

STUDIEREN. WISSEN. MACHEN.

CSS



HTML



CSS

Das Boxenmodell

Inhalte

- › Exkurs: Maßeinheiten in CSS
- › Gestaltung der HTML-Boxen mit CSS
- › HTML: Hilfsboxen

CSS: Größenangaben

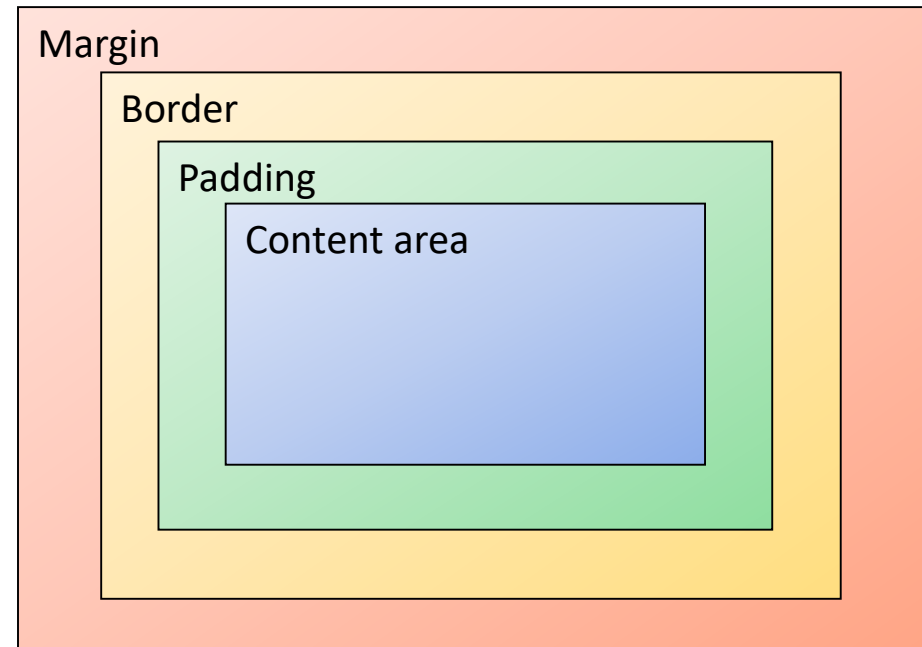
- › Eine *absolute Maßeinheit* für die Ausgabe am Bildschirm ist ein *Pixel*:
 - › **px**: Anzahl in Pixel (in CSS3 wird ein Pixel angenommen als 1/96 Zoll (1 Zoll = 2,54cm, 1 Pixel = ca. 0,25mm))
- › Für *relative Größenangaben* sind auch *Prozentzahlen* geeignet.
 - › *Prozentangaben* beziehen sich immer auf die entsprechenden Größenangaben des *übergeordneten Elternelementes*.

Inhalte

- › Maßeinheiten in CSS
- › Gestaltung der HTML-Boxen mit CSS
- › HTML: Hilfsboxen

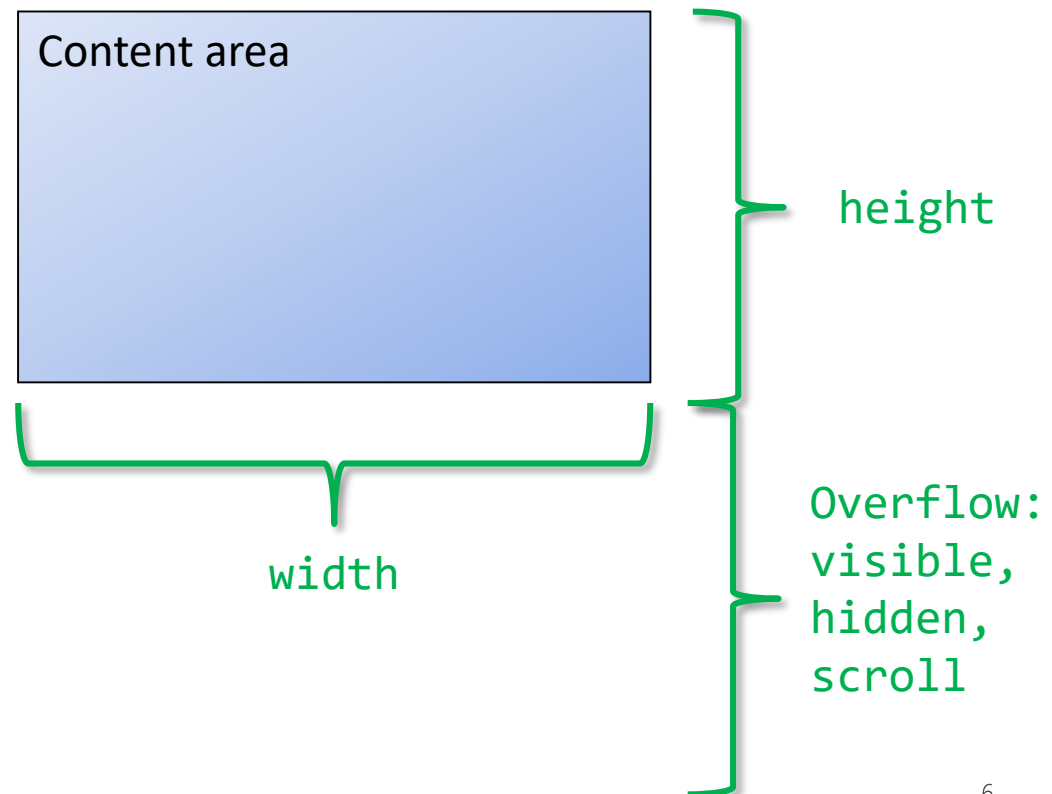
CSS - Boxenmodell

- › HTML-Boxen können mit CSS vielfältig gestaltet werden.
- › Eine Box ist aus CSS-Sicht in vier Bereiche untergliedert:
 - › *Content area*: Inhalt der Box
 - › *Padding*: Innenrand („Polster“)
 - › *Border*: Rahmen
 - › *Margin*: Außenrand
- › Jeder Bereich kann unabhängig von den anderen Bereichen mit CSS-Mitteln gestaltet werden.



