



Seitengestaltung mit HTML und CSS

Stefan Rothe, Thomas Jampen

2013-09-12



Rechtliche Hinweise

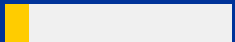
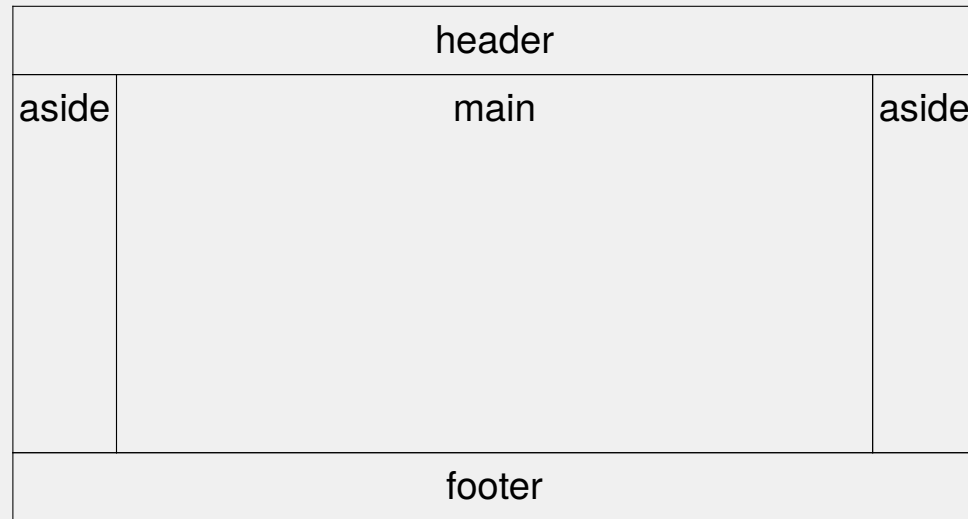
Dieses Werk von *Stefan Rothe* steht unter einer *Attribution-NonCommercial-ShareAlike 3.0 License*.



Zudem verzichtet der Autor auf sämtliche Urheberrechtsansprüche für die in diesem Werk enthaltenen Quelltexte.



Seitenstruktur





Elemente für die Seitenstrukturierung

Element	Bedeutung	Bemerkung
<code><header></code>	Kopfbereich der Webseite	
<code><footer></code>	Fussbereich der Webseite	
<code><main></code>	Hauptbereich der Webseite	
<code><aside></code>	Seitlicher Bereich	
<code><nav></code>	Navigationsbereich, z.B. für Menüs	
<code><div></code>	Allgemeiner Bereich (<i>division</i>)	



Positionierung

Mit der CSS-Eigenschaft `position` können HTML-Elemente positioniert werden. Die vier möglichen Positionierungsarten sind in der nebenstehenden Tabelle ersichtlich.

Einige Positionierungsarten können durch die Eigenschaften `left`, `top`, `right` und `bottom` beeinflusst werden.

```
nav {  
  position: fixed;  
  top: 0;  
  left: 0;  
}
```

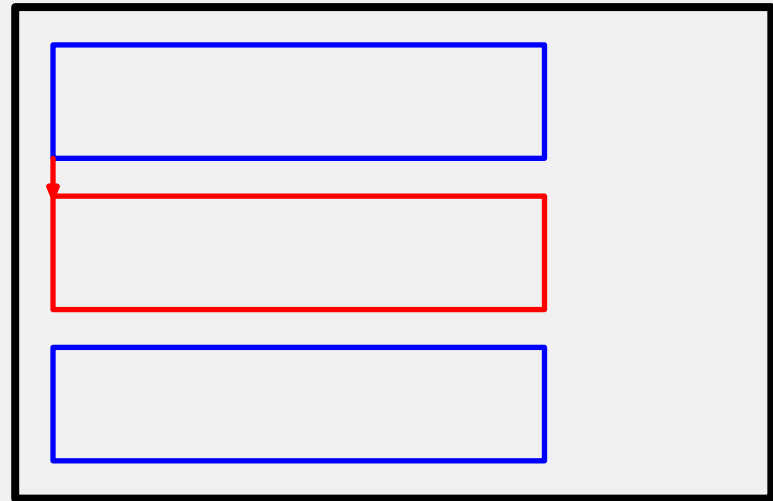
Bei speziell positionierten Elementen ist of auch eine explizite Angabe der Grösse des Elements mit `width` und `height` sinnvoll.

