

[SELFHTML aktuell](#) [Artikel](#) [CSS](#)



Footer aller Art - feststehende Elemente realisieren



- ↓ [Dennis Riehle](#)
- ↓ [Was ist ein Footer?](#)
- ↓ [Einfacher Footer](#)
- ↓ [Feststehender Footer](#)
- ↓ [Feststehender Footer für den Internet Explorer](#)
- ↓ [Kombination: Footer für alle](#)
- ↓ [Feststehende Kopf- und Fußzeile](#)
- ↓ [Alternative Positionierung mit feststehender Navigation](#)
- ↓ [Vollständiges Layoutbeispiel](#)
- ↓ [Abschließende Zusammenfassung](#)



Dennis Riehle

E-Mail: [✉ selfhtml@riehle-web.com](mailto:selfhtml@riehle-web.com)

Homepage-URL: <http://tutorial.riehle-web.com/>

Bei Fragen zu diesem Beitrag bitte den Autor des Beitrags kontaktieren!



Was ist ein Footer?

Das Wort Footer (*engl. foot = Fuß*) bezeichnet eine ↓ [Fußzeile](#), wie sie auch die SELFHTML-Seiten aufweisen.

Es stellt sich die Frage, welche Gründe dafür sprechen, Webseiten mit einem Footer zu versehen. Dies sind beispielsweise:

- Ein Footer schließt eine Seite sowohl optisch als auch inhaltlich ab. Insbesondere, wenn Elemente vorhanden sind, die sich vertikal über die gesamte Seite erstrecken, ist ein optischer Abschluss interessant.
- Bei fehlendem Footer kann der Eindruck entstehen, die Seite ginge nach unten noch weiter, so dass der Besucher vergeblich weiterscrollt und erst nach einem zweiten Blick auf den Scrollbalken feststellt, dass er sich bereits am Seitenende befindet.
- Darüber hinaus sind Footer geeignet, Elemente unterzubringen, die sich auf jeder Seite befinden sollen (z.B. ein Urheberrechtshinweis) oder unter Umständen sogar müssen (z.B. ein Link zum Impressum).



Einfacher Footer

Einfache Footer werden im SELFHTML-Kapitel [CSS-basierte Layouts](#) dargestellt, der Vollständigkeit halber hier ein kleines Beispiel:

Beispiel:

☐ Anzeigebeispiel: So sieht's aus

```
#footer {
  clear:left;
  margin-top:1em;
  background:#efefef;
  text-align:center;
  padding:10px;
}
```


Erläuterung:

Gegeben sei an dieser Stelle ein Element mit einem  **id-Attribut** und dessen Wert `footer`. Diesem Element wird die CSS-Eigenschaft  **clear** mit dem Wert `left` bzw. je nach Aufbau der Seite auch `right` oder `both` zugewiesen, wodurch ein eventuell zuvor mittels  **float** geänderte Elementfluss beendet wird und das Element die Seite nach unten hin abschließen kann.

Die weiteren Angaben dienen lediglich der optischen Aufwertung des Footers im Beispiel.



Feststehender Footer

Eine erweiterte Form des vorhergehenden Beispiels ist ein Footer, der am unteren Bildrand feststeht und damit immer zu sehen ist, unabhängig davon, wie die Seite gescrollt wird. Moderne Browser kennen dazu die CSS-Eigenschaft  **position** mit dem Wert `fixed` - dieser wird jedoch vom Internet Explorer bis einschließlich Version 6 noch nicht interpretiert, weswegen das folgende Beispiel in diesem Browser nicht wie gewünscht funktionieren wird.

Beispiel:

☐ Anzeigebeispiel: So sieht's aus



```
#footer {
  position:fixed;
  bottom:0px;
}

#content_container {
  padding-bottom:3em;
}
```

Erläuterung:

Mittels `position:fixed` wird das Element mit der ID `footer` als feststehendes Element definiert, mittels `bottom:0px`; wird es 0 Pixel von der Unterkante des Darstellungsbereiches entfernt, also ganz unten positioniert.

Beachten Sie:

Sie sollten für den Bereich `content_container` noch einen unteren  **Innen-** oder  **Außenabstand** definieren, da sonst die Fußzeile den unteren Bereich des Elements überdeckt!



