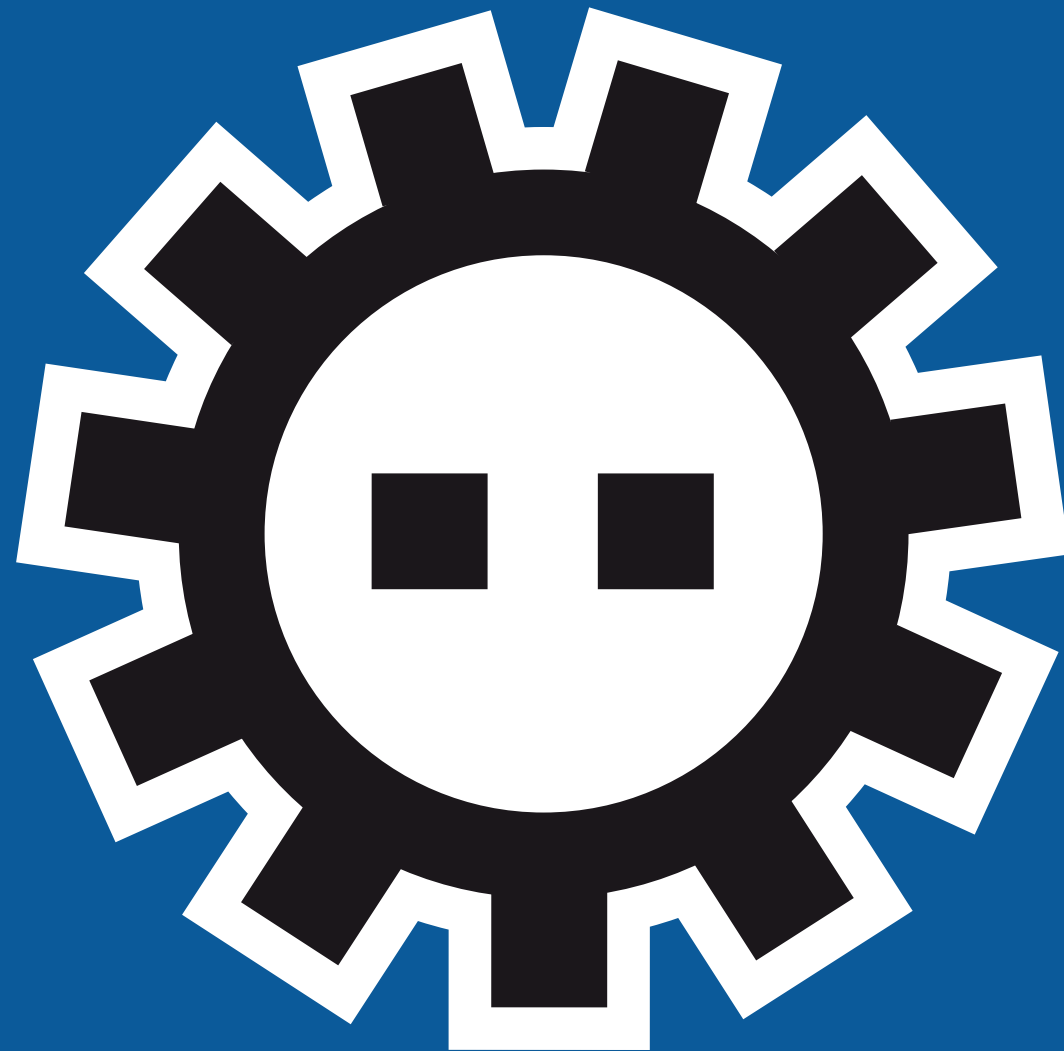


SOFTWARE ENGINEERING 2

05 - CSS3

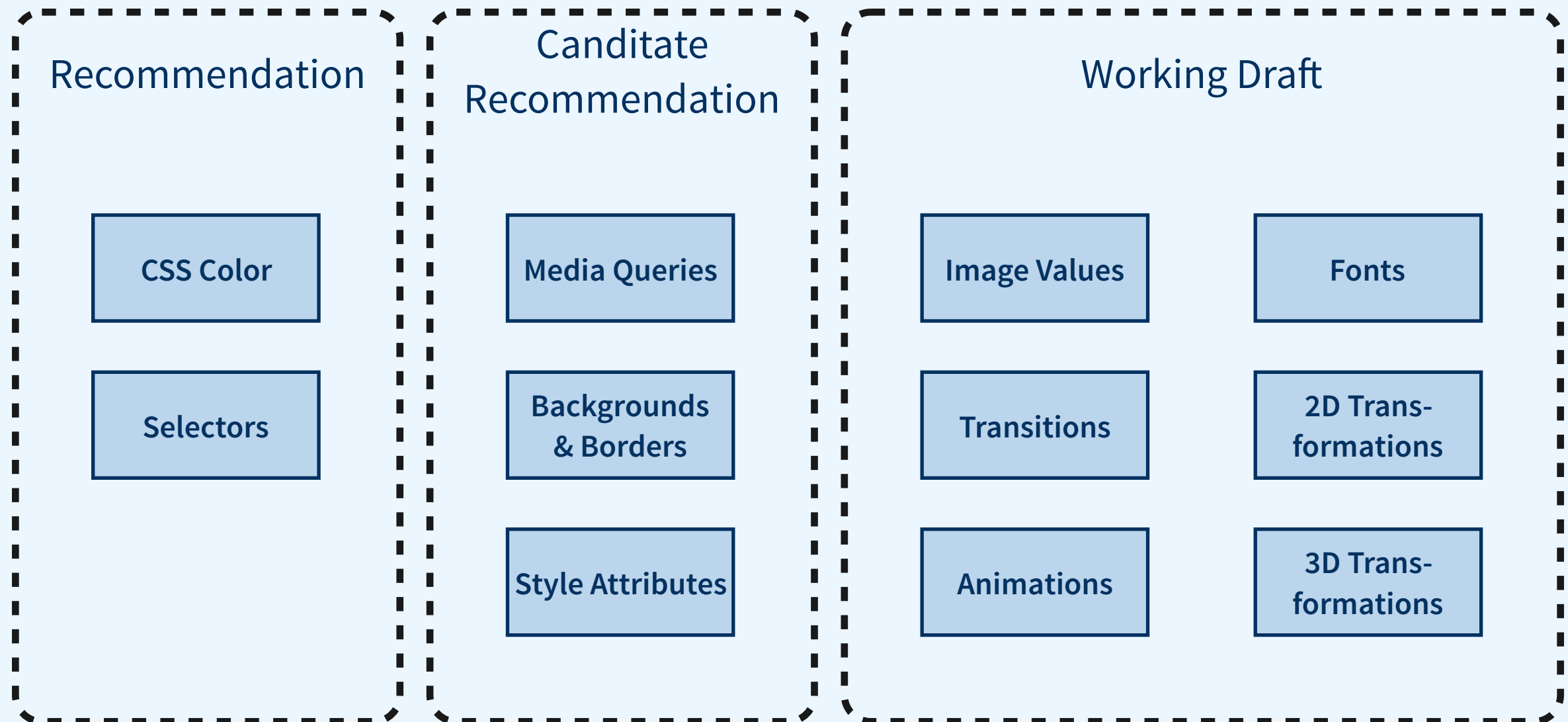


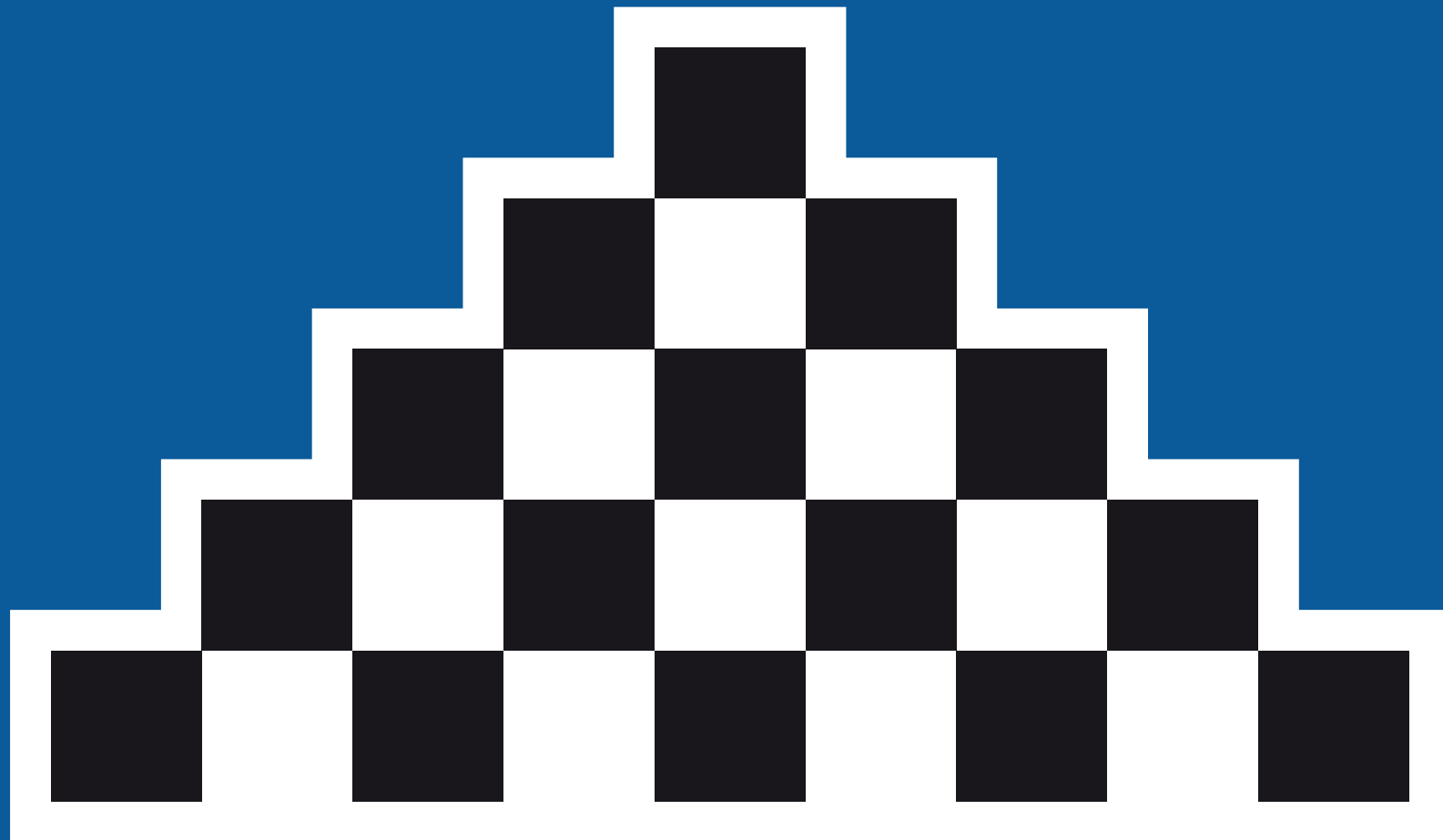
MOTIVATION

Motivation

- CSS dient zur Definition von Gestaltungseigenschaften
 - ▶ Farben, Form, Schriftarten, Abstände
- Die (aktuelle) Version CSS 2 stammt noch aus dem Jahr 1998
 - ▶ die Version CSS3 ist seitdem in der Entwicklung
 - ▶ ähnliche Probleme wie Übergang von HTML 4 nach HTML 5
- Der Sprachumfang wird immer größer
 - ▶ Aufteilung des Standards in Module
 - ▶ jedes mit einer eigenen Arbeitsgruppe und eigener Zeitplanung
 - ▶ Browser unterstützen bereits viele Module

