war es das Ziel von Zen Garden, das Stigma der Minimalisierung, das dem CSS-Design anhaftete, zu überwinden. Rubin schmückte *Grazey* aus, und so entstand *Not so Minimal*.

Hinweis

Das ursprüngliche Design mit dem Namen *Grazey* finden Sie unter www.superfluous-banter.org/projects/zengarden/grazay/zen-garden-grazay.html.

Überlaufender Inhalt

Eine der frustrierendsten Angelegenheiten beim Erstellen eines CSS-Layouts ist die Planung im Hinblick auf die variable Beschaffenheit des Webs. Das Problem liegt insbesondere in der Arbeit mit den vordefinierten Breiten und Höhen von CSS, wenn der Umfang des Inhalts vorab noch nicht bekannt ist.

Eine inkonsistente Länge des Inhalts liegt einfach in der Natur des Webs begründet; Seiten können so wenig oder so viel Inhalt wie nötig enthalten: Dynamische Sites, die mit einer Software wie einem Content-Management-System (CMS) betrieben werden, haben die Möglichkeit, Inhalte mit variabler Länge einzufügen, die ungewöhnlich kurz oder unglaublich lang sein können.

Sofern die Länge des Inhalts nicht vom Autor oder dem CMS strikt durchgesetzt wird – was in den meisten Fällen unrealistisch ist –, muss jeder Bereich, der Inhalte enthält und Änderungen unterliegt, sowohl mit kurzen als auch langen Inhalten getestet werden. Textbasierte Inhalte sind weitaus unvorhersehbarer als Bilder, da die Bilder feste vorgegebene Maße haben und daher nicht skalierbar sind, wobei es sich allerdings um keine allgemeingültige Regel handelt. Ein CMS könnte auch mehr als ein Bild einfügen, die Höhe könnte variabel sein oder es könnten mehrere andere Faktoren vorliegen, die zu Größenunterschieden führen.

Heute bieten die meisten Browser dem Benutzer die Möglichkeit, die Größe des Texts auf der Seite zu ändern. Benutzer mit eingeschränkter Sehkraft sind darauf angewiesen. Die Schriftvergrößerung könnte ähnliche Überschneidungsprobleme im Layout verursachen. Bei Textkörperbereichen, die die Probleme mit dem Inhalt variabler Länge lösen, werden zwar theoretisch auch weitere, sich aus der Schriftvergrößerung ergebende Komplikationen vermieden, aber kurze Textelemente wie Navigationslinks sind besonders anfällig. Es ist immer sinnvoll, sicherzustellen, dass eine Seite stabil genug ist, um eine Skalierung um 150 Prozent oder mehr zu überstehen, da es einige Benutzer gibt, die sie mit einer solchen Schriftvergrößerung *betrachten*.

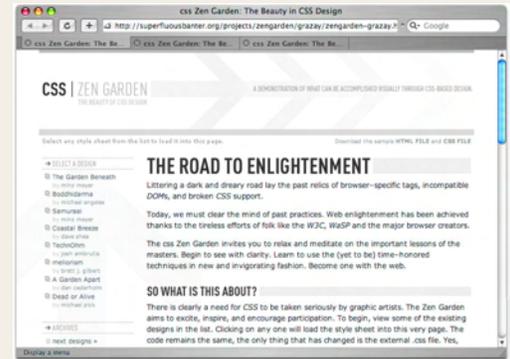


ABBILDUNG 1 Grazay, das ursprüngliche Design von Dan Rubin, zeigt, wie sich Not So Minimal entwickelt hat, um eben nicht so minimal zu erscheinen.

Hinweis

Ein weiteres Problem bei der gleichzeitigen Arbeit mit einem standardbasierten Design und CMS-Software besteht darin, dass viele bekannte Systeme ein ungültiges, mit veralteten Präsentationstags gemischtes Markup erzeugen. Die Gestaltung eines font-Tags ist eine vergebliche Übung und die benutzerdefinierte Kodierung, die notwendig ist, um diese Systeme voll auszunutzen, liegt traditionsgemäß außerhalb vieler Budgets.

Die Situation bessert sich jedoch und die neuere CMS-Software funktioniert bei der Erstellung eines gültigen XHTML/CSS-Codes wesentlich besser. Zu den bekannten Open-Source-CMS-Paketen gehören Drupal (www.drupal.org), Typo 3 (www.typo3.de) und Mambo (www.mambo.com).

Tipp

Eric Meyer hat das Clearing von Floats ausführlich behandelt. Versäumen Sie nicht »Containing Floats« (www.complexspiral.com/publications/containing-floats) zu lesen.