Feststehender Footer für den Internet Explorer

Ziel ist es nun, eine Möglichkeit zu finden, wie sich ein feststehender Footer auch im Internet Explorer realisieren lässt, sozusagen ein Work-Around für `position:fixed`.

Man kann zwei Elemente ("Container") untereinander setzen, wobei das obere Element den kompletten Inhalt enthält und das untere den Footer. Daraus ergibt sich folgende HTML-Struktur:

Beispiel:

Anzeigebeispiel: So sieht's aus

```
<div id="content_container">

    <!-- Hier folgt der komplette Inhalt der Seite inklusive Kopfzeile -->
    <h1 id="header">Kleine Seite mit Footer (3)</h1>
    <!-- usw. -->

</div>

<div id="footer">
    <b>Und hier ist der langerwartete Footer!</b>
</div>
```

Der Workaround besteht nun darin, für das erste Element Scrollbalken zu definieren und es auf eine bestimmte Höhe zu beschränken. Dadurch hat unter diesem Element noch ein ganz normales weiteres Element, hier der Footer, Platz.

```
html, body {
    margin:0;
    padding:0;
    height:100%;
    overflow:hidden;
}

#content_container {
    height:95%;
    overflow:auto;
}

#footer {
    background:#efefef;
    height:5%;
    text-align:center;
    padding-top:1%;
}
```

Erläuterung:

Zuerst müssen `html` und `body` auf eine Höhe von 100% gebracht werden. Dies ist nötig, damit sich nachfolgende

Prozent-Angaben an dieser Basis orientieren. Mittels `overflow:hidden` verhindert man, dass der Browser Scrollbalken darstellt, welche nicht benötigt werden - so würde z.B. der Internet Explorer ausgegraute Scrollbalken darstellen. Sowohl `margin` als auch `padding` wurden auf 0 gesetzt, um Innen- und Außenabstände zu entfernen.

Für den `content_container` definiert man beispielsweise eine Höhe von 95% und weist ihm die CSS-Deklaration `overflow:auto` zu. Theoretisch könnte man auch `overflow:scroll` einsetzen, dann bieten aber die meisten Browser auch einen (unerwünschten) horizontalen Scrollbalken an.

Dem Footer muss nun lediglich noch die verbliebene Höhe zugewiesen werden, in diesem Beispiel sind es 5%.

Nachteil dieser Methode:

- So, wie der Code hier notiert wurde, ist es nicht möglich, absolute Werte für die Höhe des Footers zu vergeben, dazu müsste man die Elemente anders positionieren, lesen Sie diesbezüglich den Abschnitt [Alternative Positionierung - feste Navigation](#).
- In Gecko-basierten Browsern (Mozilla, Firefox, neue Versionen von Netscape) ist es nicht möglich, das obere `div`-Element per Mausrad zu scrollen - dem Besucher bleibt also nichts anderes übrig, als den Scrollbalken herkömmlich zu bedienen.
- Konqueror 3.2.1 unter Linux stellt bei diesem Beispiel zwei vertikale Scrollbalken dar.
- Zusätzlich mag es als unschön bzw. ungewohnt empfunden werden, dass der Scrollbalken nicht über die komplette Höhe der Seite verläuft.

↑ ↓

Kombination: Ein Footer für alle

Wie Sie bestimmt schon gemerkt haben, hatten beide Beispiele ihre Nachteile: Ersteres funktionierte nicht im Internet Explorer, letzteres funktionierte zwar im Internet Explorer, dafür aber in anderen Browsern nicht wie gewollt.

Was wir also bräuchten, wäre ein Kombination aus beiden Möglichkeiten, so dass der Internet Explorer speziell auf ihn zugeschnittene Formatierungen erhält, alle hinreichend CSS-fähigen Browser dadurch jedoch nicht durcheinandergebracht werden. Natürlich könnte man serverseitig den User-Agent-String untersuchen und entsprechend eine ganz andere Seite ausgeben, dies hat bei fehlerhafter Implementierung jedoch den Nachteil, dass z.B. Opera oder Firefox, wenn sie sich als Internet Explorer ausgeben, das falsche Stylesheet referenzieren. Auch eine clientseitige Unterscheidung mit JavaScript kommt nicht in Frage, da JavaScript deaktiviert sein kann und somit auch diese Unterscheidung nicht zuverlässig funktionieren würde.