CSS3 Module





GRUNDLAGEN

Layout

- Bis jetzt haben wir CSS nur für die optische Gestaltung einzelner Elemente eingesetzt
- Jetzt folgen CSS Techniken, um das Layout der gesamten Seite zu verändern
 - ▶ Box-Modell
 - ▶ Fluss der Elemente
 - ▶ Positionierung von Boxen

Die Anatomie eines Block-Elements

Boxmodell

Boxmodell

- Jedes Block-Element (p, div, h1, etc.) besteht aus folgenden Bereichen:
 - ▶ Inhaltsbereich
 - ▶ Rahmen
 - ▶ Innenabstand
 - ▶ Außenabstand
- Vergleichbar mit Tabellenzellen, jetzt aber für alle Block-Elemente

Boxmodell

Der Inhalt eines Blockelements wird immer von einem rechteckigen Bereich umschlossen. Dies ist der Inhaltsbereich.

Innenabstand

Rahmen

Außenabstand

Boxmodell

- Inhaltsbereich (immer vorhanden)
 - ▶ Kasten um den Inhalt des Elements
- Innenabstand (optional)
 - ▶ Breite anpassbar, Hintergrundfarbe wie Inhaltsbereich
- Rahmen (optional)
 - ▶ Farbe, Breite und Stil anpassbar
- Außenabstand (optional)
 - ▶ Breite anpassbar, immer transparent

Abstände

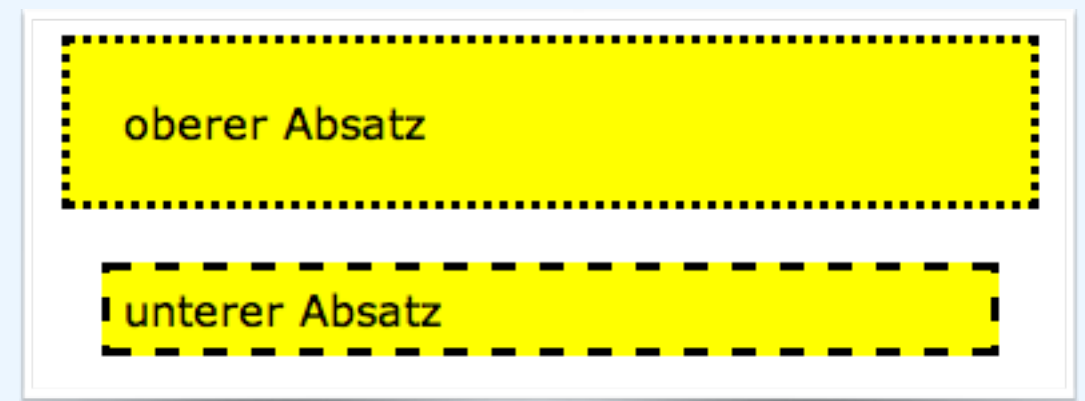
- `padding` (Innenabstand allgemein)
numerischer Wert
- `margin` (Außenabstand allgemein)
numerischer Wert
- zusätzlich existieren noch Sonderformen mit `-top`,
`-bottom`, `-left` und `-right`
 - z.B.: `padding-right` um nur den rechten Innenabstand zu setzen

Rahmen

- `border-width` (Rahmendicke)
numerischer Wert
- `border-color` (Rahmenfarbe)
Farbangabe
- `border-style` (Rahmentyp)
none (nicht sichtbarer Rahmen)
dotted (gepunktet)
dashed (gestrichelt)
solid (durchgezogen)

Boxmodell

```
<html>
  <head>
    <style type="text/css">
      .info {
        background-color : yellow;
        padding          : 20px;
        border-style      : dotted;
        border-width      : 3px;
        margin            : 5px;
      }
      .wichtig {
        background-color : yellow;
        padding          : 5px;
        border-style      : dashed;
        border-width      : 3px;
        margin            : 20px;
      }
    </style>
  </head>
  <body>
    <p class="info">oberer Absatz</p>
    <p class="wichtig">unterer Absatz</p>
  </body>
</html>
```

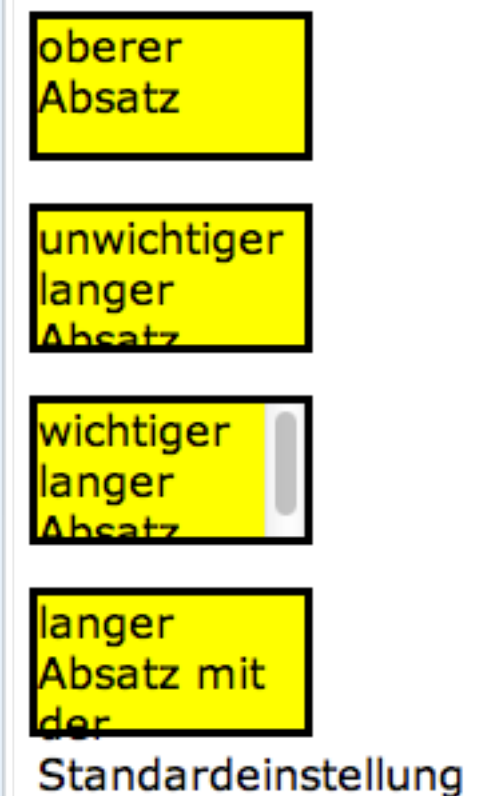


Größe des Inhaltsbereichs

- `width` (Breite)
numerischer Wert
- `height` (Höhe)
numerischer Wert
- `overflow` (wie soll übergroßer Inhalt dargestellt werden?)
 - `visible` (übergroßer Inhalt ragt aus dem Element heraus)
 - `hidden` (übergroßer Inhalt wird abgeschnitten)
 - `scroll` (im Element werden Scrollbalken dargestellt)
 - `auto` (Scrollbalken werden bei Bedarf dargestellt)

