

Das Boxmodell

Alle Elemente eines HTML5-Dokuments können als kleine Kästchen bzw. Boxen, sogenannte »little boxes« betrachtet werden. Sobald es um das Layout einer Webseite geht, wird der Begriff des *Boxmodells* verwendet. Das Boxmodell bezieht sich auf den Inhalt, die Ränder, den Rahmen und die Abstände von jeder einzelnen dieser Boxen bzw. HTML5-Elementen. Das Denken in Kästchen bzw. Boxen beim Layout einer Seite sehr von Vorteil.

Jede Box besteht aus den folgenden Teilen:

- » Inhalt (content)
Der Inhalt der Box kann aus Text, Bildern, Videos usw. bestehen.
- » Innenabstand (padding)
Erzeugt einen Abstand zwischen dem Inhalt einer Box und dem Rahmen.
- » Rahmen (border)
Eine Begrenzung jeder Box, diese schließt sich um den Inhalt und den Abstand.
- » Außenabstand (margin)
Definiert einen „Schutzbereich“ um die gesamte Box herum.

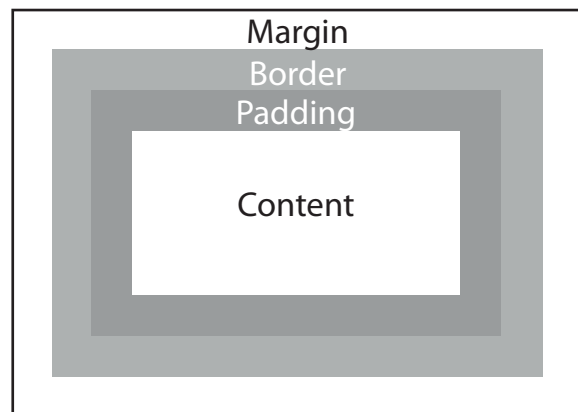


Abb. 1: Aufbau des Boxmodell, Quelle: AH

Breite und Höhe eines Elements

Das Boxmodell ist äußerst wichtig, wenn beim Layouten einer Webseite die Breite und Höhe eines HTML5-Elements definiert wird. Wird in CSS3 nur die Breite und die Höhe eines Elements eingestellt, wirkt sich dies nur auf den Inhaltsbereich aus. Um die gesamte Größe zu ermitteln, müssen der Rahmen, der Innenabstand und der Außenabstand ebenfalls berücksichtigt werden.

```
div { width: 320 px;
      height: 150 px;
      padding: 25 px;
      border: 10 px solid navy;
      margin: 5 px;
}
```

Die Breite eines Elements wird wie folgt berechnet:

Breite (gesamt) = width (content) + left padding + right padding

+ left border + right border + left margin + right margin

Breite: 400 px = 320 px + 25 px + 25 px + 10 px + 10 px + 5 px + 5 px

Die Höhe (height) eines Elements wird auf die gleiche Art berechnet wie die Breite.

Höhe (gesamt) = height (content) + top padding + bottom padding

+ top border + bottom border + top margin

+ bottom margin

Höhe: 230 px = 150 px + 25 px + 25 px + 10 px + 10 px + 5 px + 5 px

Das HTML5-Element hat eine Breite von 400 px und eine Höhe von 230 px. Diese müssen beim Layout einer Webseite für das entsprechende Element eingeplant werden.

Layout in HTML5

Das Layout von Webseiten mit `<div>`-Elementen war schon vor der Veröffentlichung von HTML5 möglich und kann nach wie vor verwendet werden. An machen Stellen macht es sogar Sinn, z. B. beim Einsatz eines *wrappers*, auf dieses Element zurückzugreifen. Beim Verwenden von `<div>`-Elementen ist es immer notwendig, mit IDs zu arbeiten, um die verschiedenen Bereiche einer Webseite auseinanderhalten zu können. Dies ist mit den neuen HTML5-Elementen nicht mehr nötig, denn die neuen Tags zum Layouten von Webseiten haben eine Bedeutung (Semantik). Ein Vorteil von diesen *semantischen Tags* ist, dass sie für Menschen und Maschinen, z. B. Screenreader, eindeutig lesbar sind. Durch den Einsatz dieser neuen semantischen Elemente wird die Barrierefreiheit einer Webseite stark unterstützt.

Die neuen semantischen Tags zum Layouten von Webseiten sind:

<code><header></code>	ist der Kopfbereich für ein Dokument oder einen Abschnitt. Er ist nicht mit dem <code><head></code> -Element eines HTML5-Dokuments zu verwechseln.
<code><nav></code>	definiert einen Bereich für die Navigation einer Webseite.
<code><section></code>	wird zum Einteilen eines Dokuments in inhaltliche Abschnitte verwendet. Die Inhalte, die innerhalb eines <code><section></code> -Elements stehen, sollten einen Sinnzusammenhang haben.
<code><article></code>	ist ein in sich geschlossener inhaltlicher Bereich.
<code><aside></code>	definiert eine Seitenleiste bzw. Marginalspalte.
<code><footer></code>	enthält die Informationen der Fußzeile.
<code><figure></code>	enthält Abbildungen jeglicher Art.
<code><figcaption></code>	dient zur Beschriftung eines <code><figure></code> -Elements.