Es werden also sogenannte `CSS-Browserweichen` ("CSS-Hacks") benötigt.

Beispiel:

[Anzeigebeispiel: So sieht's aus](#)

```
/* Angaben für alle standardkonformen Browser */

body {
    margin:0;
    padding:0;
}

#content_container {
    padding-bottom:3em;
}
```

```
#header {
  margin: 0 0 0.7em;
  background:#efefef;
  text-align:center;
  padding:10px;
}

#footer {
  position:fixed;
  bottom:0px;
  background:#efefef;
  text-align:center;
  padding:10px;
  width:100%;
}

/* Angaben nur für den Internet Explorer mittels Star-HTML-Hack */

* html, * html body {
  margin:0;
  padding:0;
  height:100%;
  overflow:hidden;
}

* html #content_container {
  padding:0;
  height:95%;
  overflow:auto;
}

* html #footer {
  background:#efefef;
  height:5%;
  text-align:center;
  padding-top:1%;
}
```

Erläuterung:

Zuerst wurden die Deklarationen aus dem Beispiel [↑ "Feststehender Footer"](#) übernommen. Anschließend werden unter Nutzung des [☰ Star-HTML-Hacks](#) sämtliche Angaben aus dem Beispiel [↑ "Feststehender Footer für den Internet Explorer"](#) hinzugefügt.

Zunächst werden für alle Browser diverse CSS-Eigenschaften gesetzt. Im Internet Explorer, der auch Selektoren der Form `* html` (Star-HTML-Hack) interpretiert, werden diese Angaben überschrieben.

Dieses Beispiel wurde unter Windows mit Mozilla 1.7, Firefox 1, Opera 7, Netscape 7 und natürlich dem Internet Explorer 6 erfolgreich getestet (Im IE 5.0 funktioniert es ebenfalls, jedoch lässt sich das Scrollrad der Maus nicht nutzen). Ebenso problemlos hat es unter Linux im Konqueror 3.2 und Mozilla 1.7 funktioniert.

↑ ↓

Feststehende Kopf- und Fußzeile

Wenn Sie nun vielleicht noch gerne eine feste Kopfzeile definieren möchten, damit die Seite sowohl nach oben als

auch nach unten hin stetig "abgeschlossen" ist, lässt sich das auf die gleiche Art und Weise realisieren wie der Footer. Um bei der obigen HTML-Struktur zu bleiben, muss man das h1-Element lediglich aus dem Container herausnehmen und davor setzen. Folgende CSS-Deklarationen werden ferner benötigt:

Beispiel:

☐ **Anzeigebeispiel: So sieht's aus**

```
#header {
  position:fixed;
  top:0px; left:0px; right:0px;
  background:#efefef;
  text-align:center;
  padding:10px;
}

/* Und für den Internet Explorer */

* html #header {
  margin:0;
  height:10%;
}
```