Größe des Inhaltsbereichs

```
<html>
  <head>
    <style type="text/css">
      .info {
        width           : 100px;
        height          : 50px;
        background-color : yellow;
        border-style     : solid;
      }
      .unwichtig { overflow : hidden; }
      .wichtig  { overflow : auto; }
    </style>
  </head>
  <body>
    <p class="info">oberer Absatz</p>
    <p class="info unwichtig" >
      unwichtiger langer Absatz
    </p>
    <p class="info wichtig" >
      wichtiger langer Absatz
    </p>
    <p class="info" >
      langer Absatz mit der
      Standardeinstellung
    </p>
  </body>
</html>
```



Textfluss der Elemente

Schwebende Elemente

Fluss der Elemente

```
<html>
  <body>
    <h1>Rezept</h1>
    
    <p>
      Beschreibung des Rezepts
    </p>
    <h2>Zubereitung</h2>
    <p>
      Absatz mit der Zubereitung
    </p>
  </body>
</html>
```

h1

img

p

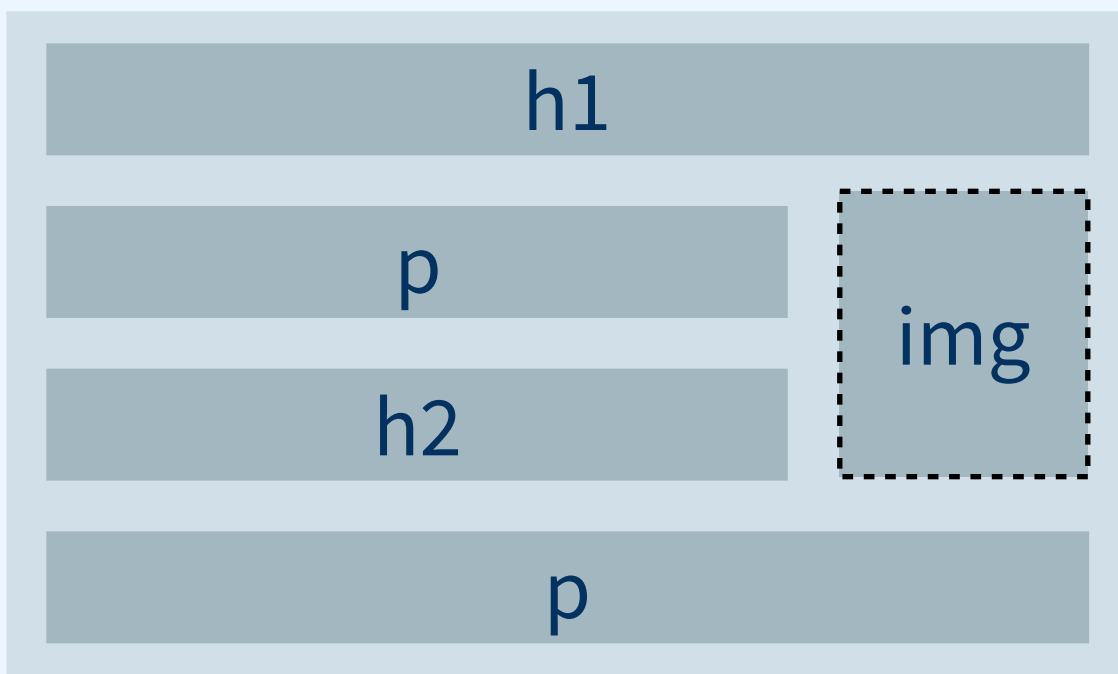
h2

p

- Blockelemente werden der Reihe nach dargestellt
- Blockelemente nehmen volle Breite ein
- Nach jedem Blockelement erfolgt ein Zeilenumbruch

Schwebende Elemente

```
<html>
  <body>
    <h1>Rezept</h1>
    
    <p>
      Beschreibung des Rezepts
    </p>
    <h2>Zubereitung</h2>
    <p>
      Absatz mit der Zubereitung
    </p>
  </body>
</html>
```



- float (schwebendes Element)
 - right (auf der rechten Seite)
 - left (auf der linken Seite)
- Nimmt Element aus dem Fluss heraus
- Fügt es unter das vorhergehende Element
- Bedingung: Element muss eine feste Breite haben

