

WEB DEVELOPMENT TRENDS

Christopher Beck

02 • HTML & CSS REVISITED

HTML REVISITED

AUFBAU EINER HTML-SEITE

Jede HTML-Seite besteht aus *Elementen*, die im Quellcode durch *Tags* definiert werden. Durch das Positionieren der *Elemente* wird ein Layout geschaffen.

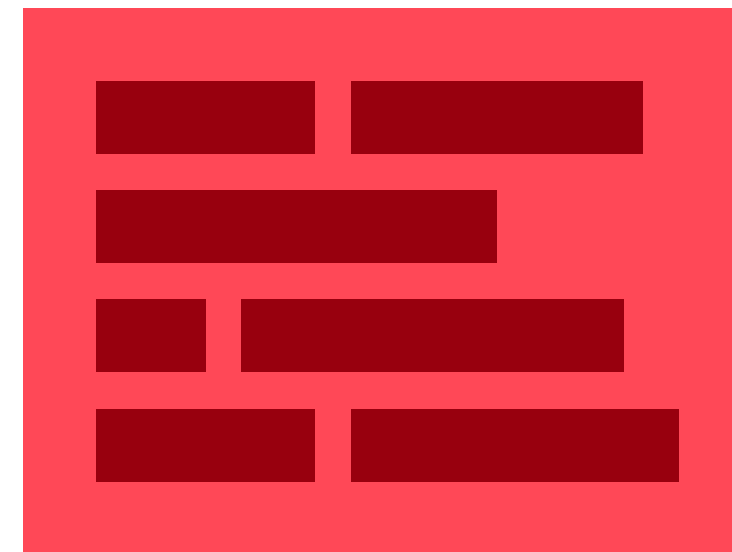
```
1  <html>
2
3  <head>
4      <title>Seitentitel</title>
5  </head>
6
7  <body>
8
9      <h1>Überschrift</h1>
10
11     <p>Ein Absatz innerhalb des Dokuments.</p>
12 </body>
13
14 </html>
```

BLOCK- UND INLINE-ELEMENTE

Elemente werden in zwei Gruppen eingeteilt. Beide unterscheiden mehrere Eigenschaften. Für die Positionierung ist jedoch ein Wesentlicher relevant:



Block-Elemente erzeugen einen Umbruch im Seitenfluss



Inline-Elemente erzeugen keinen Umbruch, die Größe richtet sich nach dem Inhalt

FRAGEN

- ➔ Welche weiteren Eigenschaften unterscheiden die beiden Elemententypen?
- ➔ Gibt es weitere Typen?
- ➔ Nennen Sie Beispiele für Block- und Inline-Elemente