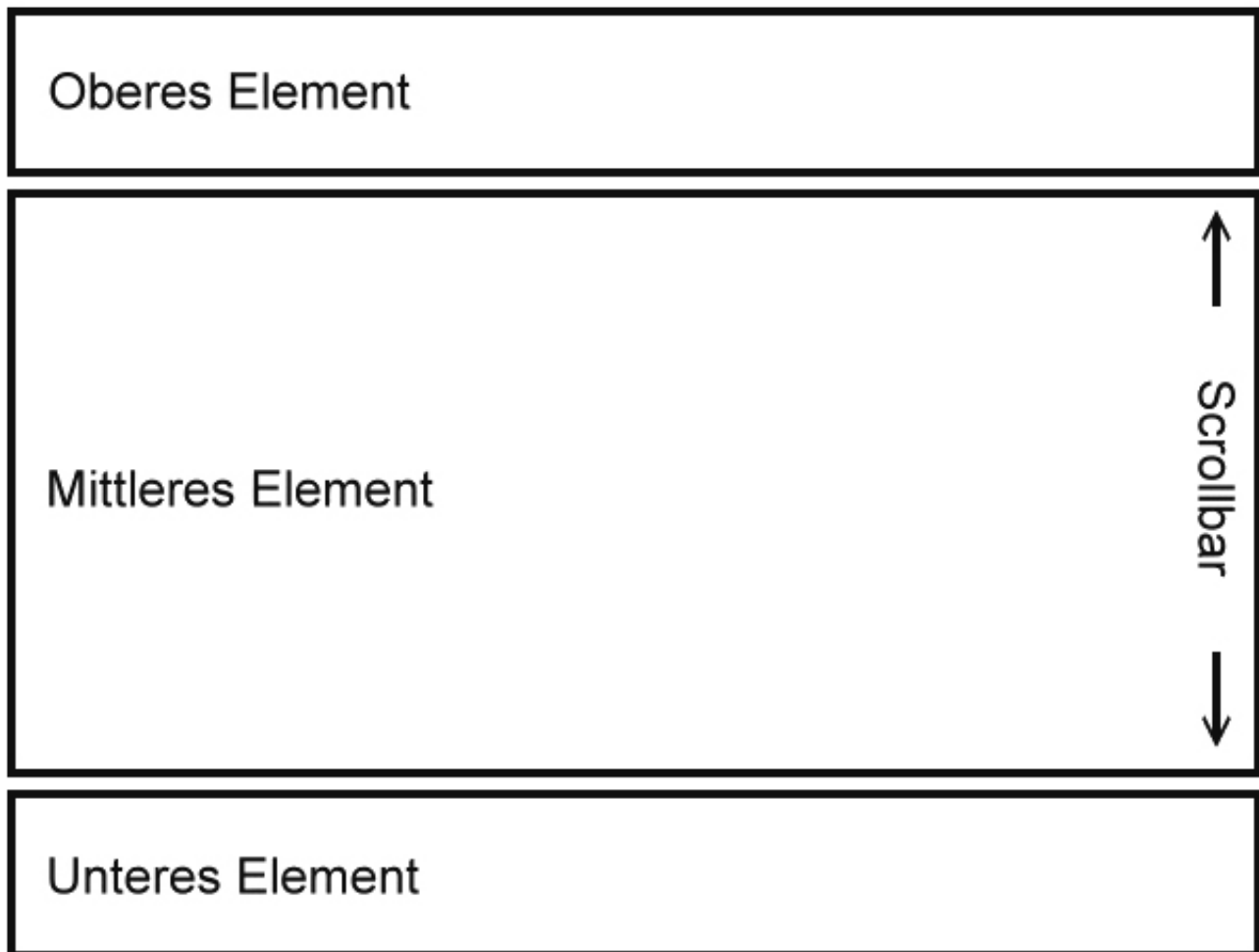
Erläuterung:

Die Funktionsweise ist beinahe identisch mit dem vorhergegangenen Beispiel - lediglich der `content_container` ist von der Höhe her gemäß dem obigen Beispiel auf 85% anzupassen.