Schwebende Elemente

Ohne schwebende
Elemente

Rezept



Beschreibung des Rezepts

Zubereitung

Absatz mit der Zubereitung

Mit Grafik als
schwebendes Element

Rezept

Beschreibung des Rezepts

Zubereitung

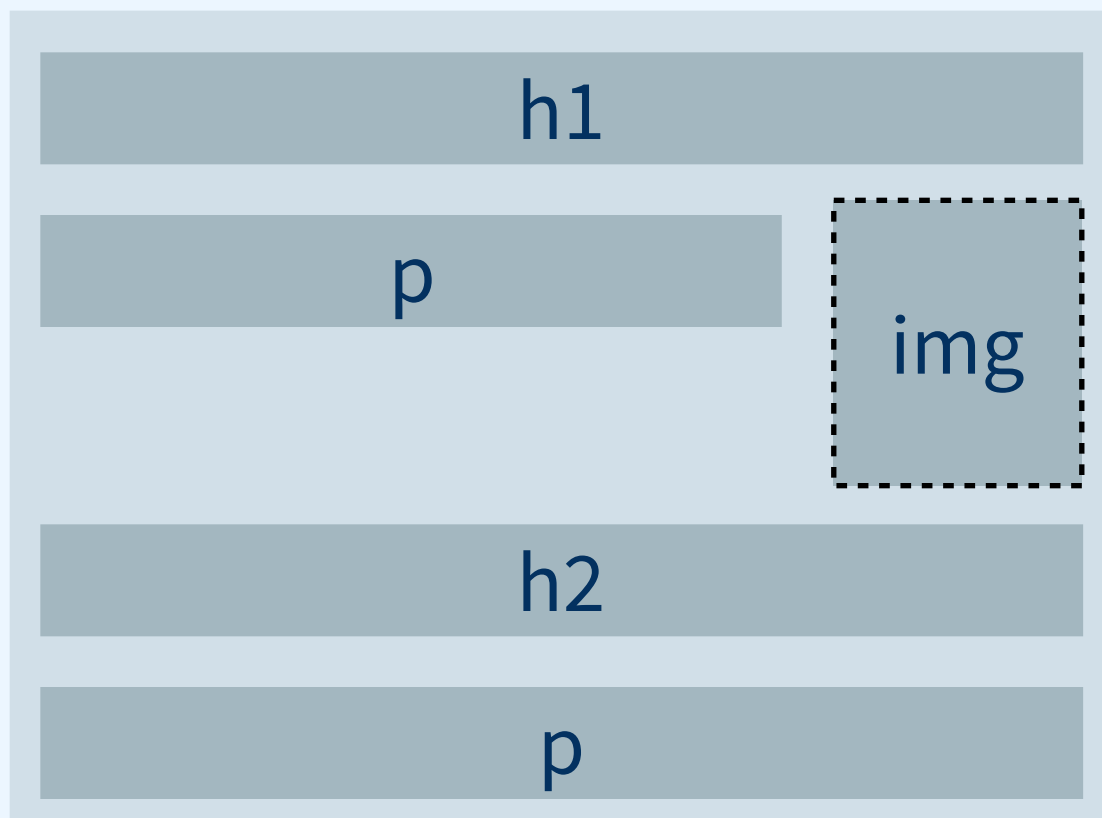
Absatz mit der Zubereitung



Schwebende Elemente

```
<html>
  <body>
    <h1>Rezept</h1>
    
    <p>Beschreibung des Rezepts</p>
    <h2 style="clear:right;"
        Zubereitung
    </h2>
    <p>Absatz mit der Zubereitung</p>
  </body>
</html>
```

- `clear` (schweben beenden)
`right` (auf der rechten Seite)
`left` (auf der linken Seite)
`both` (auf beiden Seiten)
- Das Element umfließt nicht mehr das schwebende Element
- Element befindet sich unter dem schwebenden Element
- Leerraum wird eingefügt



Schwebende Elemente

Ohne clear

Mit clear bei der
zweiten Überschrift

Rezept

Beschreibung des Rezepts

Zubereitung

Absatz mit der Zubereitung



Rezept

Beschreibung des Rezepts

Zubereitung

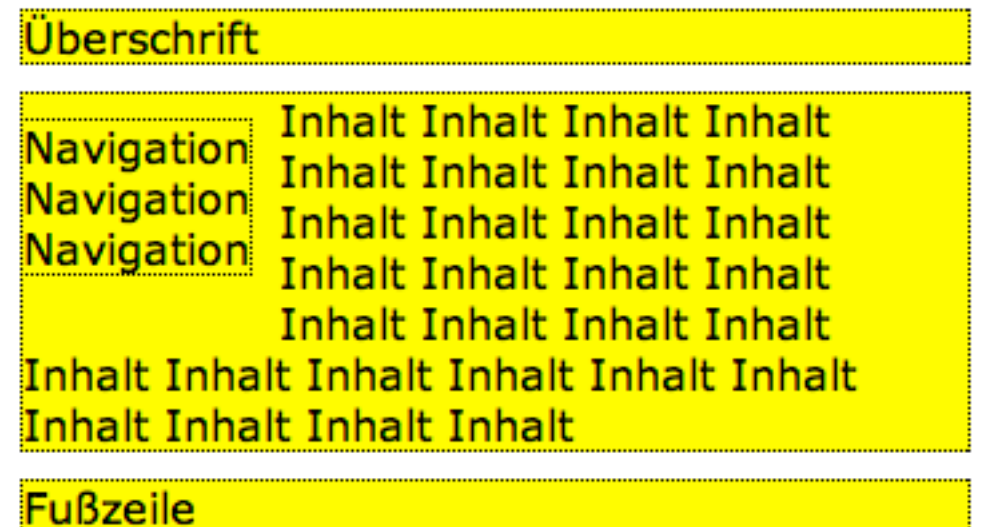
Absatz mit der Zubereitung



Zwei Spalten Layout

mit schwebenden Elementen

```
<html>
  <head>
    <style type="text/css">
      div {
        background-color : yellow;
        border-style      : dotted;
        border-width      : 1px;
        margin            : 10px;
      }
      #navigation {
        float            : left;
      }
      #fusszeile {
        clear            : left;
      }
    </style>
  </head>
  <body>
    <div>Überschrift</div>
    <div id="navigation">Navigation...</div>
    <div>Inhalt...</div>
    <div id="fusszeile">Fußzeile</div>
  </body>
</html>
```



Veränderung der Position

Positionierte Elemente

Einleitung

- Der Fluss der Elemente (und auch schwebende Elemente) lassen nur wenig Spielraum für genaue Layouts
 - ▶ HTML und CSS sollte nicht pixelgenau sein, damit der Browser einen Entscheidungsspielraum für die Darstellung hat
 - ▶ HTML ist nicht PDF oder JPG
- Dem stehen aber die Wünsche der Designer/innen entgegen
 - ▶ der Kompromiss sind positionierte Elemente
 - ▶ HTML-Dokument bleibt unverändert, CSS Eigenschaften positionieren Elemente beliebig

Positionierung

- `position` (Positionsart)
 - `static` (normaler Elementfluss, keine Verschiebung möglich)
 - `relative` (relative Positionierung, Verschiebung)
 - `absolute` (absolute Positionierung, gemessen am Rand des Anzeigebereichs)
- Die Eigenschaft bewirkt direkt keine sichtbare Änderung
 - Elemente erscheinen weiterhin an der ursprünglichen Position
 - sie legt den Modus für die Positionierung fest
 - erst weitere Eigenschaften verschieben die Elemente