BEISPIELE

Block-Elemente

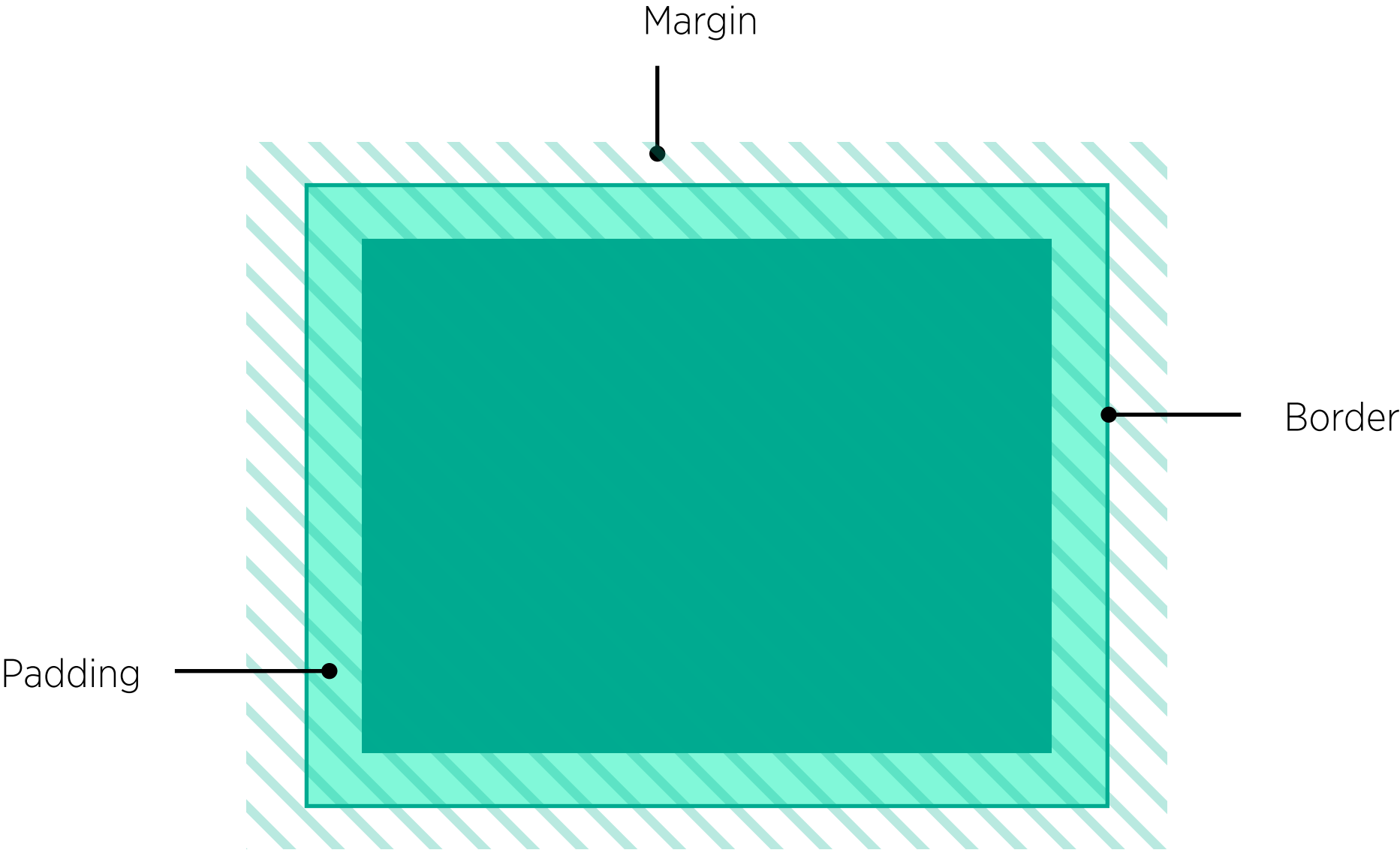
<div>
<table>
<fieldset>

<hr>

Inline-Elemente

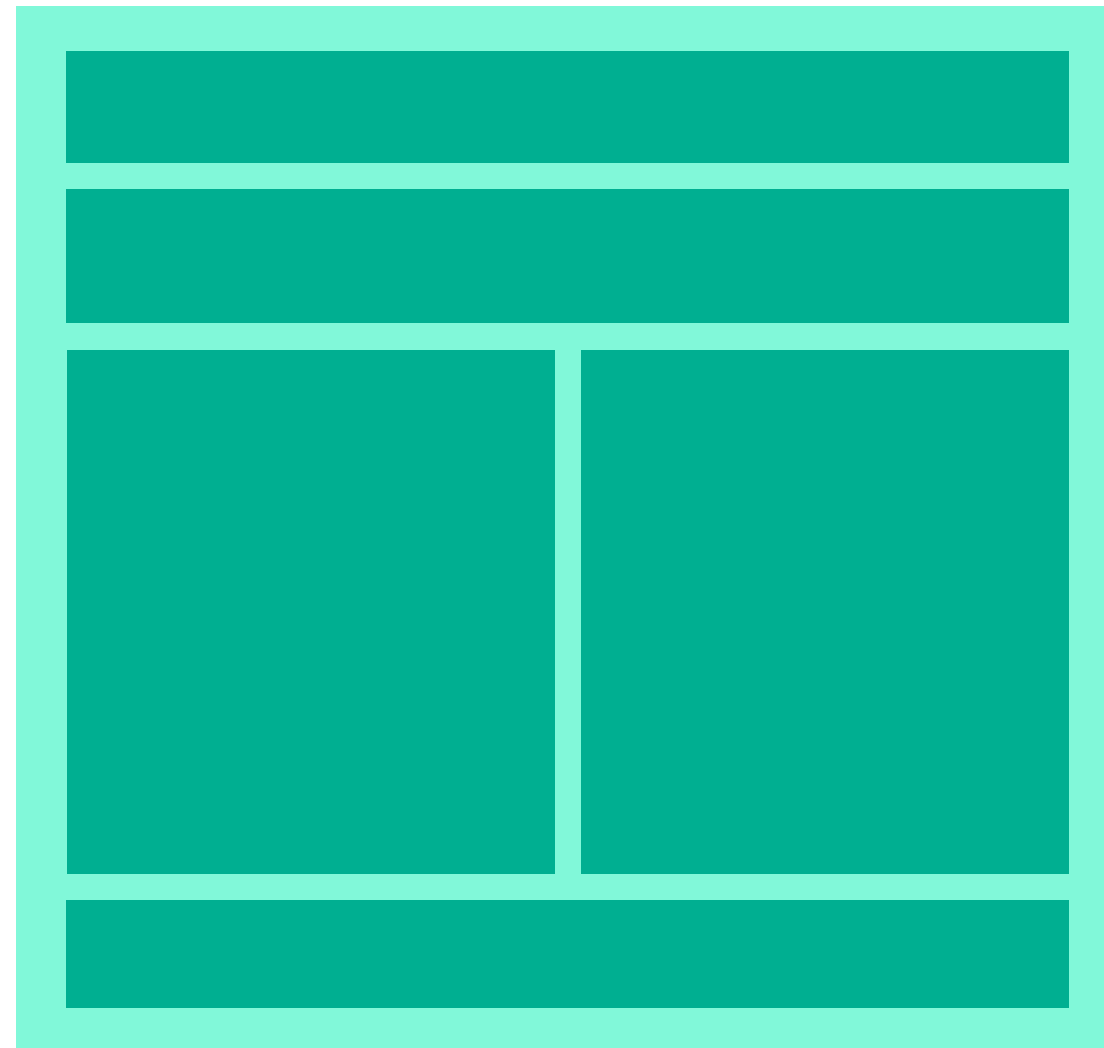
<a>
<input>

DAS BOX-MODELL



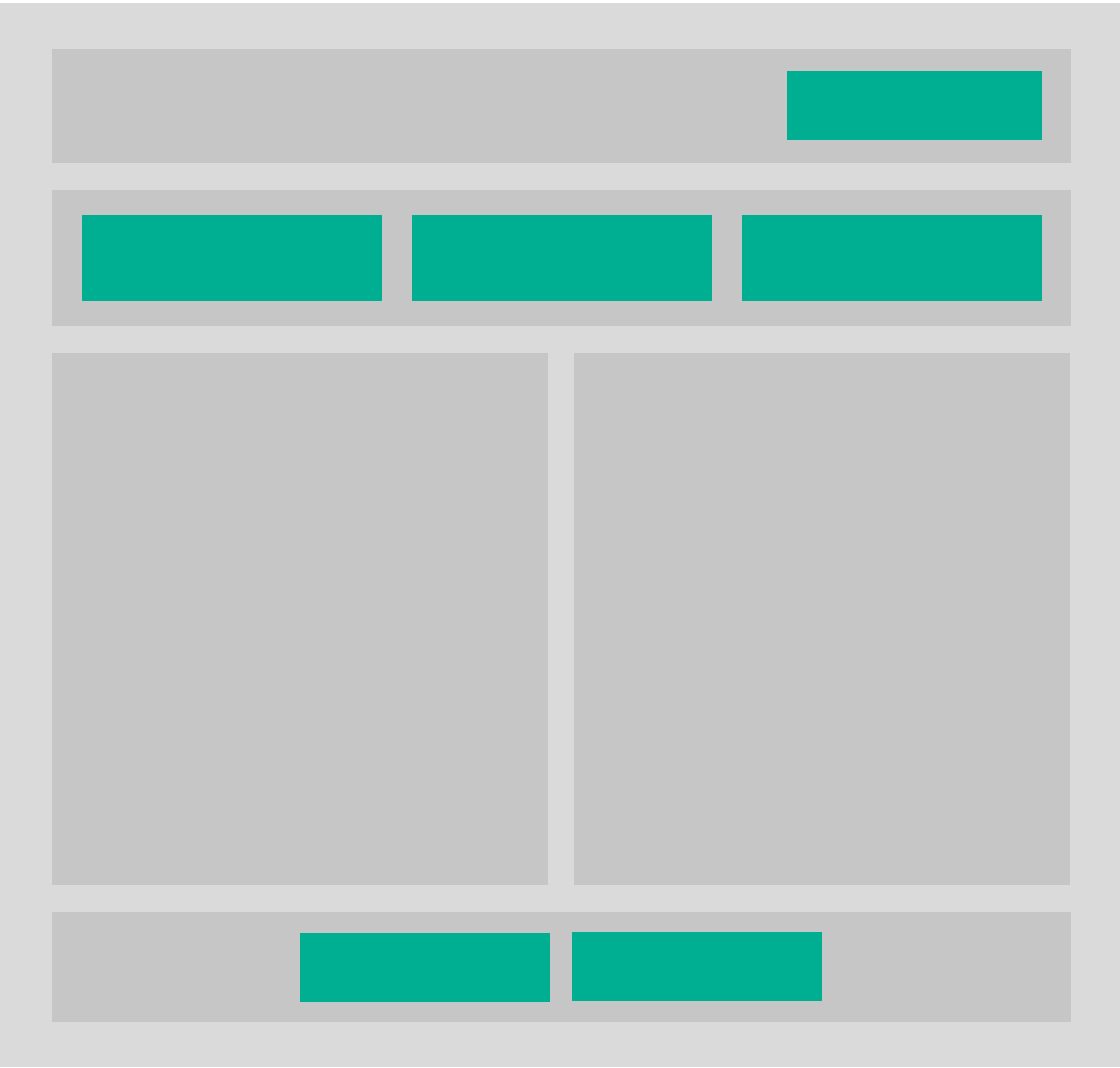
EIN LAYOUT AUS ELEMENTEN BAUEN

Durch die Positionierung mehrerer *Elemente* wird ein Layout generiert.



VERSCHACHTELN

Jedes *Element* kann weitere Elemente enthalten. So wird das Layout weiter unterteilt.



CSS REVISITED

ELEMENTE SELEKTIEREN

Der Zugriff auf HTML-Elemente mit CSS erfolgt über *Selektoren*.

ID

IDs sind einzigartig und sollten nur einmal in einem Dokument vergeben werden.

```
1 <div id="start">
2   Inhalt
3 </div>
```

```
1 #start {
2   ...
3 }
```

KLASSE

Klassen werden mehrfach verwendet. So können die gleichen Stile auf verschiedene Elemente angewendet werden.

```
1 <h2 class="blau">
2   Inhalt
3 </h2>
```

```
1 .blau { ... }
2
3 h2.blau { ... }
```

FRAGEN

- Es gibt noch einen weiteren Selektor. Welchen?
- Neben dem Zugriff über Selektoren gibt es noch eine weitere Möglichkeit Elemente zu formatieren. Welche?

DER SELEKTOR *

ACHTUNG!

*Der Selektor * formatiert jedes Element in einem Dokument mit den definierten Eigenschaften. Beispielsweise erhält ein Element die Hintergrundfarbe, die Sie so formatieren, wenn das Element diese Eigenschaft besitzt. Aus diesem Grund sollten Sie diesen Selektor mit Vorsicht einsetzen. Bedenken Sie, dass Sie die formatierten Eigenschaften später von jedem Element entfernen müssen, da die gewünschte Formatierung nicht erhalten soll.*

In der Entwicklung wird der Selektor deshalb sparsam eingesetzt. Sinnvoll kann beispielsweise die Definition einer einheitlichen Zeilenhöhe für alle Elemente eines Dokuments sein.

