



Dave Shea
Molly E. Holzschlag

Zen und die Kunst des CSS-Designs

Inspiration und
Umsetzung

www.csszengarden.com
2., aktualisierte Auflage

 ADDISON-WESLEY

The Addison-Wesley logo is located in the bottom left corner. It features a red stylized 'A' symbol composed of three triangles, followed by the text 'ADDISON-WESLEY' in a bold, black, sans-serif font.



White Lily, Seite 120



Not So Minimal, Seite 144

Layout

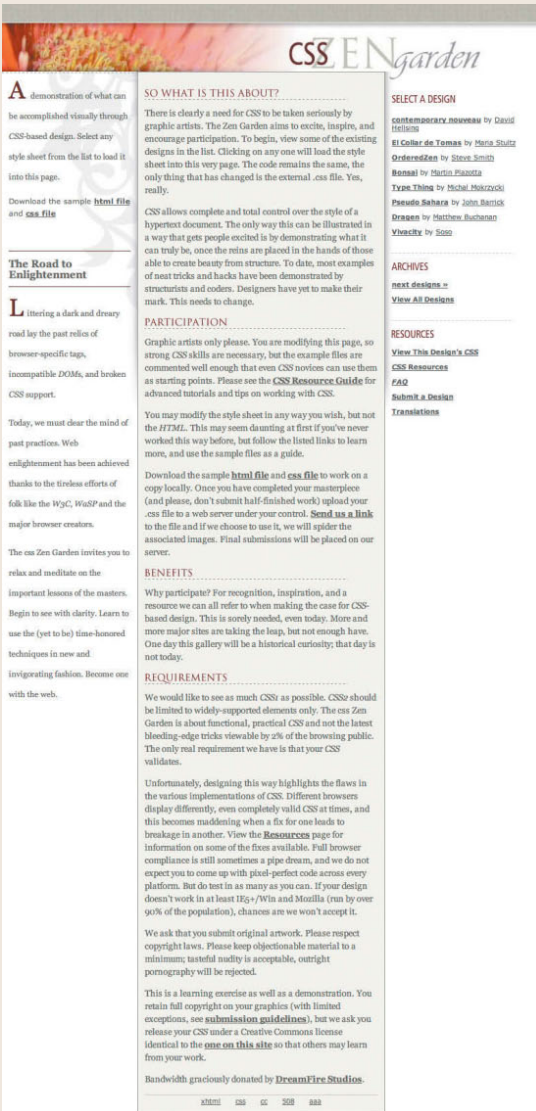
DIE SEITENGESTALTUNG mit Hilfe von CSS erfordert eine Kombination aus technischem Wissen und Kreativität. Unabhängig davon, ob Sie ein festes oder ein fließendes Layout wählen und ob Sie Ihre Designs linksbündig oder zentriert ausrichten, müssen Sie eine feste Vorstellung von der Funktionsweise der CSS-Layouts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

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.



CSS Zen Garden

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 is 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 structurists 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 as 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 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 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](#).

css zen garden

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**).

