

Kapitel 12 - Teil 3

Kästchen messen

Inhalte in diesem Kapitel

- Ausmaße von Elementen sowie Abstände bestimmen
 - Schatten und abgerundete Ecken definieren
 - Button mit animierten Hoverzustand erstellen
-

Ausmaße von Elementen

- Blockelemente:
 - Absätze, Überschriften, div, article, usw.
 - Nehmen allen Platz in der Breite ein, den sie bekommen

Breite

- Breite definieren über `width`
- Alle gängigen Einheiten nutzbar
- Häufig mit Prozent arbeiten da sehr flexibel
 - Breite 50% richtet sich immer am umgebenen Element
 - Bsp: Umliegendes Element 600px Breite, innenliegendes Element 50% => 300px Breite
- Zusätzlich: `min-width` & `max-width`
 - Sinnvoll in Kombination mit relativer Breitenangabe
- Auch sinnvoll bei `max-width` mit `rem` arbeiten

```
width: 80%;  
min-width: 400px;  
max-width: 800px;
```

Höhe

- Höhe definieren über `height`, `min-height` und `max-height`
- Wird selten gebraucht da Elemente automatisch so hoch sind, wie Inhalt fordert
- Sinnvoll für
 - kleine Symbole
- Kombination `min-height` und `vh`

Innenabstand

- Innenabstand: padding
- Pixel nur in Ausnahmefällen

```
padding-top | padding-right | padding-bottom | padding-left
padding (alle Seiten)
padding: 1px 2px 3px 4px; // oben, rechts, unten, links
padding: 6px (oben/unten) 8px (rechts/links);
padding: 6px (oben) 8px (rechts/links) 9px (unten);
```

Verkürzte Angaben mit mehreren Werten

	Beispiel	Erklärung	Auflösung
1 Wert	padding: 50px;	Angegebener Wert gilt für alle 4 Seiten	padding-top: 50px padding-right: 50px; padding-bottom: 50px; padding-left: 50px;
2 Werte	padding: 20px 12px;	1. Wert: oben/unten 2. Wert: rechts/links	padding-top: 20px padding-right: 12px; padding-bottom: 20px; padding-left: 12px;
3 Werte	padding: 20px 12px 10px;	1. Wert: oben 2. Wert: rechts/links 3. Wert: unten	padding-top: 20px padding-right: 12px; padding-bottom: 10px; padding-left: 12px;
4 Werte	padding: 20px 40px 4px 12px;	1. Wert: oben 2. Wert: rechts 3. Wert: unten 4. Wert: links	padding-top: 20px padding-right: 40px; padding-bottom: 4px; padding-left: 12px;

- Gilt für margin, padding
- Eselsbrücke: Kompass

Rahmen

- Anweisung: border

```
border: 2px solid blue; // none
=> ausgeschrieben
border-width: 2px;
border-style: solid;
// dotted, dashed, solid, double, groove, ridge, inset, outset
border-color: blue;
=> einzelne Seiten
border-top | border-right | border-bottom | border-left
```

- Witzige Tricks: <https://css-tricks.com/snippets/css/css-triangle/>

Überfließen steuern mit overflow

- Element benötigt mehr Platz als zur Verfügung steht: `overflow`

```
overflow: hidden // Inhalte werden abgeschnitten
overflow: scroll  // Inhalte sind mit Scrollbalken erreichbar
overflow: visible // Inhalte sind sichtbar (Standardwert)
overflow-x | overflow-y
```