Der Inhaltsbereich

- › Die Inhalte eines HTML-Elementes werden im *Inhaltsbereich* (*content area*) der zugehörigen Box dargestellt.
- › Der Inhaltsbereich ist ein Rechteck mit einer definierten *Breite* und *Höhe*.
- › Die CSS-Eigenschaften zur Beschreibung des Inhaltsbereichs sind:
 - **width**: Angabe der Breite in Maßeinheiten (meistens **px**) oder Prozent
 - **height**: Angabe der Höhe in Maßeinheiten (meistens **px**)



Der Inhaltsbereich

⚙
HTML
▼

```

1 <p class="box1">
2   Das ist ein viel zu langer Text.
3 </p>
4 <p class="box2">
5   Das ist auch ein Text, der für die
   umgebende Box zu lang ist.
6 </p>

```

⚙
CSS
▼

```

1 .box1 {
2   background: salmon;
3   width: 100px;
4   height: 50px;
5 }
6 .box2 {
7   background: lightskyblue;
8   width: 120px;
9   height: 60px;
10 }

```

Das ist ein viel zu langer Text.

Das ist auch ein Text, der für die umgebende Box zu lang ist.

[Interaktives Beispiel](#)

Der Inhaltsbereich: Textüberlappung

HTML

```

1 <p class="box1">
2   Das ist ein viel zu langer Text.
3 </p>
4 <p class="box2">
5   Das ist auch ein Text, der für die
   umgebende Box zu lang ist.
6 </p>

```

CSS

```

1 .box1 {
2   background: salmon;
3   width: 50px;
4   height: 50px;
5 }
6 .box2 {
7   background: lightskyblue;
8   width: 120px;
9   height: 60px;
10 }
11

```

Das ist ein viel zu langer Text, der für die umgebende Box zu lang ist.

[Interaktives Beispiel](#)

Der Inhaltsbereich: Overflow

- › Die Eigenschaft **overflow** wird benutzt, um die Darstellung der Inhalte zu steuern, die für ihre umgebende Box zu lang sind.
- › Mögliche Werte:
 - › **visible**: der Inhalt wird außerhalb der Box angezeigt. Dadurch überlappt er eventuell andere Inhalte (Standardwert).
 - › **hidden**: der Inhalt wird an der Boxgrenze abgeschnitten und ist nicht sichtbar.
 - › **scroll**: ein Scrollbalken wird eingeblendet, so dass
 - › der Inhalt in der Box passend angezeigt wird und
 - › die bei **hidden** versteckten Inhalte durch Scrollen erreichbar werden.

Der Inhaltsbereich: Overflow

```

HTML
1 <p class="box1">
2   Das ist ein viel zu langer Text.
3 </p>
4 <p class="box2">
5   Das ist auch ein Text, der für die
   umgebende Box zu lang ist.
6 </p>

CSS
1 .box1 {
2   background: salmon;
3   width: 70px;
4   height: 50px;
5   overflow: hidden;
6 }
7 .box2 {
8   background: lightskyblue;
9   width: 150px;
10  height: 50px;
11  overflow: scroll;
12 }
  
```

Das ist ein
viel zu
langer

Das ist auch ein Text,
der für die

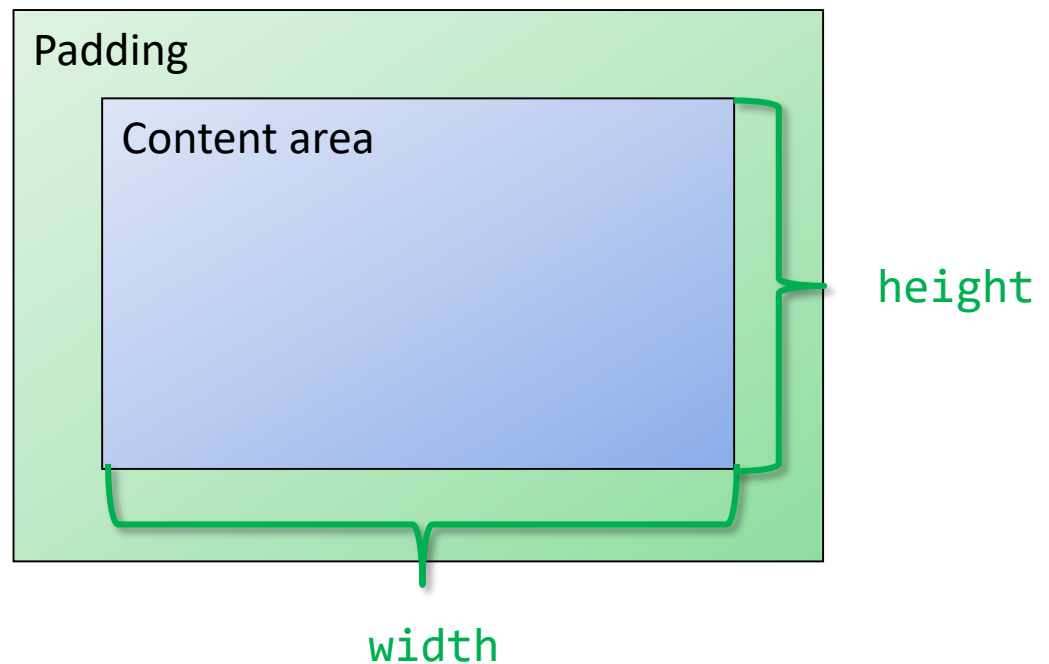
[Interaktives Beispiel](#)

Der Inhaltsbereich: Minimale und maximale Größe

- › Neben einer genauen Angabe der Breite und Höhe einer Box können auch *begrenzende Angaben* vorgenommen werden.
- › Diese spielen vor allem dann eine Rolle, wenn die HTML-Seite auf unterschiedlich großen Bildschirmen dargestellt werden soll.
- › Mindestangaben:
 - › Eine Box soll nie schmaler / niedriger als ein vorgegebener Wert sein: **min-width, min-height**
- › Maximalangaben:
 - › Eine Box soll nie breiter / höher als ein vorgegebener Wert sein: **max-width, max-height**