Tipp

Es steht eine Methode für das Clearing von Floats zur Verfügung, wenn der HTML-Code kein `clear`-Element enthält. Zur Herstellung der browserübergreifenden Kompatibilität sind zwar einige nicht intuitive Hacks erforderlich, aber die Methode funktioniert. Siehe »How to Clear Floats Without Structural Markup« (www.positioniseverything.net/easyclearing.html).

Es ist höchst wahrscheinlich, dass Sie sich auf die eine oder andere Weise mit einem horizontal oder vertikal überlaufenden Textkörper auseinander setzen müssen. Anzeichen, auf die Sie achten sollten, sind sich überschneidende Inhalte und große Textkörperbereiche, die weiter unten auf der Seite angezeigt werden, als sie sollten. Ersteres deutet auf ein Überlaufproblem bei der absoluten Positionierung hin, das zweite auf Floats, die für den Bereich, in dem sie sich befinden sollten, zu breit oder zu hoch sind.

Float-Überlauf

Bei dem Attribut `float` handelt es sich genau genommen nicht um ein Layoutwerkzeug. Floats wurden entwickelt, damit Text ein Inline-Element umfließen kann, und waren niemals für den Einsatz als ausgewachsenes Layoutverfahren vorgesehen. Wir *müssen* Floats dennoch verwenden, und zwar aus einem einfachen Grund: Wir haben kein anderes Layoutwerkzeug zur Verfügung.

In ihrer einfachsten Form lassen sich das Attribut `clear` und das Clearing-Konzept als Möglichkeit beschreiben, Elementen mitzuteilen, erst zu beginnen, wenn ein anderes Element endet. Die Überlaufprobleme bei der absoluten Positionierung entstehen meistens, weil die Möglichkeit fehlt, das Umfließen eines positionierten Objekts zu beenden.

Floats sind allerdings unvollkommen und doppelt problematisch, weil viele Browser zusätzliche Schwierigkeiten verursachen.

CLEARING GEHT NICHT OHNE MARKUP

Damit das Clearing ordnungsgemäß vorgenommen werden kann, brauchen Sie mindestens zwei Elemente in Ihrem HTML-Code: eines, das das Attribut `float` übernimmt, und ein zusätzliches Element, um es beenden. Zwei `div`-Elemente können ein Umfließen auf der linken bzw. rechten Seite ermöglichen, in diesem Fall wäre jedoch ein drittes zu deren Clearing erforderlich. Oft ist es möglich, bereits vorhandene Elemente zu nutzen – zum Beispiel für das Clearing eines `h3`-Floats durch ein andernfalls unbesetztes `p`-Tag.

Zuweilen ist es allerdings ein notwendiges Übel, ein zusätzliches leeres Element in Ihr Markup einzufügen, das speziell zum Clearing eines Floats gedacht ist. Ist dies der Fall, empfehlen wir, ein Allzweck-`div`-Element zu verwenden und die Inhalte leer zu lassen, wie das folgende Beispiel zeigt:

```
<div class="clear"></div>
```

Dann kann ein `clear`-Wert auf die Klasse angewendet werden. Das Verfahren ist nicht schön, funktioniert aber, ohne Ihrem HTML-Code zu viele zusätzliche bedeutungslose Strukturelemente hinzuzufügen.

FLOATS BENÖTIGEN PRÄZISION

Was geschieht, wenn Sie ein 40 Liter-Aquarium bis zum Rand füllen und dann ein Schöpfelchen Sand hineingeben? Es ist natürlich nicht genügend Platz dafür vorhanden, so dass etwas Wasser überläuft.

Floats unterliegen ähnlichen Einschränkungen, was meist zu einem horizontalen Überlauf führt. Wenn zwei Floats, die 50 Prozent der Bildschirmbreite einnehmen, nebeneinander positioniert werden und für einen davon ein zusätzlicher Rand von 100 Pixeln festgelegt wird, wird der andere Float hinausgeschoben und fällt damit unter den ersten Float. Eine vorherige Berechnung der Breiten ist wichtig.

Allerdings unterstützt der Microsoft Internet Explorer für Windows die Attribute `width` und `height` nicht so, wie es die CSS-Spezifikation empfiehlt. Was *vermutlich* passiert, ist, dass ein Element, dessen Inhalt breiter ist als sein `width`-Wert, den Inhalt entweder umformatiert oder, falls dies nicht geschieht, die Breite beibehält und erlaubt, dass der Inhalt über seine Grenzen hinausläuft. Im Ergebnis bleiben die Floats unbeeinflusst, selbst wenn der Inhalt beginnt, andere Elemente zu überschneiden.



ABBILDUNG 2 Wenn die Float-Elemente zu breit für den ihnen zugewiesenen Bereich sind, können eines oder mehrere dieser Elemente nach außen gedrückt werden und somit unterhalb der restlichen Elemente erscheinen.