SELEKTOREN KOMBINIEREN

Sie können die verschiedenen Selektoren miteinander kombinieren um so eine feinere Auswahl zu bekommen.

```
1      <div id="start">
2          <a href="index.html">Startseite</a>
3          <a href="links.html">Linkliste</a>
4          <a href="impressum.html">Impressum</a>
5      </div>
```

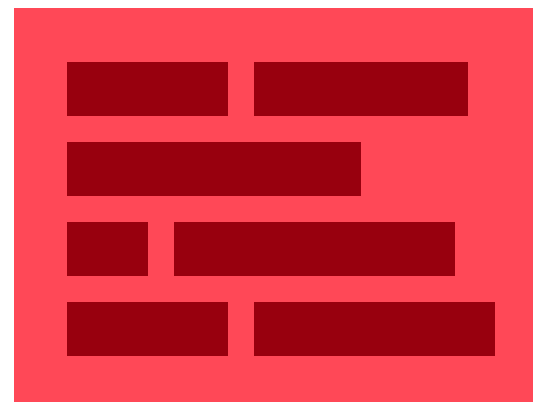
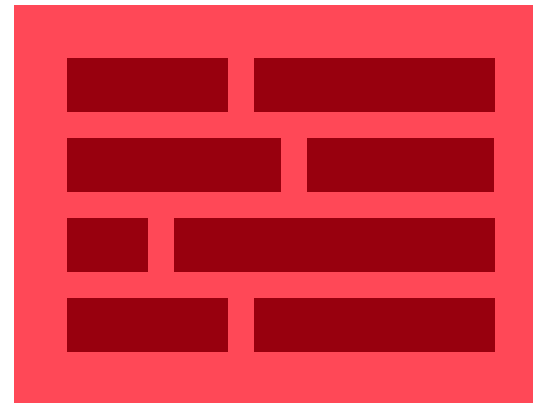
```
1      #start a {
2          ...
3      }
```

STILE AUF MEHRERE SELEKTOREN ANWENDEN

Sie können mehrere Selektoren gleichzeitig auswählen. So können Sie definierte Regeln auf mehrere Elemente anwenden.

```
1      <a class="blau" href="links.html">Links</a>
2
3      <p>Ein Text</p>
4
5
```

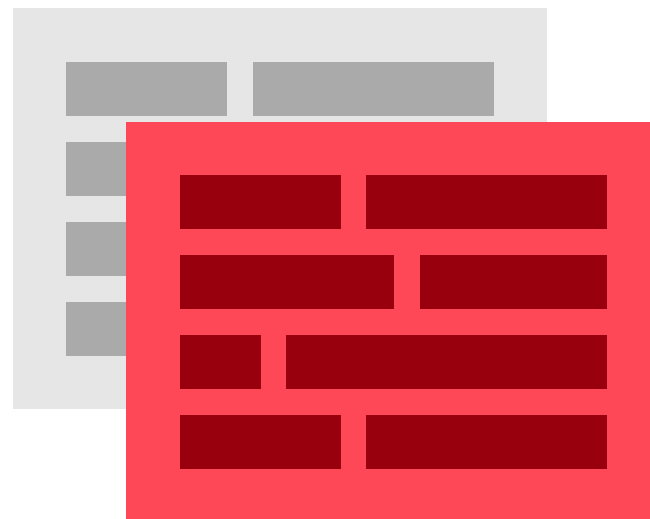
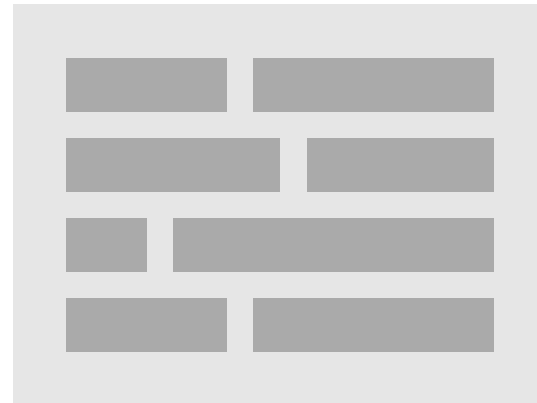
```
1      .blau, p {
2          text-align: left;
3      }
```

ELEMENTE POSITIONIEREN

Blockelemente erzeugen einen Umbruch.
So wird jedes Blockelement untereinander
angeordnet.

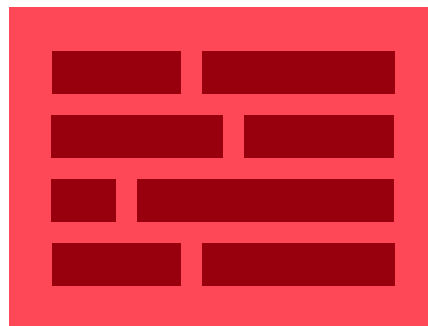
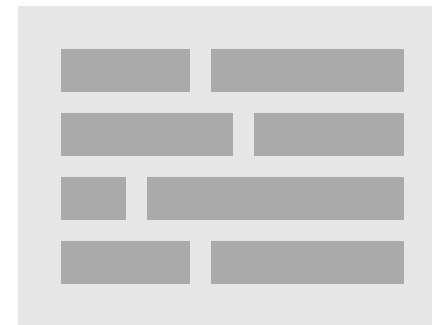
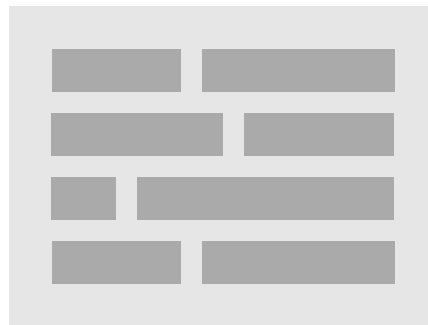
Oft ist es jedoch nötig auch Blockelemente
horizontal anzuordnen. Hierfür gibt es mitt-
lerweile mehrere Möglichkeiten.