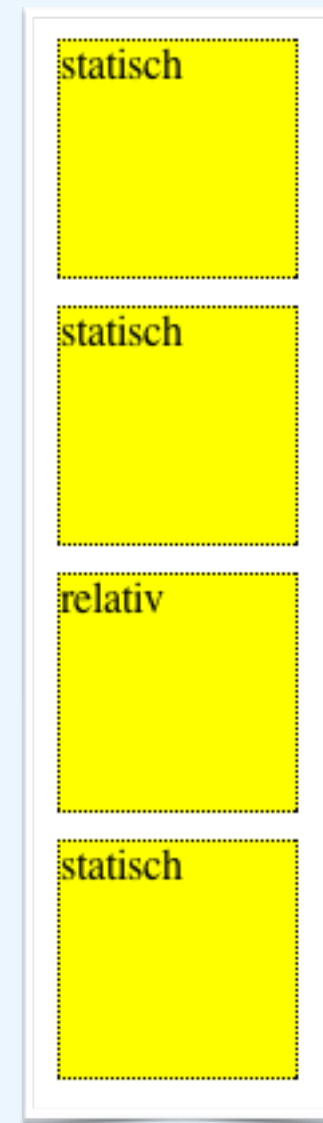
Positionierung

- `top` (Startposition von oben)
numerischer Wert
- `left` (Startposition von links)
numerischer Wert
- `bottom` (Startposition von unten)
numerischer Wert
- `right` (Startposition von rechts)
numerischer Wert
- Positive Werte → Verschiebung nach innen
Negative Werte → Verschiebung nach außen

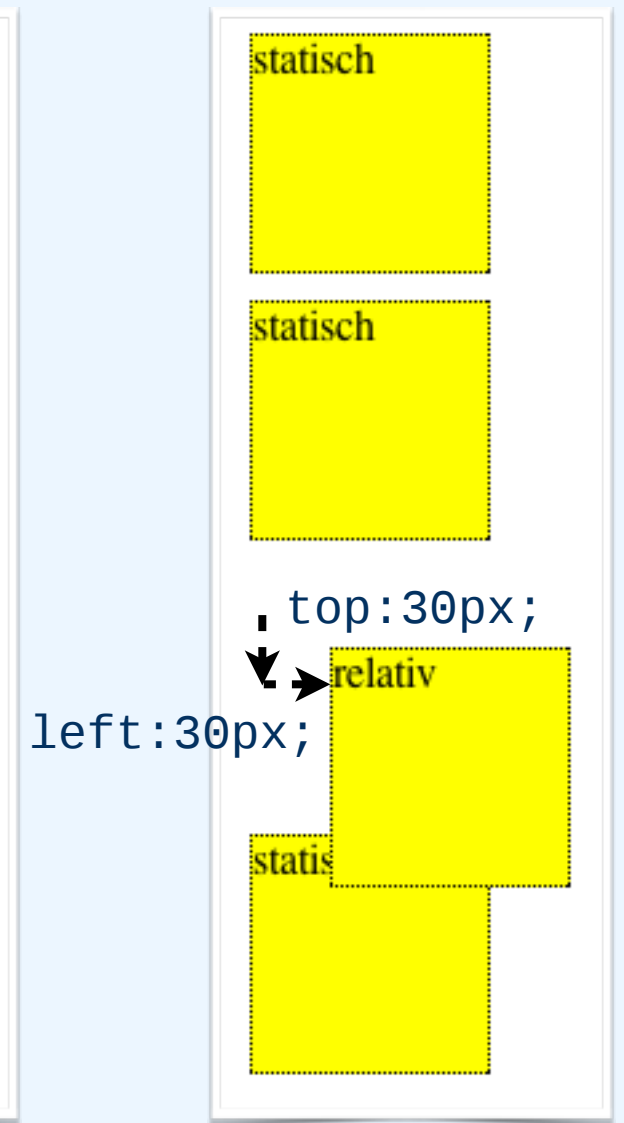
Position relative

```
<html>
  <head>
    <style type="text/css">
      div {
        background-color : yellow;
        border-style      : dotted;
        border-width      : 1px;
        margin            : 10px;
        width             : 88px;
        height            : 88px;
      }
      #relativ {
        position : relative;
        top      : 30px;
        left     : 30px;
      }
    </style>
  </head>
  <body>
    <div>statisch</div>
    <div>statisch</div>
    <div id="relativ">relativ</div>
    <div>statisch</div>
  </body>
</html>
```

ohne position



mit position



Position relative

- Nimmt Element nicht aus dem Textfluss heraus
 - ▶ Leerraum bleibt erhalten
- Verschiebt Element im Vergleich zur ursprünglichen Position
- Elemente können sich aufgrund der Verschiebung auch überlappen
- Eigentlich nur geeignet für kleine Verschiebungen
 - ▶ der Leerraum ist sonst immer ein Problem

Position absolute

- Im Modus `position: absolute` wird das Element absolut zu dem Anzeigebereich positioniert
 - ▶ z.B. bewirkt `top: 0px`, dass das Element oben auf der Seite erscheint
 - ▶ z.B. bewirkt `bottom: 0px`, dass das Element unten auf der Seite erscheint
- Das Element wird aus dem Fluss der Elemente herausgenommen
 - ▶ Es bleibt kein Leerraum an der ursprünglichen Stelle
- Es gibt Ausnahmen → `selfhtml.de`