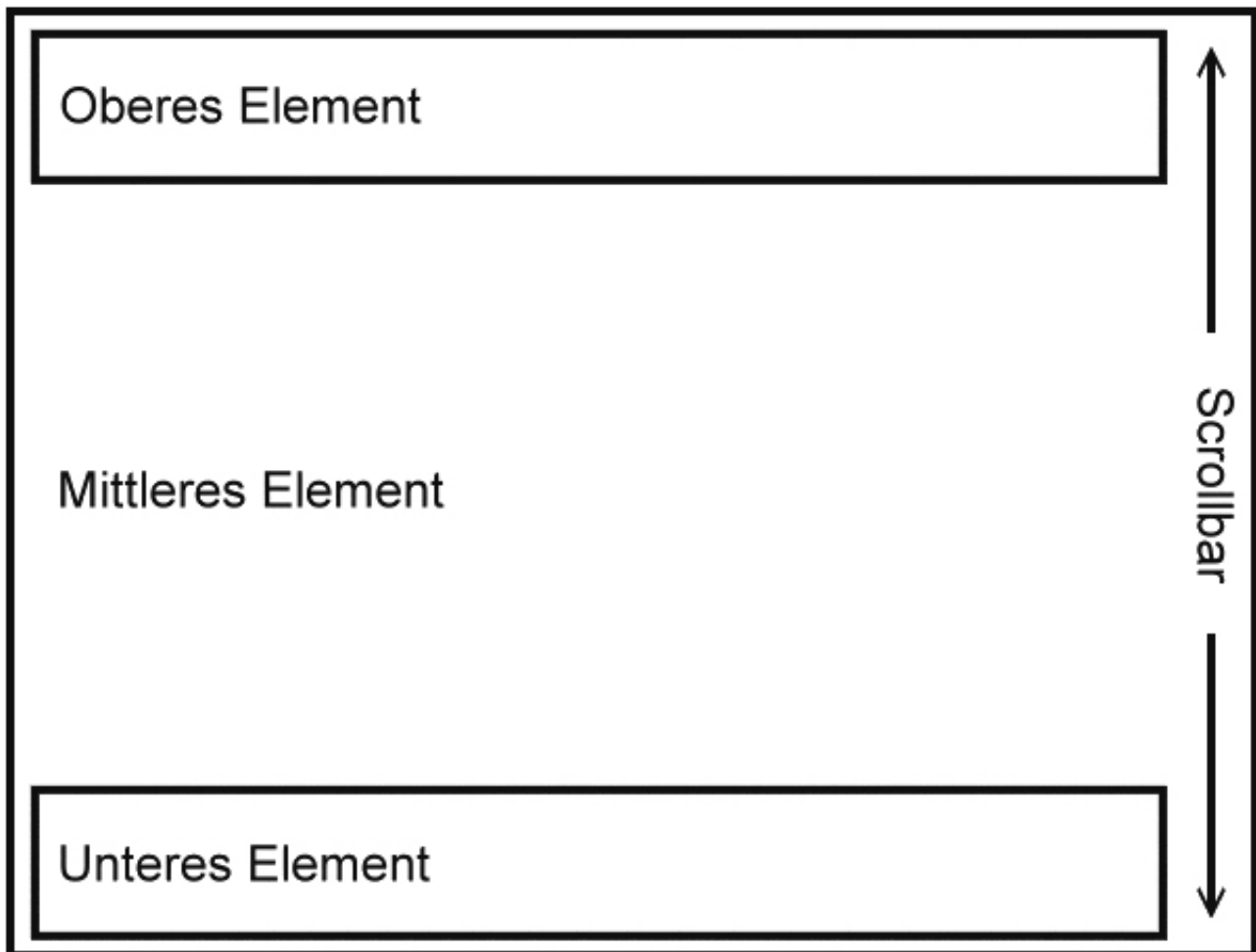


Alternative Positionierung mit fester Navigation

Bisher wurden die Elemente übereinander angeordnet, so dass lediglich in der Mitte ein Element gescrollt wurde:



Für einen Header und einen Footer ist das ausreichend. Möchte man aber noch weitere Elemente fixieren, z.B. die Navigation, so wird dies dadurch erschwert. Aus diesem Grund wird noch eine weitere Möglichkeit vorgestellt, mit der es möglich ist, noch weitere Elemente zusätzlich zur Kopf- und Fußzeile zu fixieren.



In diesem Fall wird das mittlere Element, also das Element mit dem eigentlichen Inhalt über die gesamte Größe des Browserfensters ausgedehnt und mit Scrollbalken versehen. Über diesem Element lassen sich dann z.B. eine Kopfzelle, eine Fußzelle oder auch eine Navigation absolut positionieren.

Folgende HTML-Struktur ist hierfür nötig:

Beispiel:

```
<div id="header_container">
  <h1 id="header">Seite mit Header und Footer (2)</h1>
</div>

<ul id="menu">
  <!-- Hier folgt das Menü -->
</ul>

<div id="content">
  <!-- Hier folgt der Inhalt -->
</div>

<div id="footer_container">
  <div id="footer">
    <b>Und hier ist der endgültige Footer!</b>
  </div>
</div>
```


Erläuterung:

Was auf den ersten Blick fragwürdig erscheint, sind die zusätzlichen Container-Elemente für den Header und den Footer. Diese werden für den Internet Explorer benötigt, da dieser den Header und den Footer bei einer Breite von 100% über den Scrollbalken legt. Dem kann man entgegenwirken, indem man den Container 100% breit macht, einen Innenabstand nach rechts von 16px (entspricht der Standard-Breite des Scrollbalkens im Internet Explorer) zuweist und darin Header bzw. Footer platziert.

Für das zu positionierende Menü wird kein Container benötigt, da das Menü nicht über die komplette Breite des Fensters gehen, sondern nur 9em breit sein soll.