Element	Bedeutung
<code>static</code>	statisch
<code>fixed</code>	fixiert
<code>relative</code>	relativ
<code>absolute</code>	absolut



Statische Positionierung

Normalerweise werden Elemente **statisch** positioniert. Das heisst, dass die Elemente an die Stelle zu liegen kommen, wo sie ‚hingehören‘.





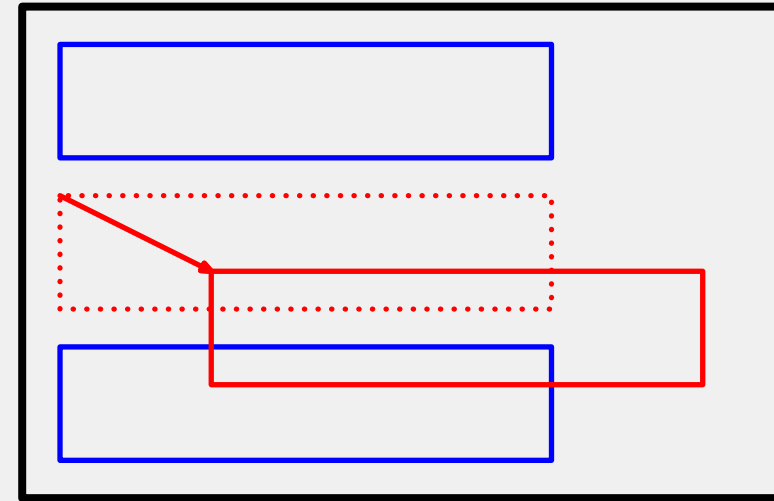
Relative Positionierung

Bei der **relativen** Positionierung mit `relative` wird wie bei der statischen Positionierung Platz für das Element im normalen Elementfluss reserviert. Das Element kann jedoch relativ zu seiner eigentlichen Position verschoben werden.

Mit den CSS-Eigenschaften `left` und `top` wird die Verschiebung des Elements zu seiner ursprünglichen Position angegeben. Dabei können auch negative Werte angegeben werden.

So können beispielsweise Absätze nach rechts verschoben werden, um am linken Rand Platz für ein Menü zu schaffen:

```
p {  
  position: relative;  
  left: 2cm;  
  top: 1cm;  
}
```





Fixe Positionierung

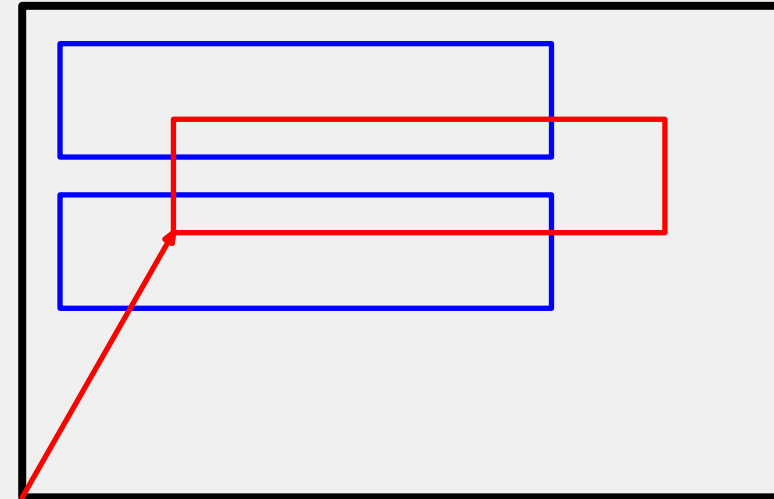
Bei der **fixen** Positionierung mit **fixed** wird das Element aus dem normalen Elementfluss entfernt. Das Element wird fest auf dem Browserfensters positioniert, es scrollt nicht mit den restlichen Elementen mit.