DER KLASSISCHE WEG MIT *FLOAT*

Durch *float* wird das Element aus der regulären Anzeigestruktur herausgelöst.

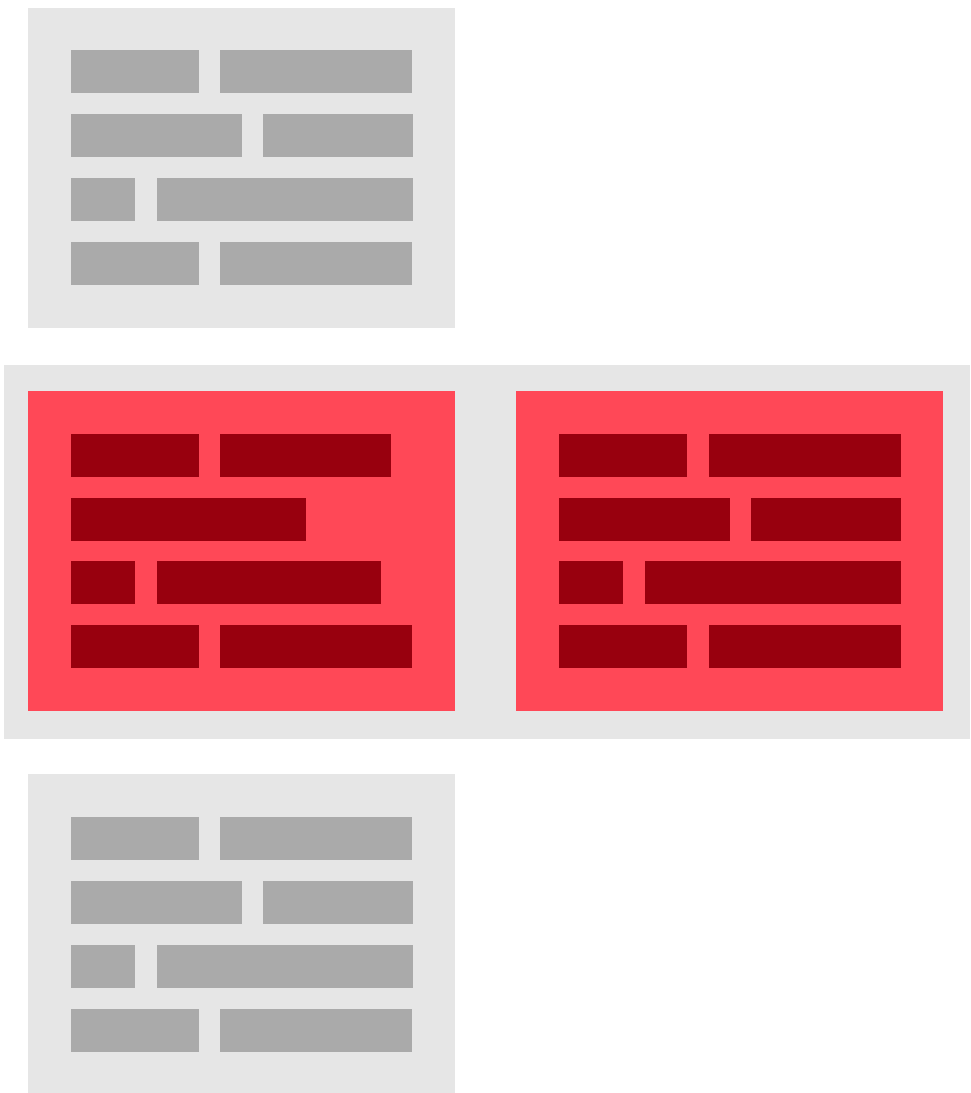
Die nachfolgenden Elemente ignorieren das Element. Ohne Formatierung können Überschneidungen entstehen.



FLOAT AUFHEBEN

Wird ein weiteres Element mit *float* formatiert, schließt es sich an das vorherige Element an.

Mit *clear* wird dieser Effekt aufgehoben. Die CSS-Eigenschaft erhält das **erste Element** das in einer **neuen Zeile** beginnen soll.



ALTERNATIVE LÖSUNG

Ein umschließendes Element verhindert das Überlappen der Elemente.