Auf der anderen Seite lässt der Internet Explorer zu, dass sich der Inhalt über die benötigte Breite erstreckt. Sobald ein Float nur ein Pixel breiter ist, als er sein sollte, drängt er andere Float-Elemente hinaus (**ABBILDUNG 2**). Es gibt einige Möglichkeiten, dieses Problem zu beheben.

Den Bereich verbreitern

Wenn Sie die Möglichkeit haben, den Bereich, in dem sich der Float befindet, zu verbreitern, kann das Problem in vielen Fällen gelöst werden. Obwohl diese Lösung nicht optimal ist, sollten Sie diese Methode zuerst in Erwägung ziehen, da sie relativ leicht zu implementieren ist.

Die Breite des Inhalts einschränken (Kursivschrift)

Wenn es möglich wäre, sicherzustellen, dass der Inhalt keinen horizontalen Überlauf verursacht, ließe sich das Problem vermeiden. Dies ist normalerweise eine unrealistische Lösung, da das Problem nicht existieren würde, wenn der Inhalt nicht erst einmal zu breit wäre.

Ein besonders dummer Fehler, auf den Sie aufpassen müssen, liegt vor, wenn Inhalt zwar die richtige Breite zu haben scheint, aber dennoch weiterhin Schwierigkeiten bereitet, besonders im Internet Explorer für Windows. Wenn Sie auf dieses Problem stoßen und Text innerhalb des Float-Bereichs kursiv dargestellt wird, versuchen Sie, die Kursivschrift zu entfernen, und prüfen Sie, ob das Problem damit behoben ist. Kursiver Text kann im Internet Explorer einen horizontalen Überlauf verursachen und falls das Float-Problem nach dem Deaktivieren dieses Effekts gelöst ist, haben Sie den Übeltäter gefunden. Eigentlich sollte die Textformatierung ein Layout nicht auf diese Weise beeinflussen, aber hier ist es der Fall.

Überlaufenden Inhalt einschränken

Im nächsten Abschnitt, der sich dem vertikalen Überlauf widmet, werden wir das Attribut `overflow` erörtern, das aber auch zur Begrenzung des horizontalen Überlaufs eingesetzt werden kann. Lösungen, bei denen `overflow` auf diese Weise verwendet wird, sind oft nicht sehr elegant, so dass wir empfehlen, dabei Vorsicht walten zu lassen.

Überläufe bei absoluter Positionierung

Bei den bei der absoluten Positionierung auftretenden Problemen handelt es sich normalerweise um Schwierigkeiten mit dem vertikalen Überlauf. Die Ausnahme sind Fälle, in denen ein positioniertes Element eine bestimmte Breite hat, aber nicht genügend Platz zur Verfügung gestellt wurde, um dieser Breite Rechnung zu tragen (**ABBILDUNG 3**).

VERTIKALER ÜBERLAUF

Die Behandlung des vertikalen Überlaufs ist weitaus häufiger erforderlich. Das Problem ist einfach: Im Gegensatz zu Float-Elementen kann bei absolut positionierten Elementen kein Clearing erfolgen. Wurde die Positionierung erst einmal angewendet, wird das Element aus dem Dokumentfluss entfernt und hat anschließend keinen Einfluss mehr auf die umliegenden Elemente.

Dieses Thema kommt häufig zur Sprache, wenn es um die Behandlung der Seiten-Footer geht – eine absolut positionierte Spalte kann länger als andere Spalten werden, die die Position des Footers beeinflussen. Die positionierte Spalte läuft vertikal direkt über den Anfang des Footers hinweg und weiter (**ABBILDUNG 4**). Leider gibt es wirklich keine narrensichere Lösung; es stehen lediglich Hilfslösungen zur Verfügung.

Floats einsetzen

Die effektivste Hilfslösung besteht darin, Floats anstelle der absoluten Positionierung zu verwenden und die Vorteile des von ihnen angebotenen Clearings zu nutzen. Dies ist natürlich nicht immer möglich, aber wenn ein Layout mit Floats dadurch funktionieren könnte, lohnt es sich, diese Methode auszuprobieren.



ABBILDUNG 3 Das 150 Pixel breite Element #linkList ist zu breit für den ihm zugewiesenen 100 Pixel umfassenden Bereich.

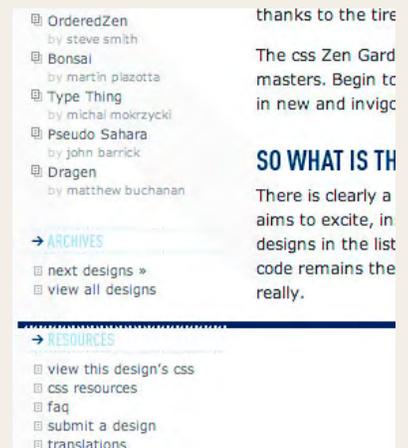


ABBILDUNG 4 Wenn der Textkörper kürzer wäre, würde die absolut positionierte Bildlaufleiste von Not So Minimal die Fußleiste überschneiden.