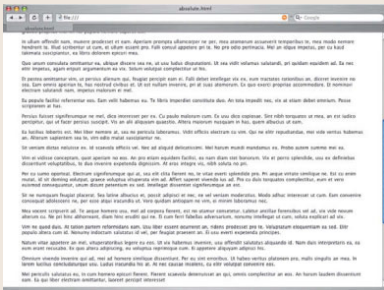


ABBILDUNG 3 Regulärer Text

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 folgende Markup 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>
```



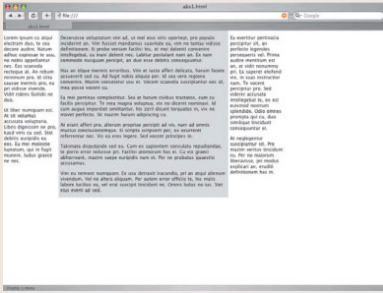
ABBILDUNG 4 Wenn ein Foto hinzugefügt wird, erscheint eine Lücke im Text.



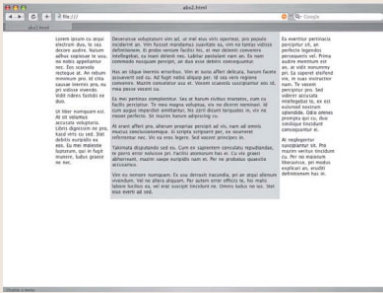
ABBILDUNG 5 Der Text wird fließend um das Foto angeordnet, um die Lücke zu schließen.

Tipp

Merkwürdigerweise verdoppelt der Internet Explorer für Windows die horizontalen Ränder einiger Float-Elemente, was zu Verwirrung 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)).



ABILDUNG 6 Einfaches dreispaltiges Layout



ABILDUNG 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 (**ABILDUNG 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 **ABILDUNG 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.

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!

css Zen Garden

the beauty of CSS design



(select a design:)
CONTEMPORARY NOUVEAU
by David Heilberg

EL COLLAR DE TOMAS
by Maria Stultz

ORDEBREDEN
by Steve Smith

BONSAI
by Martin Flaxotta

TYPE THING
by Michal Mokrzycki

PSEUDO SAHARA
by John Barrick

DRAGEN
by Matthew Buchanan

VIVACITY
by Sozo



(archives:)
next designs »
View All Designs



(resources:)
View This Design's CSS

CSS Resources

FAQ

Submit a Design

Translations



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 part relics of browser-specific tags, incompatible DOCS, 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, WAF 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 is 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 structurists 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 5% of the browsing public. The only real requirement we have is that your CSS2 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 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](#). Coming soon: [Zen Garden, the book](#).

{ xhtml } { css } { oc } { 500 } { aaa }

Festgelegte und fließende Layouts

Es gibt zwei herkömmliche Arten von Weblayouts: *feste* oder *fließende*. 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.

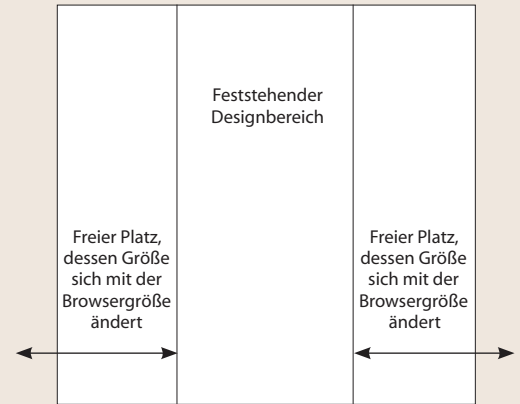


ABBILDUNG 1 Darstellung des Verhaltens bei fester Breite

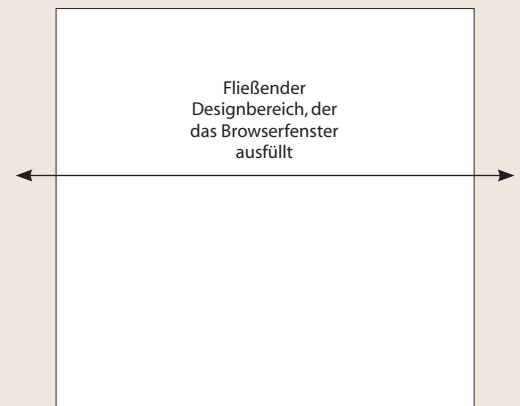


ABBILDUNG 2 Darstellung des Verhaltens bei einem fließenden Design

In den letzten Jahren ist die Beliebtheit fester Designs wieder gestiegen, insbesondere unter den auf Standards zurückgreifenden Designern. Zahlreiche führende Fachleute verwenden 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 der Schmetterlingsbilder 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 ab der Version 6 für Windows, sofern Sie sich im Standard-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.

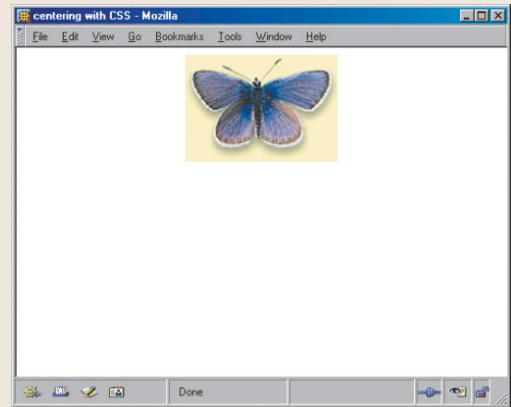


ABBILDUNG 3 Zentrieren eines Elements mit automatischen Rändern