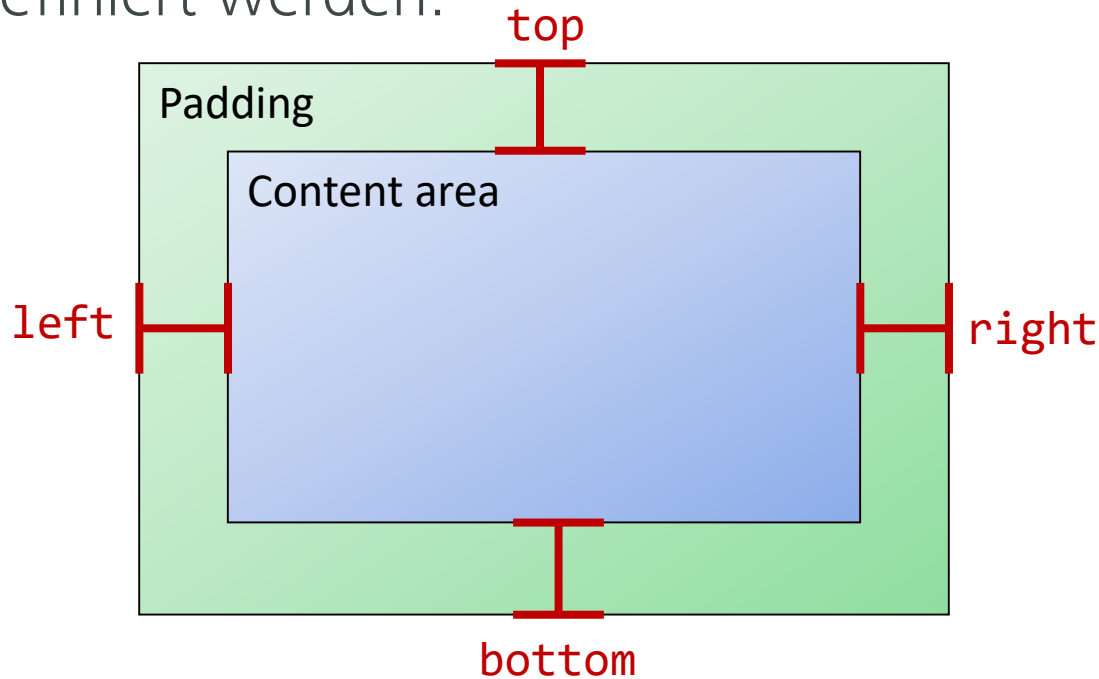
Padding

- › Das *Padding* (Innenrand, Polster) definiert einen Zwischenraum zwischen dem Inhaltsbereich und der Elementgrenze.



Padding

- › Das *Padding* kann für jede Seite des Inhaltsbereiches separat definiert werden.



Gesamtbreite: $\text{width} + \text{padding-left} + \text{padding-right}$
 Gesamthöhe: $\text{height} + \text{padding-top} + \text{padding-bottom}$

Padding

- › Die CSS-Eigenschaft **padding** beschreibt den Abstand des Inhaltsbereichs vom jeweiligen Rand:
 - › **padding-top**: Abstand vom oberen Rand der Content-Box
 - › **padding-right**: Abstand vom rechten Rand der Content-Box
 - › **padding-bottom**: Abstand vom unteren Rand der Content-Box
 - › **padding-left**: Abstand vom linken Rand der Content-Box
- › **padding**:
 - › Zusammenfassung der oberen vier Eigenschaften
 - › Reihenfolge: im Uhrzeigersinn: oben – rechts – unten – links
 - › Abkürzungen:
 - › Ein Wert gilt für alle Seiten: **padding: 10px**
 - › Zwei Werte gelten für oben/unten bzw. rechts/links **padding: 10px 20px**
 - › Drei Werte gelten für oben, rechts/links bzw. unten **padding: 10px 20px 30px**

Boxen ohne und mit Padding

```

CSS
1 ▾ .box1 {
2   background: salmon;
3   width: 70px;
4 }
5
6 ▾ .box2 {
7   background: lightskyblue;
8   width: 150px;
9 }
10

```

Das ist ein Text in einer Box ohne Padding.

Das ist auch ein Text in einer Box ohne Padding.

Interaktives Beispiel

```

CSS
1 ▾ .box1 {
2   background: salmon;
3   width: 70px;
4   padding: 10px;
5 }
6
7 ▾ .box2 {
8   background: lightskyblue;
9   width: 150px;
10  padding: 30px 40px;
11 }

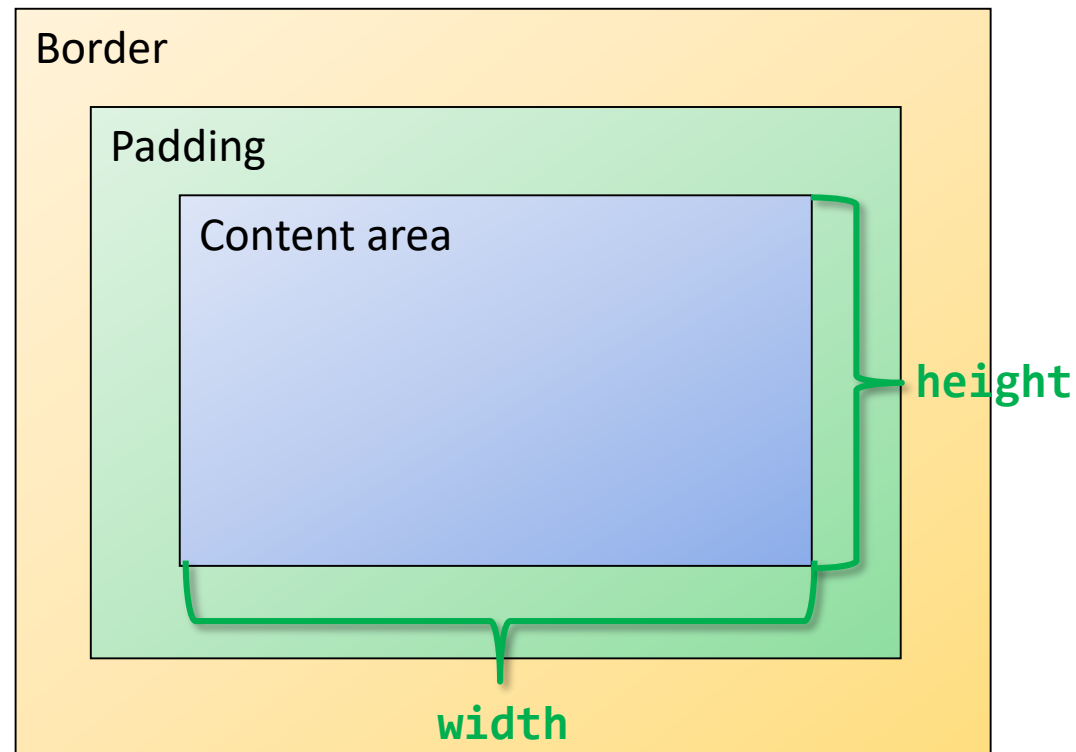
```

Das ist ein Text in einer Box mit einem geringem Padding.