Gestalten Sie um das Problem herum

Als Alternative könnte dem absolut positionierten Element der Rest der Seite in vertikaler Richtung als freier Bereich zugestanden werden. Überschneidet sich das Element mit dem Footer am Seitenende, besteht ein möglicher Ausweg darin, den Footer so umzugestalten, dass ihm ein Rand in der Breite des Elements zugewiesen wird. Werden unter dem positionierten Element keine anderen Elemente platziert, können keine Überschneidungen auftreten, so dass auch keine Probleme entstehen.

Überläufe einschränken

Anstatt zu versuchen, die Länge des Inhalts zu steuern, ist es weitaus günstiger, die in CSS integrierten Funktionen zur automatischen Überlaufsteuerung auszuprobieren. Das Attribut `overflow` wurde theoretisch aus genau diesem Grund geschaffen, stellt aber nicht die erhoffte Wunderwaffe dar. Wenn Sie den `height`-Wert eines Elements auf `150px` begrenzen und den `overflow`-Wert auf `auto` setzen, wird die Höhe auf genau 150 Pixel gestutzt und dem Benutzer eine Bildlaufleiste angezeigt, damit er sich den über 150 Pixel hinausgehenden Inhalt ansehen kann.

Zu viele Bildlaufleisten auf einer Seite verärgern den Betrachter allerdings schnell. Als Alternative können Sie den `overflow`-Wert auf `hidden` setzen und damit den gesamten über den Schnittpunkt hinausgehenden Inhalt verbergen. In den meisten Fällen ist es nicht sinnvoll, den Inhalt auf diese Weise zu verbergen, da versteckte Inhalte für den Betrachter nutzlos sind, aber in seltenen Fällen kann es genau die richtige Methode zur Überlaufbehandlung sein.

Skripts machen den Weg frei

Wenn Sie nichts dagegen haben, als Ergänzung zu Ihrem CSS-Code in geringem Maße JavaScript einzusetzen, stehen Ihnen Skripts zur Bekämpfung dieses Problems zur Verfügung. Einer der Mitwirkenden an Zen Garden, Shaun Inman, hat ein nützliches Verfahren für das Clearing absolut positionierter Elemente hergestellt, das Ähnlichkeiten mit dem Clearing von Floats aufweist. Dieses Skript steht Ihnen unter »Absolute Clearance« (www.shauninman.com/mentary/past/absolute_clerance.php) zur Verfügung.

Verwenden Sie Gevierte statt PX-Werten

Dieser Tipp gilt nur, wenn Überschneidungen durch Schriftvergrößerungen entstehen; Überschneidungen in dynamischen Textkörpern lassen sich damit nicht beheben. Die absolute Positionierung mit der Einheit px stellt eine hervorragende Möglichkeit zur exakten Steuerung dar, bietet jedoch keine Flexibilität – ist ein Element erst einmal positioniert, dann verbleibt es an der betreffenden Stelle. Durch die Verwendung von Gevierten für Attribute wie `top` und `left` wird die Position des Elements jedoch etwas flexibler: Sie wird an die vom Benutzer gewählte Schriftgröße angepasst. Weist man Stapелеlementen Gevierte als Wert zu, ändert sich der Anfang des nächsten Elements, sobald das Ende eines Elements die Position wechselt. Diese Methode ist ungenau, funktioniert aber in vielen Situationen.

Den Überlauf gestalten

Die absolut positionierte linke Bildlaufleiste in Not So Minimal dient der sicheren Vermeidung des Überlaufproblems, da der Textkörper auf der rechten Seite viel länger ist. Handelte es sich bei Zen Garden um eine größere Site mit mehreren Inhaltsseiten, so wären möglicherweise verschiedene Seiten mit einer längeren Bildlaufleiste ausgestattet und Seiten mit viel kürzerem Inhalt vorhanden. Höchstwahrscheinlich wäre die zuvor in **ABBILDUNG 4** gezeigte Überschneidung aufgetreten und hätte zusätzliche Schritte zur Behandlung des Überlaufs erforderlich gemacht.

Obwohl Entscheidungen im Hinblick auf das Design nicht von technischen Aspekten gelenkt sein sollten, stellt der Überlauf ein wesentliches Problem beim Aufbau größerer Sites dar. Wenn Sie es einfach dadurch vermeiden können, dass Sie Ihr Design um das Problem herum gestalten, würden Ihnen die technischen Einschränkungen in den Verfahren zur Überlaufbehandlung nicht so frustrierend ungenügend erscheinen.