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+, Standard-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+, 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 Zentrieremethode 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.

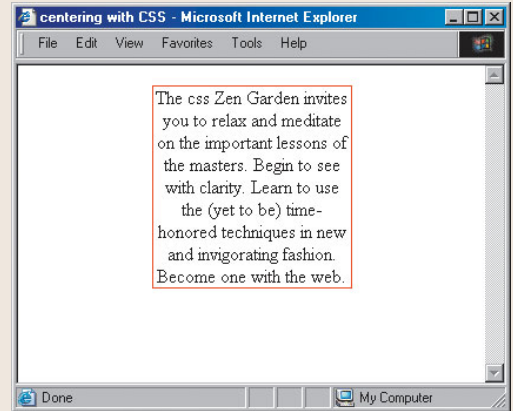


ABBILDUNG 4 Zentrieren mit Hilfe der Eigenschaft `text-align` im Internet Explorer. Beachten Sie, wie der Text des gesamten Absatzes nun innerhalb des `div`-Containers zentriert ist.

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.

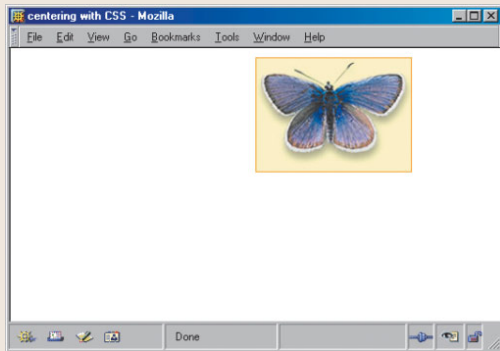


ABBILDUNG 5 Absolute Positionierung des Containerblocks mit einem Abstand von 50 Prozent vom linken Rand

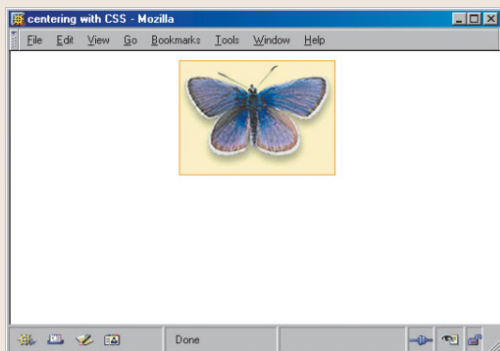


ABBILDUNG 6 Fügt man einen negativen Rand hinzu, der halb so breit ist wie die gesamte Breite des Elements, wird das Feld horizontal zentriert.

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 *Halfte* 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 bedienen 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.

White Lily

Prinzipien und Prozesse zum Gestalten wirkungsvoller Layouts

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.

The screenshot shows the Zen Garden website interface. At the top, there's a header with a white lily image and the text 'ZEN GARDEN The Beauty of CSS Design'. Below the header, there's a navigation bar with links for 'html', 'css', 'js', '508', and 'aaa'. The main content area is divided into two columns. The left column contains a 'Select a Design' section with a list of designs including 'The Final Ending', 'contemporary nouveau', 'El Collar de Tomas', 'OrderedZen', 'Bonsai', 'Type Thing', 'Pseudo Sahara', and 'Dragen'. Below this is an 'Archives' section with links for 'next designs', 'View All Designs', and 'Resources' with links for 'View This Design's CSS', 'CSS Resources', 'FAQ', 'Submit a Design', and 'Translations'. The right column features the title 'The Road to Enlightenment' and a paragraph of text. Below this is a section titled 'So What is This About?' with more text. Further down is a 'Participation' section, followed by 'Benefits' and 'Requirements' sections, each with text explaining the project's goals and requirements.

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.

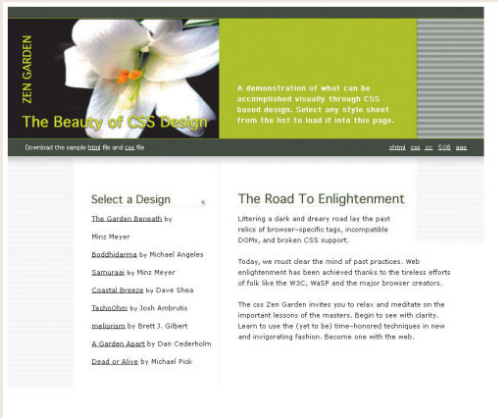


ABBILDUNG 1 Ein alternativer Stil für White Lily



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 zweispaltige 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.

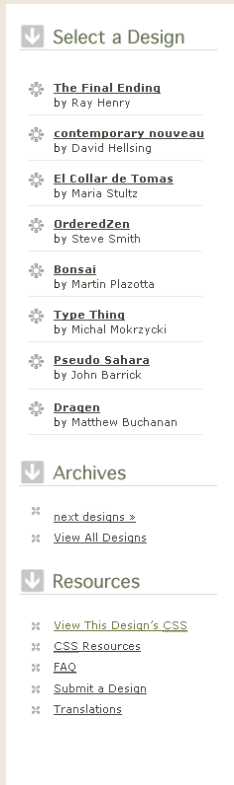


ABBILDUNG 5 Die Navigationslinks von White Lily

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 ausgeprägtes 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.

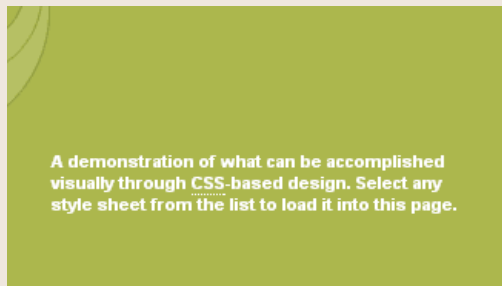


ABBILDUNG 6 Die Aufgabe der Seite in einem eigenen Bereich innerhalb des Layouts zu verdeutlichen, trägt zur leichten Verständlichkeit des Zwecks der Seite bei.



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 sind. 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 Titelelements:

```
<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 Designterteams 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.

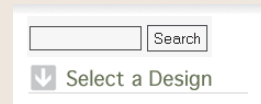


ABBILDUNG 11 Wenn in diesem Layout eine Suchfunktion zur Verfügung gestellt werden sollte, könnte diese möglicherweise über den Link innerhalb des Navigationsbereichs des Layouts erscheinen.



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.

css.zengarden

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 beauty of css design

the road to enlightenment

Libering a dark and dreary road by the path rocks of knowle-specific bags, 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, W3P and the major browser creators.

The css Zen Garden invites you to learn 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 inspiring fashions. Become one with the web.

as what it is, 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 achieved is 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.

participate

Graphic artists only please. You are reading 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 update the associated images. Final submissions will be placed on our server.

benefit

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 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 it as many as you can. If your design doesn't work in at least ES+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.

Headerwidth graciously donated by [BrandFire Studios](#).

about css zen garden

select a design

- by [brian williams](#)
- by [baccali](#)
- by [chris moerell](#)
- by [hastings](#)
- by [michaela maria sampl](#)
- by [skyroute](#)
- by [aesthetestrait](#)
- by [centralred](#)
- by [john oton](#)
- by [lita's lifestyle](#)
- by [emiliano pennisi](#)
- by [mediaspace](#)
- by [emiliano pennisi](#)
- by [pleasant day](#)
- by [kyle jones](#)

next designs »
view all designs

resources

- view this design's [css resources](#)
- [faq](#)
- [submit a design](#)
- [translations](#)