Das ist ein Text in einer Box mit einem hohem Padding.

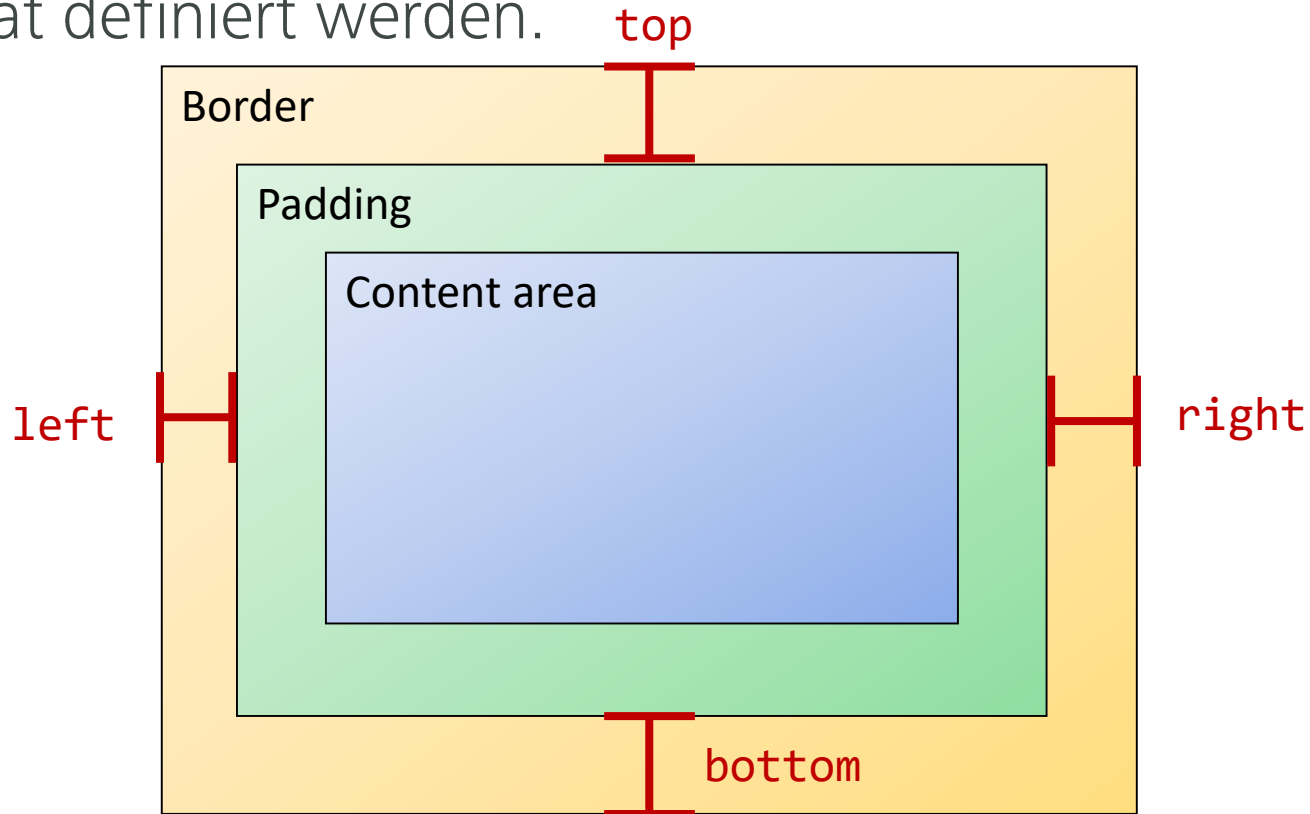
Border

- › Der *Border* (Rahmen) zeichnet einen Rahmen um den Inhaltsbereich und seinen Innenabstand herum.



Border

- › Auch der *Border* kann für jede Seite der zugehörigen Box separat definiert werden.



Border

- › Die CSS-Eigenschaft **border** beschreibt die vier Seiten des Boxrahmens (**border-top**, **border-right**, **border-bottom**, **border-left**)
- › Sie definiert für jede Seite des Rahmens drei zusätzliche Werte:

border	-width	-color	-style (solid, dashed, dotted, ...)
-top	Breite der oberen Linie	Farbe der oberen Linie	Stil der oberen Linie (durchgezogen, gestrichelt, gepunktet, ...)
-right	Breite der rechten Linie	Farbe der rechten Linie	Stil der rechten Linie
-bottom	Breite der unteren Linie	Farbe der unteren Linie	Stil der unteren Linie
-left	Breite der linken Linie	Farbe der linken Linie	Stil der linken Linie

Border

- › Der Rand kann entweder
 - › durch eine Aufzählung aller Eigenschaften als einzelne Regeln oder
 - › in zusammengefasster Form definiert werden

Border: Beispiel

HTML

```
1 <p class="box1">
2   Erste Box mit einem linken, gestrichelten Rand
3 </p>
4 <p class="box2">
5   Zweite Box mit einem linken, gestrichelten Rand
6 </p>
7 <p class="box3">
8   Dritte Box mit einem vollständigen, gestrichelten Rand
9 </p>
```

CSS

```
1 .box1 {
2   border-left-width: 4px;
3   border-left-color: blue;
4   border-left-style: dashed;
5 }
6 .box2 {
7   border-left: 4px blue dashed;
8 }
9 .box3 {
10  border: 4px blue dashed;
11 }
```

Erste Box mit einem linken, gestrichelten Rand

Zweite Box mit einem linken, gestrichelten Rand

Dritte Box mit einem vollständigen, gestrichelten Rand

[Interaktives Beispiel](#)

Border

- › Für die Breite der Linie können feste Pixelangaben, aber keine Prozentangaben gemacht werden.
- › Für die Breite der Linie können auch vier vordefinierte Werte benutzt werden:
 - › `thin`
 - › `medium`
 - › `thick`
 - › `inherit`
- › Für die Eigenschaft `border-style` gibt es 9 vordefinierte Werte:
 - › `none`
 - › `solid` / `dashed` / `dotted`
 - › `double`
 - › `groove` / `ridge` / `inset` / `outset`