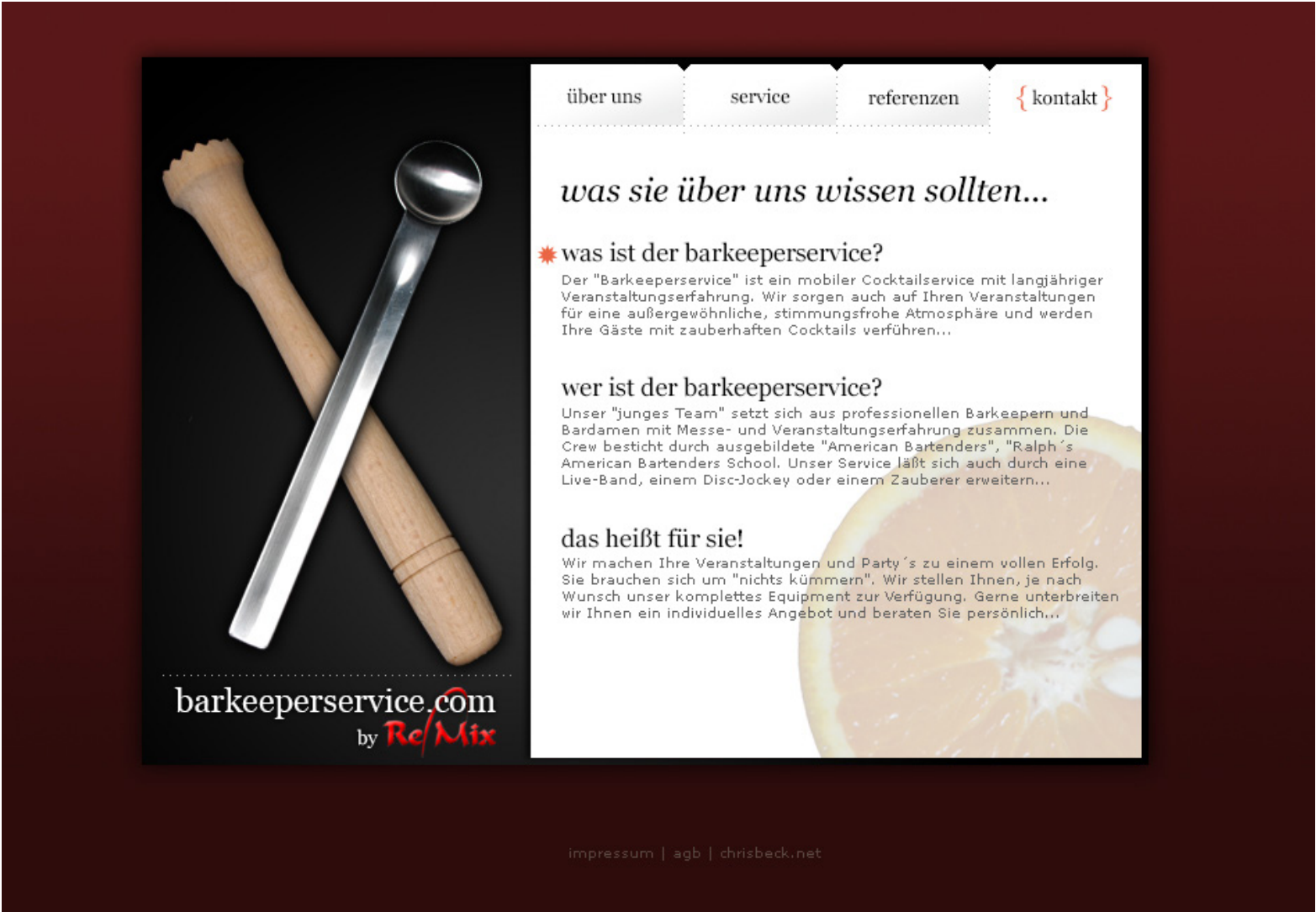
ACHTUNG! Auch das umschließende Element ignoriert die „schwebenden“ Elemente. Durch Angabe der Höhe wird Platz geschaffen.

ALTERNATIVE MIT *DISPLAY: INLINE-BLOCK*

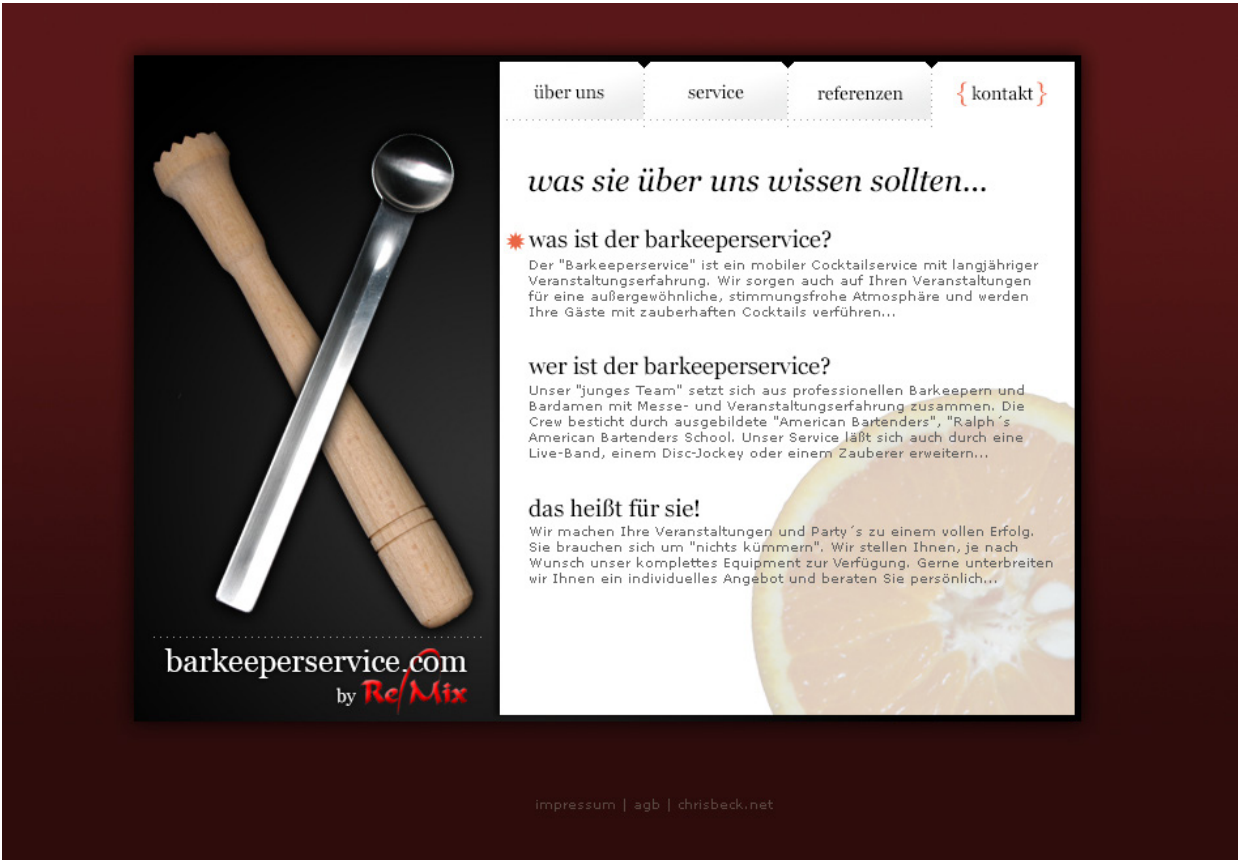
Das Inline-Block Element ist eine Kombination aus Inline und Block Element. Dabei verhält es sich innen wie ein Block Element und kann somit Angaben für Breite und Höhe erhalten. Gegenüber anderen Elementen verhält es sich jedoch wie ein Inline-Element und erzeugt somit keinen Umbruch.