HTML5 Quellcode

```

<body>

<header>
<h1>City Gallery</h1>
</header>

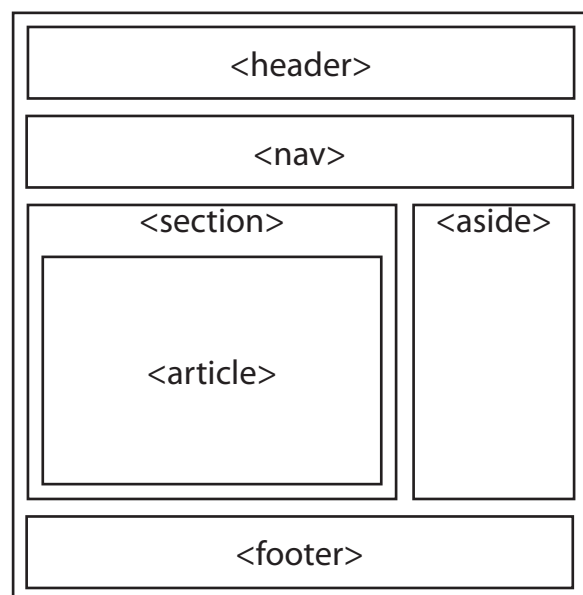
<nav>
<p>London | Paris | Tokyo</p>
</nav>

<section>
<h1>London</h1>
<p>London is the capital city of England.... </p>
</section>

<footer>Copyright © W3Schools.com</footer>
</body>

```

Wireframelayout mit HTML5-Tags



Quelle: AH

Block- und Inline-Elemente in HTML5

Jedes HTML5-Element wird auf eine bestimmte Art am Bildschirm angezeigt, je nachdem, um welche Art von Element es sich handelt. Die meisten Elemente lassen sich in sogenannte Block- oder Inline-elemente unterscheiden.

Ein Blockelement beginnt immer in einer neuen Zeile und nimmt die gesamte Breite des Browserfensters in Anspruch. Beispiele für Blockelemente sind: `<div>`, `<h1>`-`<h6>`, `<p>`, usw.

Ein Inline-Element beginnt nicht in einer neuen Zeile, sondern in eine Zeile integriert, also oft ein Teil eines Blockelements. Inline-Elemente können bspw. Text oder andere Inline-Elemente enthalten, aber niemals Blockelemente. Inline-Elemente sind immer so breit und so hoch, wie der Inhalt, den sie beinhalten. Beispiele für Inline-Elemente sind, ``, `<a>`, ``, ``, usw.

Das `<div>`-Element

Das `<div>`-Element ist ein Blockelement, das oft als „Box“ für andere HTML5-Elemente verwendet wird. Es gibt für das `<div>`-Element keine erforderlichen Attribute, es wird aber häufig durch Klassen und IDs mit CSS3 angesprochen. Das `<div>`-Element verwendet man häufig beim Layout von Webseiten.

```
<div id="content">
    <h2>London</h2>
    <p>London ist die Hauptstadt von England. Es ist die beliebteste Stadt des United Kingdom.</p>
</div>
```

Das ``-Element

Das ``-Element ist ein Inline-Element, das häufig zur optischen Gruppierung von Worten verwendet wird. Durch den Einsatz von CSS3 kann ein ``-Element über eine Klasse oder eine ID angesprochen werden.

```
<p>Meine Mutter hat <span style="color: blue">blaue</span> Augen.</p>
```

Layout mit dem `<div>`-Element

Das `<div>`-Element wird oft als Layoutelement bei der Gestaltung von Webseiten verwendet, weil es sehr leicht mit CSS3 gestaltet und positioniert werden kann. Dadurch besteht die Struktur einer Seite aus mehreren `<div>`-Elementen, die wiederum weitere Elemente, wie Überschriften, Texte, Bilder etc. enthalten. Gängige IDs für `<div>`-Elemente sind *header*, *footer*, *navigation* usw.

HTML5 Quellcode

```
<body>
<div id="header">
<h1>City Gallery</h1>
</div>
<div id="nav">
<p>London | Paris | Tokyo</p>
</div>
<div id="section">
....
```

CSS3 Quellcode

```
#header {
    background-color:black;
    color:white;
    text-align:center;
    padding:5 px;
}
#nav {
    background-color:#eeeeee;
    height:300 px;
    width:100 px;
}
```