Mit den CSS-Eigenschaften **left** oder **right** kann ein fixiertes Element relativ zu linken oder rechten Seite des Browserfensters positioniert werden.

Mit **top** oder **bottom** wird die Position des Elements in Bezug auf den oberen oder unteren Rand des Fensters festgelegt.

Eine typische Anwendung ist die Positionierung eines Menüs, welches immer zugänglich ist:

```
nav {  
  position: fixed;  
  left: 6cm;  
  bottom: 10cm;  
}
```





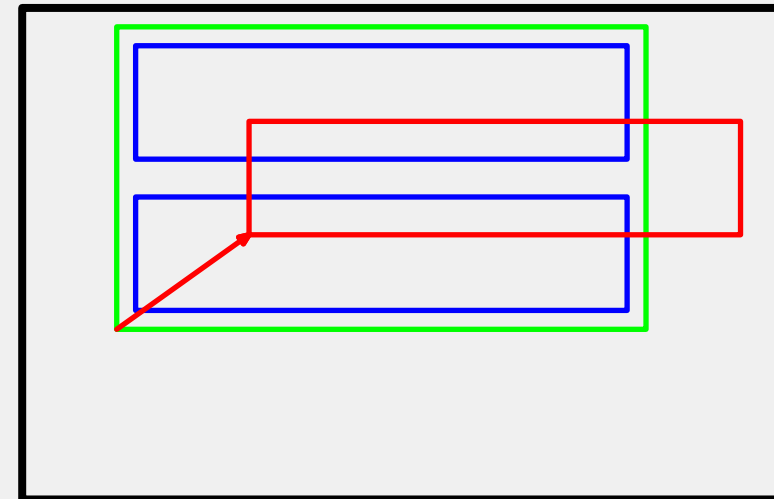
Absolute Positionierung

Auch bei der **absoluten** Positionierung mit **absolute** wird das Element aus dem normalen Elementfluss entfernt. Die Position des Elements wird bezüglich des ersten Elternelements, welches **nicht statisch** positioniert ist, angegeben. Wenn kein solches Element vorhanden ist, wird das Element bezüglich der ganzen Webseite positioniert.

Mit den CSS-Eigenschaften **left** oder **right** wird die Position des Elements bezüglich des linken oder rechten Rands des Bezugslements angegeben. Mit **top** oder **bottom** wird die Position des Elements in Bezug auf den oberen oder unteren Rand des Bezugslements festgelegt.

So kann beispielsweise ein mitscrollendes **aside**-Element am rechten Rand platziert werden:

```
div {  
  position: absolute;  
  left: 2cm;  
  bottom: 1cm;  
}
```





Übung 4: Der Zauberlehrling mit Layout

Ändern Sie die Struktur der Seite wie folgt:

- Der Titel gehört (inkl. `h1`-Tag) in den `header`-Bereich (**nicht** `head`-Bereich!).
- Das gesamte Gedicht soll im `main`-Bereich stehen.
- Um das Bild herum definieren Sie den `aside`-Bereich.

Passen Sie nun die Darstellung wie folgt an:

- Das Bild soll rechts neben dem Gedichttext stehen und beim Scrollen stehen bleiben.





Klassen-Selektoren

Oft sollen nicht alle Elemente eines Typs gleich formatiert werden. Um Elemente eines Typs zu unterscheiden, können in HTML einem Element mit dem `class`-Attribut eine oder mehrere **Klassen** zugeordnet werden.