EIN LAYOUT UMSETZEN

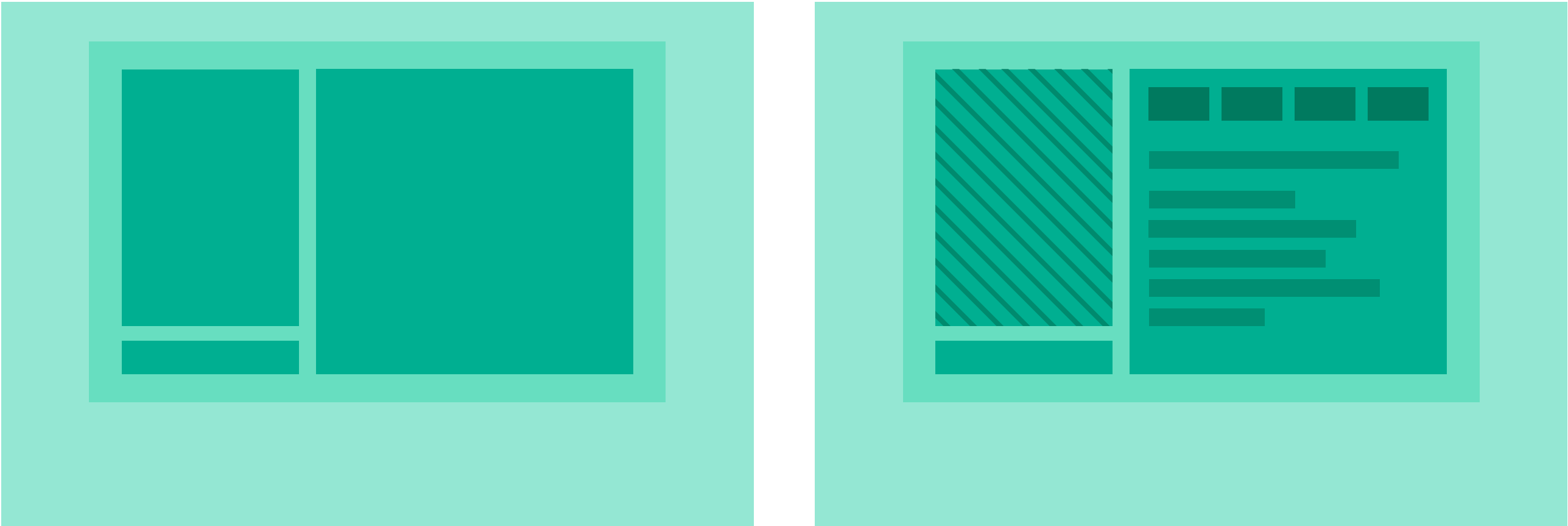
VORLAGE



RASTEREINTEILUNG

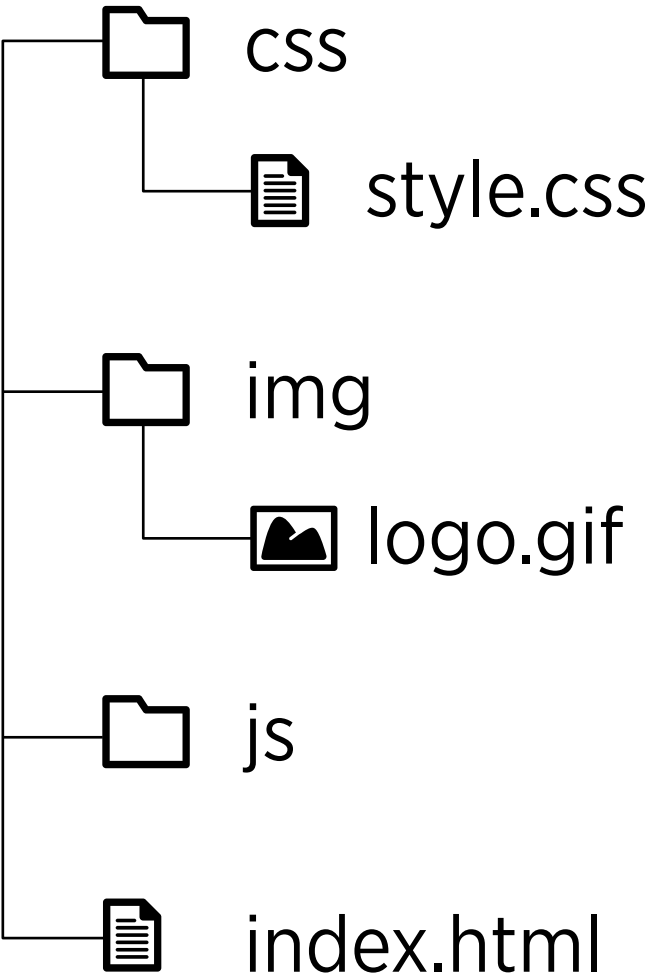


VON AUSSEN NACH INNEN



TIPS & TRICKS

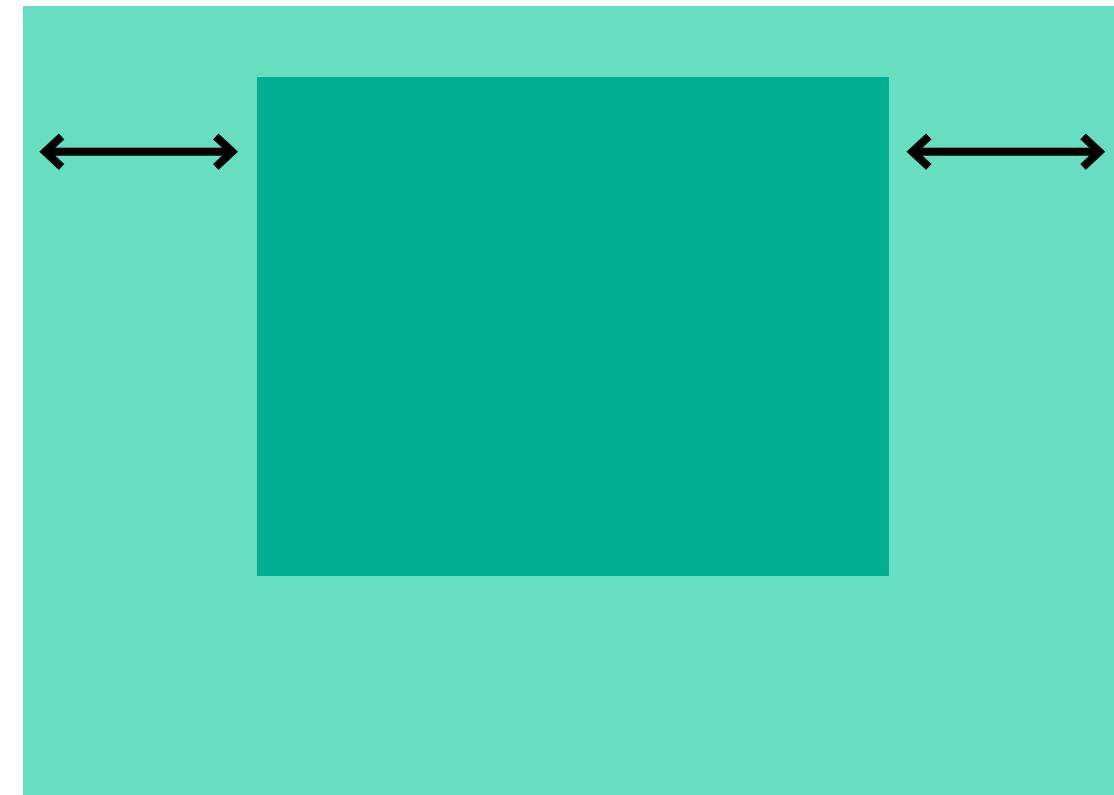
DATEIEN ORGANISIEREN



CLEVER ZENTRIEREN

Mit *margin: auto;* werden die horizontalen Außenabstände automatisch berechnet. Damit dies funktioniert muss eine Breite angegeben werden.

```
1  div {  
2      width: 150px;  
3      margin: auto;  
4  }
```



Fragen?

TRAINING 2

Setzen Sie diese Grafik in HTML um!



Setzen Sie diese Grafikvorlage als HTML-Seite um. Teilen Sie das Layout zunächst in ein Raster ein (*zeichnen!*). Erstellen Sie dann die nötigen Elemente zur Einteilung der Seite. Im letzten Schritt befüllen Sie die Seite mit Inhalten und formatieren diese.

Achten Sie auf eine „saubere Umsetzung“. Zeigen Sie mir das Ergebnis bevor Sie die Lehrveranstaltung verlassen.

- Die Angabe inkl. aller nötigen Daten finden Sie im ILIAS
- Bei der Umsetzung dürfen Sie Schriftarten, jedoch nicht die Formatierung, außer Acht lassen
- Sie müssen kein flexibles Layout erstellen