Border: Weitere Eigenschaften

- › Mit der Eigenschaft **border** können auch abgerundete Ecken – und im Extremfall runde Boxen – definiert werden.

```

HTML
1 <div id="box1"></div>
2 <div id="box2"></div>

CSS
1 #box1 {
2   background-color: tomato;
3   border: 4px tomato solid;
4   border-top-left-radius: 10px;
5   border-top-right-radius: 30px;
6 }
7
8 #box2 {
9   background-color: lightskyblue;
10  border: 4px lightskyblue solid;
11  border-radius: 50%;
12 }
13
14 div {
15   height: 100px;
16   width: 100px;
17   margin: 20px;
18 }

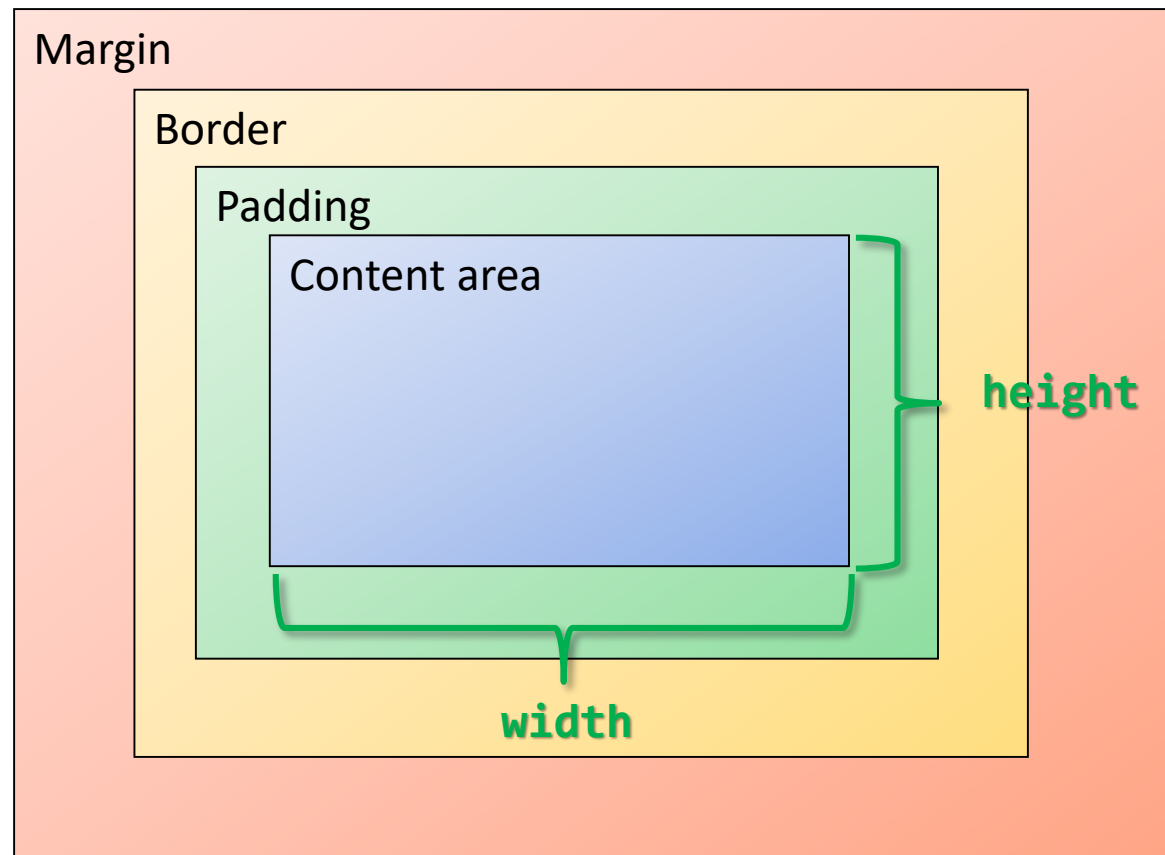
```



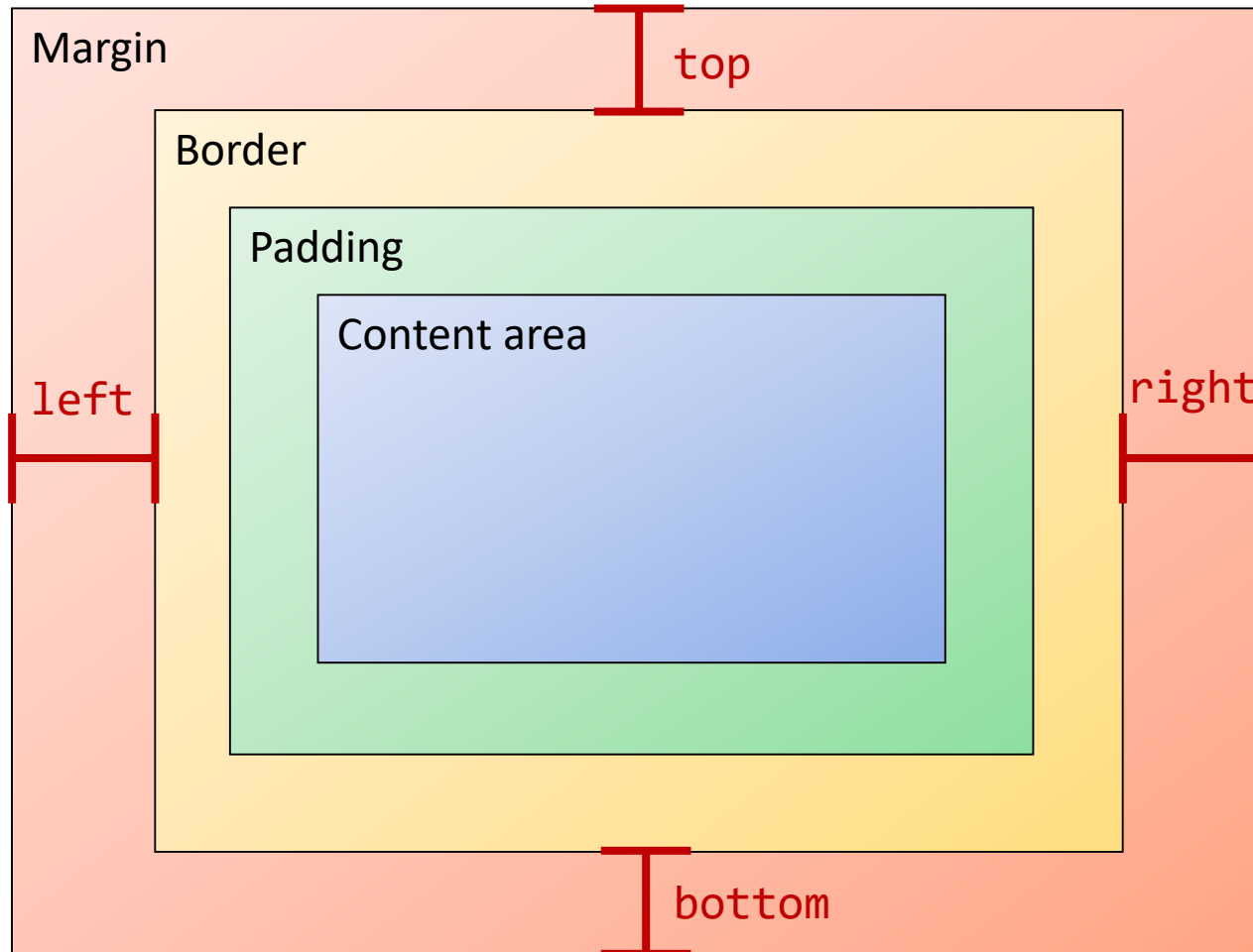
[Interaktives Beispiel](#)