Position absolute

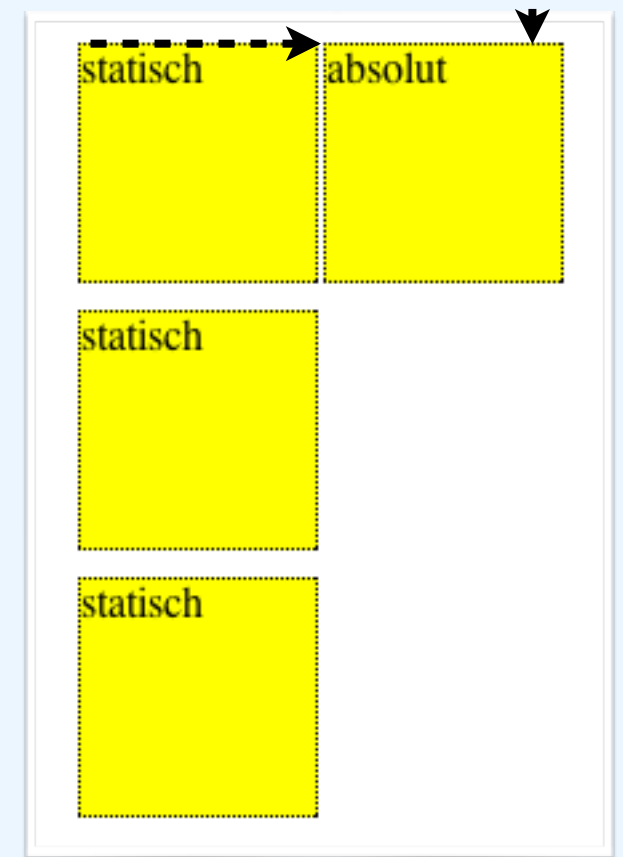
```
<html>
  <head>
    <style type="text/css">
      div {
        background-color : yellow;
        border-style      : dotted;
        border-width      : 1px;
        margin            : 10px;
        width             : 88px;
        height            : 88px;
      }
      #absolut {
        position : absolute;
        top      : 0px;
        left     : 100px;
      }
    </style>
  </head>
  <body>
    <div>statisch</div>
    <div>statisch</div>
    <div id="absolut">absolut</div>
    <div>statisch</div>
  </body>
</html>
```

ohne position



mit position

left:100px; top:0px;



