



openHPI CSS – Positionierung und Box-Modell

Prof. Dr. Christoph Meinel

Hasso-Plattner-Institut

Universität Potsdam

CSS Positionierung (1/2)

- Mittels **Positionierung** können Elemente innerhalb eines HTML-Dokumentes angeordnet werden
- Auch festgelegt wird, wie Elemente von Text umflossen werden
- Mit `position` können Elemente beliebig positioniert und aus dem regulären Elementfluss entfernt werden:
 - **absolute**
 - Position bezieht sich auf Vorfahren-Element oder Wurzelement
 - losgelöst vom Textfluss
 - Position per `left`, `right`, `top` und `bottom`
 - ...

CSS Positionierung (2/2)

- **fixed**
 - Starre Ausrichtung ohne Abhängigkeiten
 - Losgelöst vom Textfluss
 - Position per `left`, `right`, `top` und `bottom`
- **relative**
 - Bezugspunkt für absolut positionierte Kind-Elemente
 - Bleibt im Textfluss
 - Position per `left`, `right`, `top` und `bottom`
- **static**
 - Standardwert der Eigenschaft `position`
 - Bleibt im Textfluss
 - `left`, `right`, `top` und `bottom` werden ignoriert

CSS Positionierung – Beispiel

■ CSS Stylesheet:

```
div.relative {
    position: relative;
}
div.absolute {
    position: absolute;
    top: 50px;
    right: 0;
}
```

■ Ausgabe:

Dieses div hat die Position relative.

Dieses div hat die Position absolute.

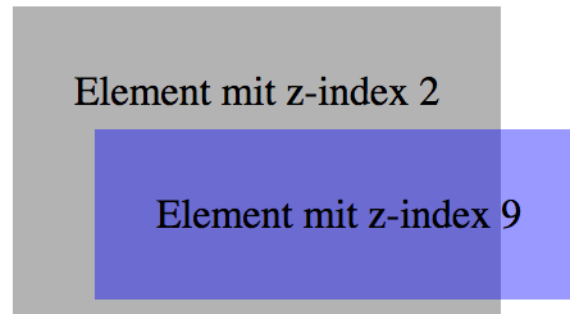
■ HTML:

```
<div class="relative">Dieses div hat die Position relative.
  <div class="absolute">Dieses div hat die Position absolute.</div>
</div>
```

CSS Positionierung – z-index

- Die **z-index** Eigenschaft kommt zur Anwendung, wenn sich Elemente überlappen
- Je größer der Wert, desto „höher“ liegt das Element und überdeckt andere Elemente
- Kann nur für Elemente angegeben werden, die eine **position** definiert haben und nicht **static** sind
- **Beispiel (CSS):**

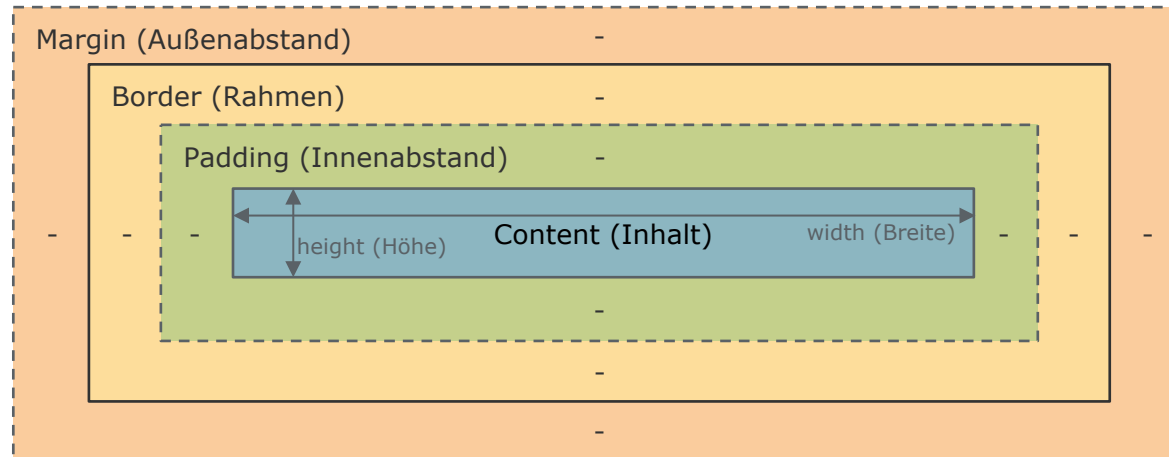
```
#element1 {
    z-index: 2;
}
#element2 {
    z-index: 9;
}
```