Der Name einer Klasse darf keine Leerzeichen enthalten.

```
<p class="gedicht strophe">Hat der alte Hexenmeister...</p>  
<p class="gedicht refrain">Walle! walle...</p>
```

In CSS werden mit einem Klassen-Selektor nur diejenigen Elemente ausgewählt, welche die entsprechende Klasse besitzen:

```
p.gedicht { background-color: black; }  
p.strophe { color: white; }  
p.refrain { color: yellow; }
```



ID-Selektor

In HTML kann jedem Element eine eindeutige Bezeichnung zugeordnet werden. Man spricht hier von einer **ID**. Diese wird über das `id`-Attribut festgelegt und muss innerhalb des HTML-Dokuments **eindeutig** sein.

```
<p id="erstestrophe">Hat der alte Hexenmeister...</p>  
<p id="zweitestrophe">Und nun komm, du alter Besen!...</p>
```

In CSS kann eine ID als Selektor für das entsprechende Element verwendet werden. Dazu wird der ID ein Doppelkreuz (`#`) vorangestellt:

```
#erstestrophe { color: black; }  
#zweitestrophe { color: white; }
```



Kombinations-Selektoren 1

Element1 + *Element2*

Selektiert ein *Element2*, das **direkt nach** einem *Element1* liegt.

```
h1 + p {  
    color: red;  
}
```

```
<h1>Beispiel</h1>  
  <p>Dieser Absatz ist rot.</p>  
  <p>Dieser nicht mehr.</p>  
<h2>Weiteres Beispiel</h2>  
  <p>Dieser ist ebenfalls nicht rot.</p>
```



Kombinations-Selektoren 2

Element1 > *Element2*

Selektiert ein *Element2*, das **direkt innerhalb** eines *Element1* liegt.

```
h1 > i {  
    color: red;  
}
```

```
<h1>  
  Beispielsweise <i>ist dies rot</i>,  
  <b><i>dies aber nicht</i></b>,  
  <i>dies ist wieder rot</i>.  
</h1>  
<p><i>Dies ist nicht rot.</i></p>
```



Kombinations-Selektoren 3

Element1 *Element2*

Selektiert ein *Element2*, das **innerhalb** eines *Element1* liegt.

```
h1 i {  
    color: red;  
}
```

```
<h1>  
  Beispielsweise <i>ist dies rot</i>,  
  <b><i>und dies auch</i></b>,  
  <i>und dies ebenfalls</i>.  
</h1>  
<p><i>Dies ist nicht rot.</i></p>
```



Übersicht Selektoren

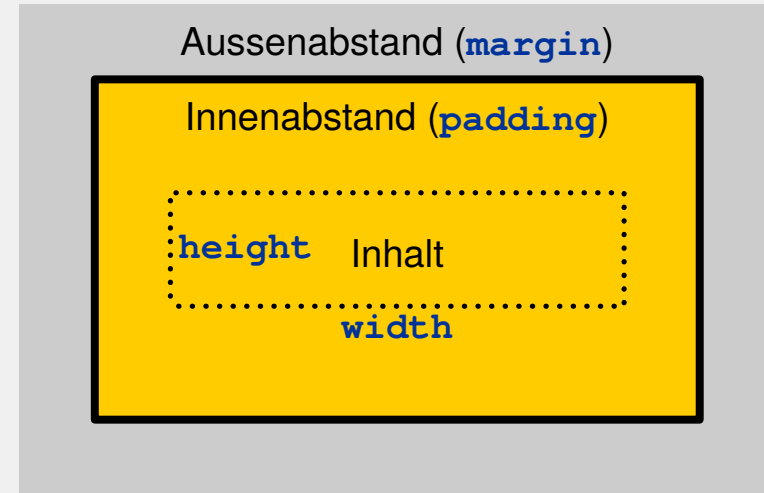
Schreibweise	Selektiert	Beispiel
	alle Vorkommen von	<code>p</code>
<code>,</code>	alle Vorkommen von und	<code>h1, h2, h3</code>
<code>.</code>	Vorkommen von mit der angegebenen	<code>p.gedicht</code>
	Vorkommen von innerhalb von	<code>h1 i</code>
<code>></code>	Vorkommen von direkt innerhalb von	<code>h1 > p</code>
<code>+</code>	Vorkommen von direkt nach einem	<code>h1 + p</code>
<code>#</code>	Wähle das Element mit der angegebenen	<code>#erstesBild</code>