Beispiel:

Anzeigebeispiel: So sieht's aus

Abschließend noch das komplette Stylesheet zu obigem Anzeigebeispiel:

```
body {
    position:absolute;
    background:#FFFFFF;
}

html, body {
    top:0; left:0; right:0;
    min-height:100%;
    margin:0;
    padding:0;
    width:100%;
}

#header_container {
    position:fixed;
    top:0; left:0; right:0;
    text-align:center;
    margin:0;
    height:3em;
    z-index:3;
}

#header {
    position:absolute;
    top:0; left:0; right:0; bottom:0;
    height:100%;
    background:#efefef;
    z-index:3;
    margin:0;
    padding:0.2em;
}

#menu {
    position:fixed;
    top:4em;
    left:0.5em;
    width:9em;
    border: 3px dotted #dedede;
    padding:0.5em;
```

```
}

#content {
  margin:0;
  padding-top:4em;
  padding-left:12em;
  padding-bottom:3em;
  z-index:2;
}

#footer_container {
  position:fixed;
  bottom:0; left:0; right:0;
  text-align:center;
  margin:0;
  height:2em;
  z-index:3;
}

#footer {
  position:absolute;
  top:0; left:0; right:0; bottom:0;
  background:#efefef;
  z-index:3;
}

/* Nur für den MSIE */

* html, * html body {
  overflow:hidden;
  bottom:0;
  height:100%;
}

* html #header_container, * html #footer_container {
  position:absolute;
  width:100%;
  padding-right:16px;
}

* html #menu {
  position:absolute;
}

* html #header,* html #footer {
  height:100%;
  position:static;
}

* html #content {
  position:absolute;
  top:0; bottom:0; left:0; right:0;
  height:100%;
  width:100%;
  overflow:auto;
  margin:0;
}
```

```
/* Weiter mit browserübergreifenden Deklarationen */

#menu li {
    list-style: none;
    margin:0;
    padding:0.3em;
}

#menu li#active {
    padding:0.5em;
    text-decoration:underline;
}

#menu a:link, #menu a:visited {
    display:block;
    background:#EEEEEE;
    color:black;
    text-decoration:none;
    margin:0;
    padding:0.2em;;
}

#menu a:hover {
    background:#4f4f4f;
    color:#FFFFFF;
    text-decoration:underline;
}

/* Der IE soll das Element breiter darstellen, da margin
   und padding hier von der Breite abgezogen werden! */


* html #menu {
    margin:0;
    width:10.5em;
}
```



Vollständiges Layoutbeispiel


An dieser Stelle wird nun noch ein mit Grafiken gestaltetes Beispiel vorgestellt, welches auch eine Anregung für eigene, neue Layouts sein kann.

[Anzeigebeispiel: Komplette Seite](#)

In diesem Beispiel wurde die  [erste Verschachtelungsmethode](#) für den Internet Explorer gewählt, aus diesem Grund können für den IE (per CSS-Hack) auch die semi-transparenten PNG-Grafiken (welche vom IE nur mangelhaft unterstützt werden) problemlos gegen nicht-transparente JPG-Grafiken ausgetauscht werden, ohne dass es zu optischen Veränderungen kommt.

Sie können das gesamte Beispiel auch als gepackte  [.zip](#)- oder  [.tar.gz](#)-Datei herunterladen. Im Download enthalten sind:

- [index.htm](#) - Die eigentliche HTML-Datei

- [self.css](#) - Ausgelagertes Stylesheet
- Die PNG-Grafiken für Header und Footer
- Die JPEG-Grafiken für Header und Footer im IE
- XCF-Dateien der Grafiken (für  [GIMP](#))

An dieser Stelle auch nochmal herzlichen Dank an Alexander Brock, der die Grafiken erstellt hat.



Abschließende Zusammenfassung

Fixierte Header und Footer sind nützlich, doch gibt es hier Probleme mit dem Internet Explorer zu überwinden. Dieser Workaround - der sich auch für ganz andere Zwecke nutzen lässt, als für Header und Footer - eröffnet auch dem IE neue Möglichkeiten, Elemente ohne Frames und ohne Abhängigkeit von JavaScript zu fixieren.



 [SELFHTML aktuell](#)  [Artikel](#)  [CSS](#)

© 2007  [Impressum](#)