Box-Modell – Einführung (1/2)

- HTML-Elemente, wie z.B. `<div>`, werden durch innerhalb von Rechtecken dargestellt
- Diese Rechtecke werden durch das **Box-Modell** beschrieben
- Bestandteile einer Box sind:
 - **Inhalt** – Texte und Bilder
 - **Innenabstand** (padding)
 - **Rahmen** (border)
 - **Außenabstand** (margin)
- Die Innen- und Außenabstände sowie der Rahmen können für alle vier Seiten der Box individuell festgelegt werden

Box-Modell – Einführung (2/2)



■ Die einzelnen Bereiche werden bezeichnet als:

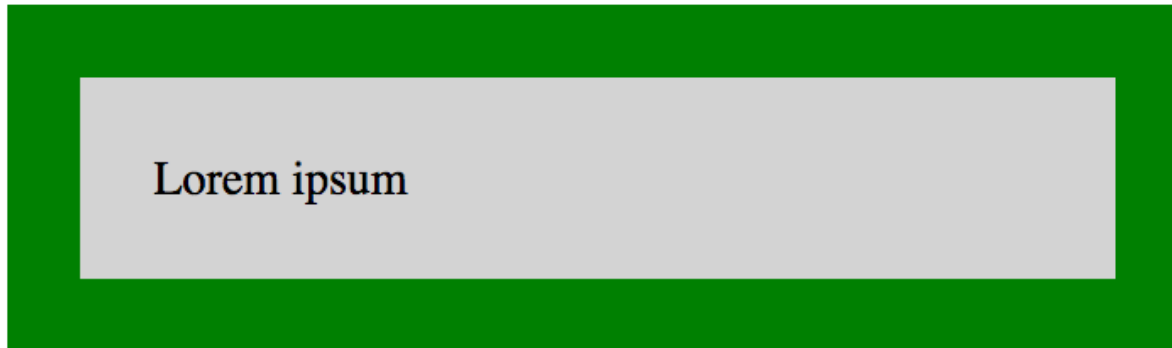
- Content-Box
- Padding-Box
- Border-Box
- Margin-Box

Box-Modell – Beispiel

■ CSS Stylesheet:

```
div {  
  background-color: lightgrey;  
  width: 300px;  
  padding: 25px;  
  border: 25px solid green;  
  margin: 25px;  
}
```

■ Ausgabe:



Box-Modell – Innen- und Außenabstand

- Abstände einzeln definierbar mit:

<code>padding-top</code>	<code>margin-top</code>
<code>padding-right</code>	<code>margin-right</code>
<code>padding-bottom</code>	<code>margin-bottom</code>
<code>padding-left</code>	<code>margin-left</code>

- Oder als zusammenfassende Eigenschaft:

- Bei einer Angabe gilt der Wert für alle vier Seiten

```
padding: 10px;
```

```
margin: 10px;
```

- Zwei bis vier Angaben: 1. Wert für top, 2. Wert für right, 3. Wert für bottom, 4. Wert für left

```
padding: 10px 5px;
```

```
margin: 10px 5px 10px 5px;
```

- Als Werte sind nicht negative numerische Längenmaße erlaubt, z.B. Pixel (**px**), Punkt (**pt**), Zentimeter (**cm**), Millimeter (**mm**), ...

Box-Modell – Rahmen

- Eigenschaft `border` bestimmt Rahmendicke, Rahmentyp und Rahmenfarbe, z.B. `25px solid green;`
- Alle vier Seiten sind ebenfalls einzeln definierbar, z.B.
 - `border-top` `border-right`
 `border-bottom` `border-left`
- Rahmendicke: `border-width`
 - Numerischer Wert (z.B. `25px`) oder
 `thin` (dünn), `medium` (mittel), `thick` (dick)
- Rahmentyp: `border-style`
 - `none` (transparent), `dotted` (gepunktet), `dashed` (gestrichelt),
 `solid` (durchgezogen), `double` (doppelt durchgezogen), ...
- Rahmenfarbe: `border-color`
 - `transparent` oder Farbangabe (z.B. `#`