Margin

- › Der *Margin* (Außenrand) definiert den äußeren, nicht beschreibbaren Rand einer Box.
- › Der Margin wird benutzt, um Abstände zwischen Elementen zu definieren.



Margin



Gesamtbreite: $\text{width} + \text{padding-left} + \text{padding-right} + \text{border-left} + \text{border-right} + \text{margin-left} + \text{margin-right}$

Gesamthöhe: $\text{height} + \text{padding-top} + \text{padding-bottom} + \text{border-top} + \text{border-bottom} + \text{margin-top} + \text{margin-bottom}$

Margin

- › Die Eigenschaft **margin** wird für die Definition des (nicht beschreibbaren) Außenrands benutzt:
 - › **margin-top**: Abstand vom oberen Rand der Box
 - › **margin-right**: Abstand vom rechten Rand der Box
 - › **margin-bottom**: Abstand vom unteren Rand der Box
 - › **margin-left**: Abstand vom linken Rand der Box
- › **margin**:
 - › Zusammenfassung der oberen vier Eigenschaften
 - › Reihenfolge: im Uhrzeigersinn: oben – rechts – unten – links
 - › Abkürzungen: wie bei **padding**

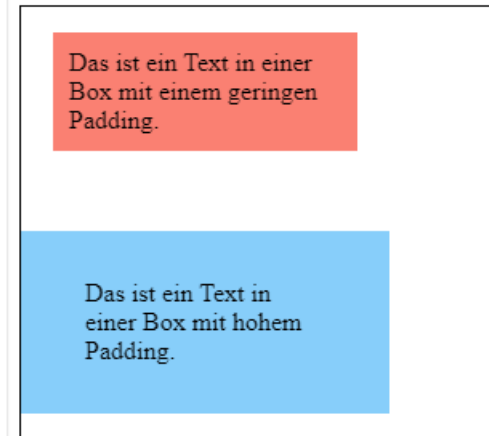
Margin - Beispiel

```

HTML
1 <body>
2 <p class="box1">
3   Das ist ein Text in einer Box mit einem geringen Padding.
4 </p>
5 <p class="box2">
6   Das ist ein Text in einer Box mit hohem Padding.
7 </p>
8 </body>

CSS
1 .box1 {
2   background: salmon;
3   width: 170px;
4   padding: 10px;
5   margin-left: 20px;
6 }
7
8 .box2 {
9   background: lightskyblue;
10  width: 150px;
11  padding: 30px 40px;
12  margin-top: 50px;
13 }
14
15 body {
16  height: 200px;
17  width: 300px;
18  border: 1px black solid;
19 }

```



[Interaktives Beispiel](#)

Inhalte

- › Exkurs: Maßeinheiten in CSS
- › Gestaltung der HTML-Boxen mit CSS
- › HTML: Hilfsboxen