Abstände

Für HTML-Blockelemente können zwei Arten von Abständen festgelegt werden:

- Der **Innenabstand** (*padding*) zwischen Inhalt und Rahmen. Er wird mit dem Hintergrund des Elements gefüllt.
- Der **Aussenabstand** (*margin*) zwischen Rahmen und den umgebenden Elementen. Hier ‚scheint‘ der Hintergrund des umgebenden Elements durch.



Innen- und Aussenabstand können entweder für alle Seiten gleich oder für einzelne Seiten unterschiedliche festgelegt werden:

	Alle Seiten	Oben	Unten	Links	Rechts
Aussenabstand	<i>margin</i>	<i>margin-top</i>	<i>margin-bottom</i>	<i>margin-left</i>	<i>margin-right</i>
Innenabstand	<i>padding</i>	<i>padding-top</i>	<i>padding-bottom</i>	<i>padding-left</i>	<i>padding-right</i>



Rahmen

Alle HTML-Elemente können mit einem Rahmen umgeben werden.

```
/* Blauer Rahmen für Bilder: */  
img {  
  border-style: solid;  
  border-color: blue;  
}
```

Eigenschaft	Bedeutung	Mögliche Werte
<code>border-color</code>	Rahmenfarbe	Farbe
<code>border-style</code>	Rahmenart	Rahmenart
<code>border-width</code>	Rahmendicke	Grösse

Die Rahmeneigenschaften können auch für jede Seite separat festgelegt werden. Dazu werden folgende Eigenschaften verwendet:

Oben	Unten	Links	Rechts
<code>border-top-color</code>	<code>border-bottom-color</code>	<code>border-left-color</code>	<code>border-right-color</code>
<code>border-top-style</code>	<code>border-bottom-style</code>	<code>border-left-style</code>	<code>border-right-style</code>
<code>border-top-width</code>	<code>border-bottom-width</code>	<code>border-left-width</code>	<code>border-right-width</code>



Rahmenart festlegen

CSS akzeptiert die nebenstehenden Werte als Rahmenart.

Damit ein Rahmen gezeichnet wird, muss unbedingt eine Rahmenart festgelegt werden, ansonsten wird der Wert **none** angenommen.

```
/* Doppellinie oberhalb von Überschriften: */  
h1 {  
  border-style-top: double;  
}
```

Wert	Bedeutung
none	kein Rahmen
dotted	gepunktete Linie
dashed	gestrichelte Linie
solid	durchgezogene Linie
double	Doppellinie
groove	3D-Effekt: Rinne
ridge	3D-Effekt: Erhebung
inset	3D-Effekt: vertieft
outset	3D-Effekt: erhöht



Übung 5: Der Zauberlehrling (gestaltet)

Lassen Sie die neuen Erkenntnisse in die Gestaltung einfließen.

Beispiele:

- Der Text soll nicht im linken Rand "kleben".
- Das Bild soll einen Rahmen erhalten (mit etwas Luft zwischen Rahmen und Bild).
- Entfernen Sie die `em`-Tags (die Sie bei jedem 2. Vers eingesetzt haben) und realisieren Sie die unterschiedliche Darstellung über das `class`-Attribut.
- Entfernen Sie das `q`-Tag beim letzten Vers und vergeben Sie stattdessen das `id`-Attribut, um diesen Vers nach Ihren Wünschen speziell hervorzuheben.
- Mit Abständen zwischen den Versen experimentieren.