-

Außenabstand

- Abstand zwischen Elementen: `margin`

Grundprinzip von margin

- Gleiches Prinzip wie `padding`

```
margin: 0 auto;
```

- `auto` für links und rechts => zentriert Elemente

Zusammenfallende Außenabstände

- manche Abstände fallen zusammen
 - unterer Abstand auf oberen Abstand trifft => frisst/absorbiert der größere den kleineren, vereinen sich
 - collapsing margin nur bei Abständen nach oben und unten, nicht bei seitlichen Abständen
- seltsame Angewohnheiten in CSS
 - bei verschachtelten Elementen kann der Außenabstand zusammenfallen
 - Bsp.: in einem `article`-Element eine `h1`-Überschrift so reicht die Überschrift ihren `margin` an das äußere Element weiter
 - Verhalten unterbinden mit Rahmen oder Innenabstand

box-sizing und die Gesamtgröße

- BSP: Box mit folgenden Abmessungen

```
width: 250px;
padding: 50px;
border: 20px solid #000;
margin: 50px;
```

- => $250\text{px} + 2 \cdot 50\text{px} + 2 \cdot 20\text{px} = \text{insg. } 390\text{px}$ Breite
=> `width` umfasst nur den inneren Bereich (250px) Element ist aber 390px breit

```
box-sizing: border-box;
```

- => praktischer, wenn `width` Gesamtbreite bezeichnet und `padding` und `border` mit einschließt
=> `border-box`

Abgerundet: border-radius

- Anweisung: border-radius
- Einheit: rem, px, Prozent, ...

```
border-radius  
border-top-left-radius // oben links  
border-top-right-radius // oben rechts  
border-bottom-right-radius // unten rechts  
border-bottom-left-radius // unten links
```

- Für einen Kreis
 - width & height gleicher Wert
- ```
border-radius: 50%;
```

## Schatten um Boxen zeichnen: box-shadow

- Anweisung: text-shadow & box-shadow
- text-shadow für Textschatten
- box-shadow für Boxschatten

```
box-shadow: #3377decc 10px 20px 4px;
text-shadow: #3377decc 10px 20px 4px;
```

- Die Angaben:
    - Farbe des Schattens (kann auch am Ende stehen)
    - 1. Angabe: Verschiebung von links (10px)
    - 2. Angabe: Verschiebung von oben (20px)
    - 3. Angabe: Grad der Verschwommenheit für den Schatten
- ```
box-shadow: #3377decc 10px 20px 4px inset;
```
- 5. Angabe: Schatten wird nach innen gezeichnet

width, padding, border, margin bei einem Inline-Element

- Eigenschaften wirken bei Inline-Elementen anders
 - margin wirkt nach rechts und links, aber nur in der Zeile
 - border und padding wirken relativ normal
 - width hat keine Auswirkung
- Bsp für Inline-Elemente:
 - Image
 - Link
 - strong,
 - span,

Verwandlungskünstler: display

- Anweisung: display

```
display: block; // macht aus einem Element ein Blockelement
display: inline; // macht aus einem Element ein Inline-Element
display: inline-block; // Mischung aus Inline- und Blockelement
```

- Angaben wie bei einem Blockelement zuweisen, aber mehrere Inline-Blockelemente werden nebeneinander dargestellt
- Bsp.: Buttons nebeneinander mit Abstand

```
display: none; // Element wird ausgeblendet
display: flex; // Flexlayout wird aktiviert (Kapitel 15)
display: grid; // Gridlayoutmodul wird aktiviert (Kapitel 15)
```

display bei der Erstellung von Buttons

- Anweisung: transition
- **Bsp.: Folien\Basisausbildung mit Dummy Bücher\Beispiele HTML CSS\kapitel_12\ a-button.html**

```
transition: all 0.2s ease;
=>
transition-property: all; // Alle Eigenschaften
transition-duration: 0.2s;
transition-timing-function: ease;
```

- Weitere Möglichkeiten:
<https://developer.mozilla.org/de/docs/Web/CSS/transition>

Elemente drehen, verschieben und skalieren

- Anweisung: transform
 - Elemente drehen: 30 Grad im Uhrzeigersinn

```
transform: rotate(30deg);
```
 - Elemente skalieren: vergrößert Doppelt

```
transform: scale(2);
```
 - Elemente verschieben: um 10 px nach links und 50px von oben

```
transform: translate(10px, 50px);
```