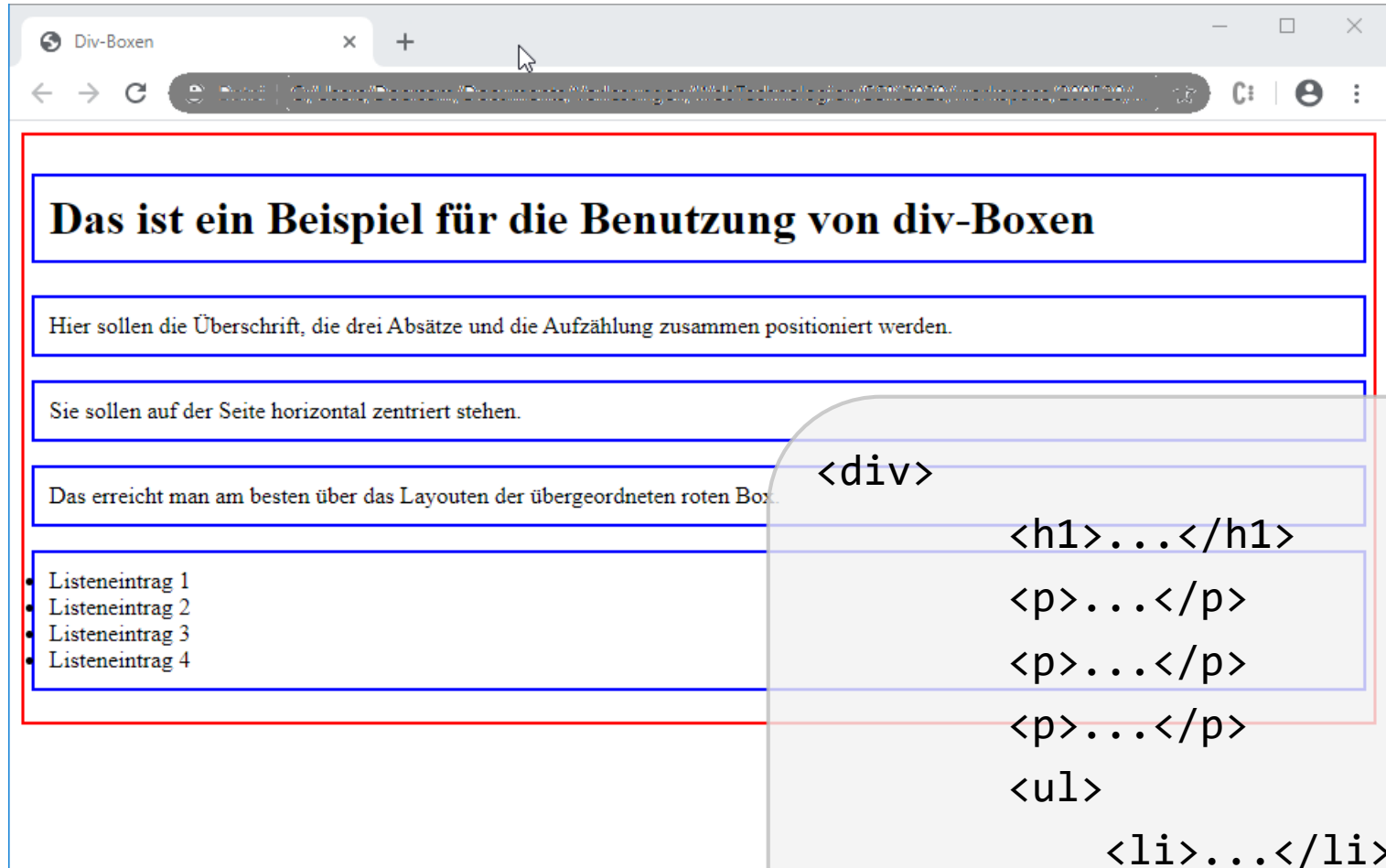
Boxen und HTML-Elemente

- › Teilweise will man zwar Boxen-Eigenschaften benutzen bzw. verändern, aber keine semantischen HTML-Elemente wie `<p>` oder `<h1>` usw. benutzen.
- › Dafür stehen zwei HTML-Elemente zur Verfügung, die lediglich Boxen für ihre Inhalte definieren:
 - › `<div>`: Definition einer Box als ein Block-Element
 - › `<span>`: Definition einer Box als ein Inline-Element
- › Die beiden Elemente werden meistens dazu benutzt, *Layouteigenschaften* von Dokumentenabschnitten zu definieren.

HTML-Element `<div>`

- › `<div>` steht für *division* (Abteilung / Bereich)
- › Zweck eines `<div>`-Elementes:
 - › Behälter für andere Elemente (z.B. Text, Bilder, Listen, ...) in *einen gemeinsamen Bereich*
 - › Basis für:
 - › das Formatieren von optisch zusammengehörigen Elementen,
 - › das Positionieren von Gruppen von HTML-Elementen,
 - › Effekte in der Darstellung und Animationen (meist in Verbindung mit JavaScript)
- › Ein `<div>`-Element ist ein Block-Element und hat ansonsten keine besonderen Eigenschaften.

HTML-Element `<div>` - Beispiel ohne CSS



Das ist ein Beispiel für die Benutzung von div-Boxen

Hier sollen die Überschrift, die drei Absätze und die Aufzählung zusammen positioniert werden.

Sie sollen auf der Seite horizontal zentriert stehen.

Das erreicht man am besten über das Layouten der übergeordneten roten Box.

- Listeneintrag 1
- Listeneintrag 2
- Listeneintrag 3
- Listeneintrag 4

```
<div>
```

```
<h1>...</h1>
```

```
<p>...</p>
```

```
<p>...</p>
```

```
<p>...</p>
```

```
<ul>
```

```
<li>...</li>
```

```
...
```

```
</ul>
```

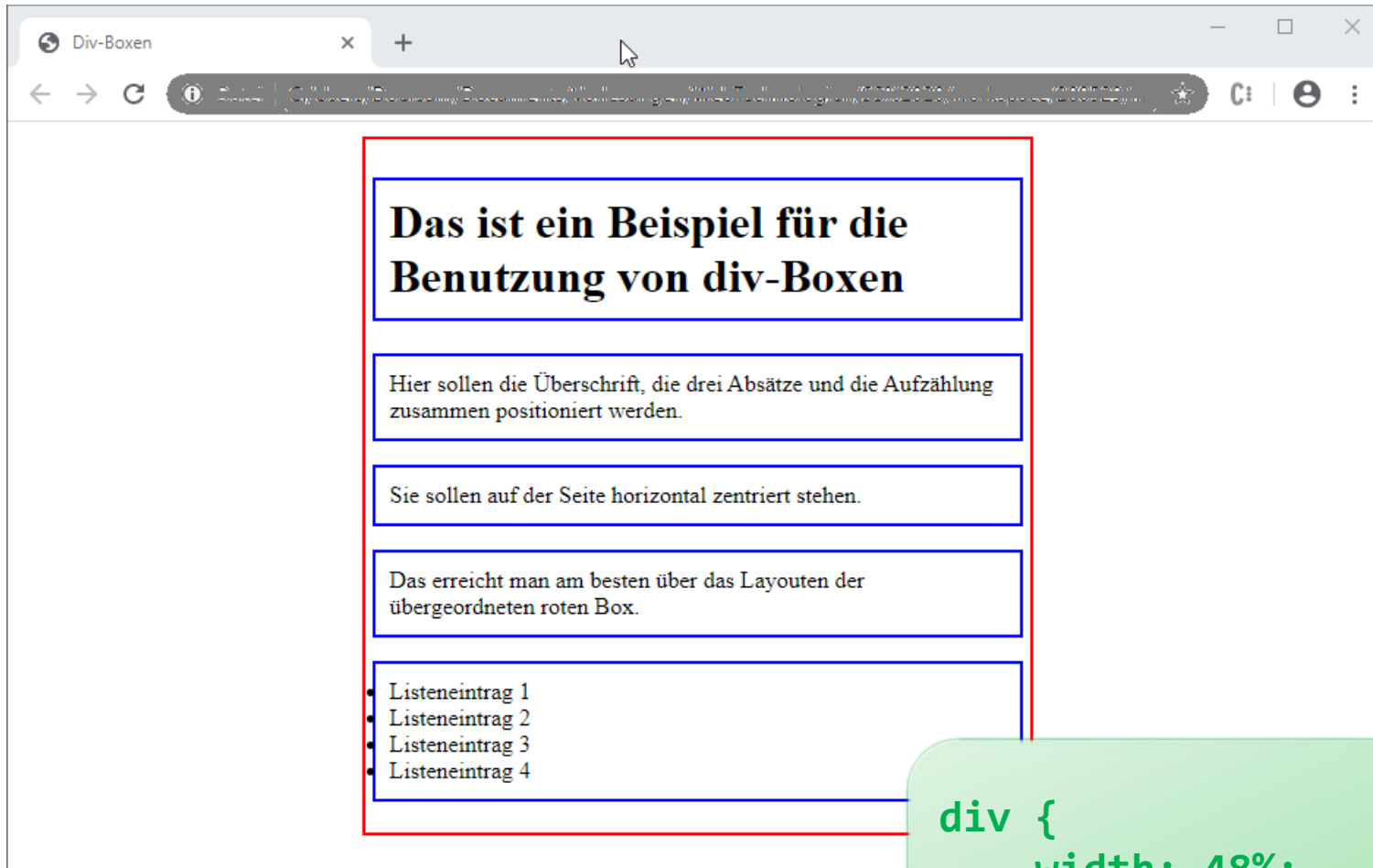
```
</div>
```

HTML-Element `<div>` - Beispiel

```
<div>
    <h1>...</h1>
    <p>...</p>
    <p>...</p>
    <p>...</p>
    <ul>
        <li>...</li>
        ...
    </ul>
</div>
```

```
div {
    width: 48%;
    margin: 10px auto;
}
```

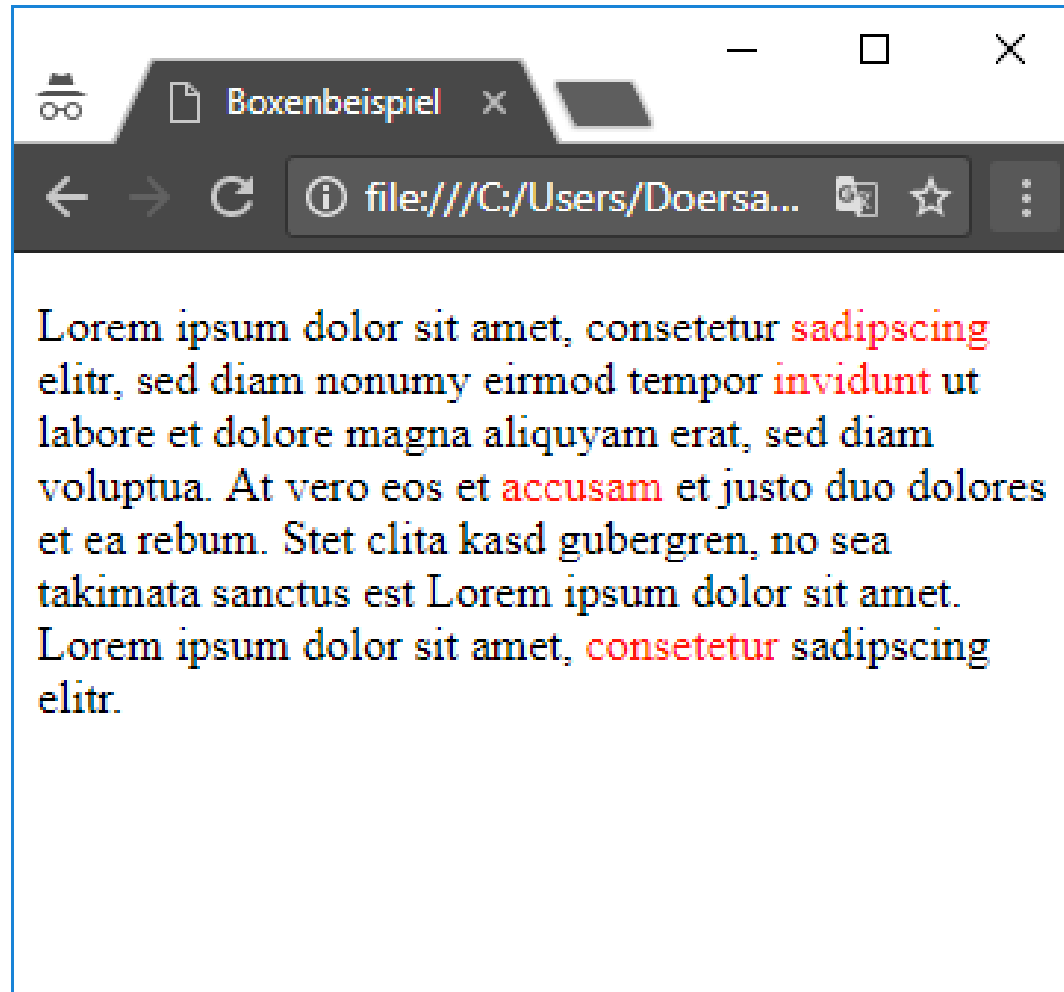
HTML-Element `<div>` - Beispiel: Zentrieren ALLER Boxen



```
div {  
  width: 48%;  
  margin: 10px auto;  
}
```

[Interaktives Beispiel](#)

HTML-Element `<span>` - Beispiel



HTML-Element `<span>`

- › `<span>` steht für *to span* (überspannen)
- › Zweck eines `<span>`-Elementes:
 - › Einschließen mehrerer Elemente (meistens Text, seltener Inline-Elemente) in *einen gemeinsamen Bereich*
 - › Dadurch werden Bereiche erzeugt, die mit CSS formatiert werden können.
- › Ein `<span>`-Element ist ein Inline-Element und hat ansonsten keine besonderen Eigenschaften.

HTML-Element `<span>` - Beispiel

`<p>`

Lorem ipsum dolor sit amet,
consetetur `<span>sadipscing</span>`
elit, sed diam nonumy eirmod
tempor `<span>invidunt</span>` ut
labore et dolore magna aliquyam
erat, sed diam voluptua. At vero
eos et `<span>accusam</span>` et
justo duo dolores et ea rebum.
Stet clita kasd gubergren, no sea
takimata sanctus est Lorem ipsum
dolor sit amet. Lorem ipsum dolor
sit amet, `<span>consetetur</span>`
sadipscing elit.

`</p>`

[Interaktives Beispiel](#)

```
span {  
    color: red;  
}
```