Zwei Spalten Layout

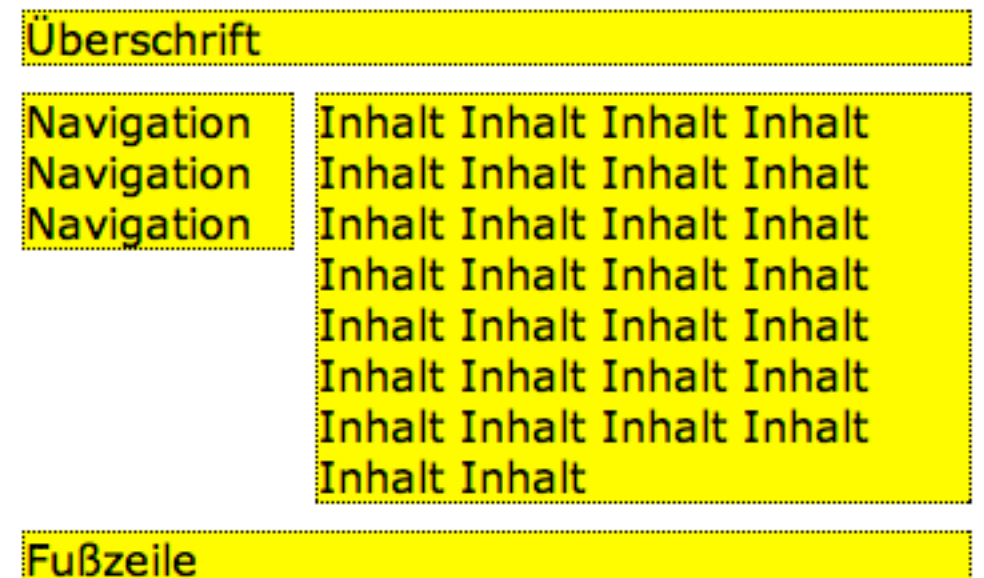
mit absolut positionierten Elementen

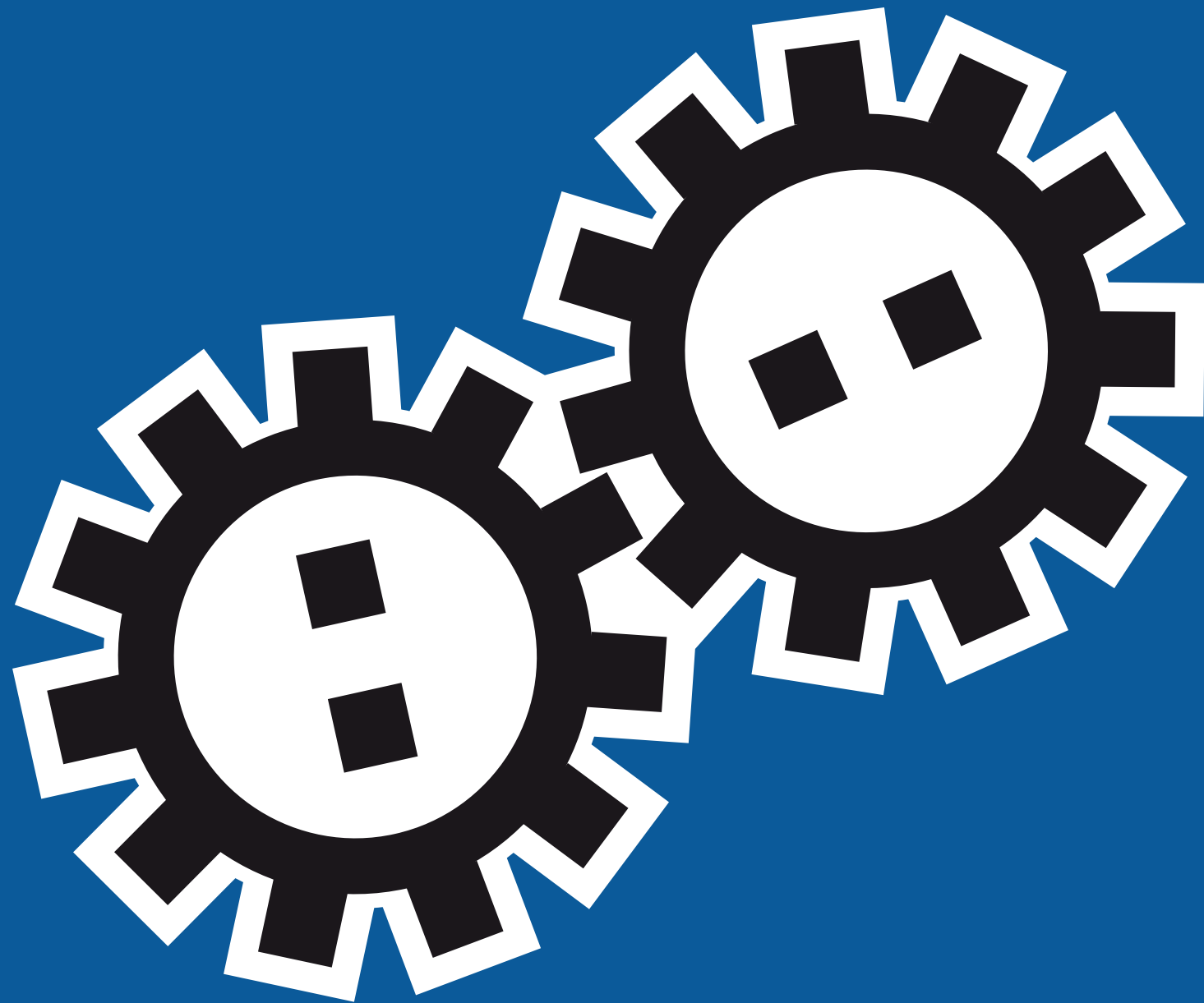
```
<html>
  <head>
    <style type="text/css">
      div {
        background-color : yellow;
        border-style      : dotted;
        border-width      : 1px;
        margin            : 10px;
      }
      #navigation {
        position          : absolute;
        width             : 100px;
      }
      #inhalt {
        margin-left       : 120px;
      }
    </style>
  </head>
  <body>
    <div>Überschrift</div>
    <div id="navigation">Navigation...</div>
    <div id="inhalt">Inhalt...</div>
    <div>Fußzeile</div>
  </body>
</html>
```

Damit Elemente sich nicht überlappen, können entsprechend breite margins bei den folgenden Elemente verwendet werden:

Navigation → width:100px;

Inhalt → margin-left:120px;

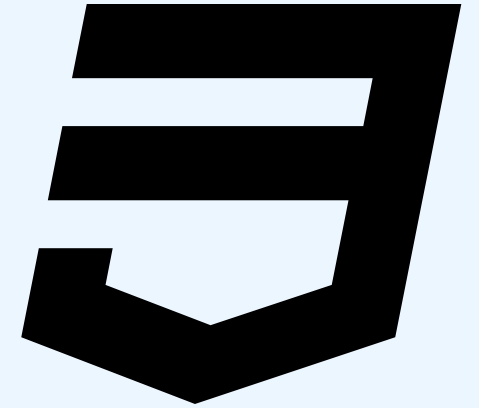




TECHNIKEN

CSS3

- Das obere Logo ist das offizielle, das untere Logo ist weit verbreitet
- Die Syntax bleibt unverändert:
Selektor {
 Eigenschaft: Wert;
}
- Einige neue Selektoren, und viele neue Eigenschaften



© w3c



© mohldesign.hr

Strukturelle Pseudoklassen in CSS3

Selektoren

CSS3 Selektoren

- Strukturelle Pseudoklassen → wählen Elemente basierend auf Ihrer Position im HTML-Dokument
- Einfachstes Beispiel :first-child
wählt nur das Element, das das erste Kindelement des Elternelements ist

```
tr:first-child {  
    background-color: red;  
}
```

Erste Zeile in der Tabelle → rote Hintergrundfarbe

- analog :last-child
- Jedes zweite Element :nth-child(even)

```
tr:nth-child(even) {  
    background-color: green;  
}
```

jede gerade Zeile in der Tabelle → grüne Hintergrundfarbe

CSS3 Selektoren

```
<html>
<head>
<style type="text/css">
  table {
    background-color: #def;
  }
  tr:first-child {
    background-color: #fdd;
  }
  tr:nth-child(even) {
    background-color: #dfe;
  }
</style>
</head>
<body>
  <table border="1">
    <tr>
      <th colspan="3">Tabelle</th>
    </tr>
    <tr>
      <td>1/1</td><td>1/2</td><td>1/3</td>
    </tr>
    <tr>
      <td>2/1</td><td>2/2</td><td>2/3</td>
    </tr>
    <tr>
      <td>3/1</td><td>3/2</td><td>3/3</td>
    </tr>
  </table>
</body>
</html>
```

nur Regel table {...}

Tabelle		
1/1	1/2	1/3
2/1	2/2	2/3
3/1	3/2	3/3

mit allen Regeln

Tabelle		
1/1	1/2	1/3
2/1	2/2	2/3
3/1	3/2	3/3

IE	Firefox	Chrome	Safari	Opera	iOS Safari +	Opera Mini +	Android Browser +	Chrome for Android
		31						
		33						
8		35	5.1				4.1	
9	31	36	7		7.1		4.3	
10	32	37	7.1		8		4.4	
11	33	38	8	25	8.1	8	4.4.4	38
	34	39		26			3.7	
	35	40		27				
	36	41						

CSS3 für Block-Elemente

Rahmen, Schatten, etc

abgerundete Rahmen

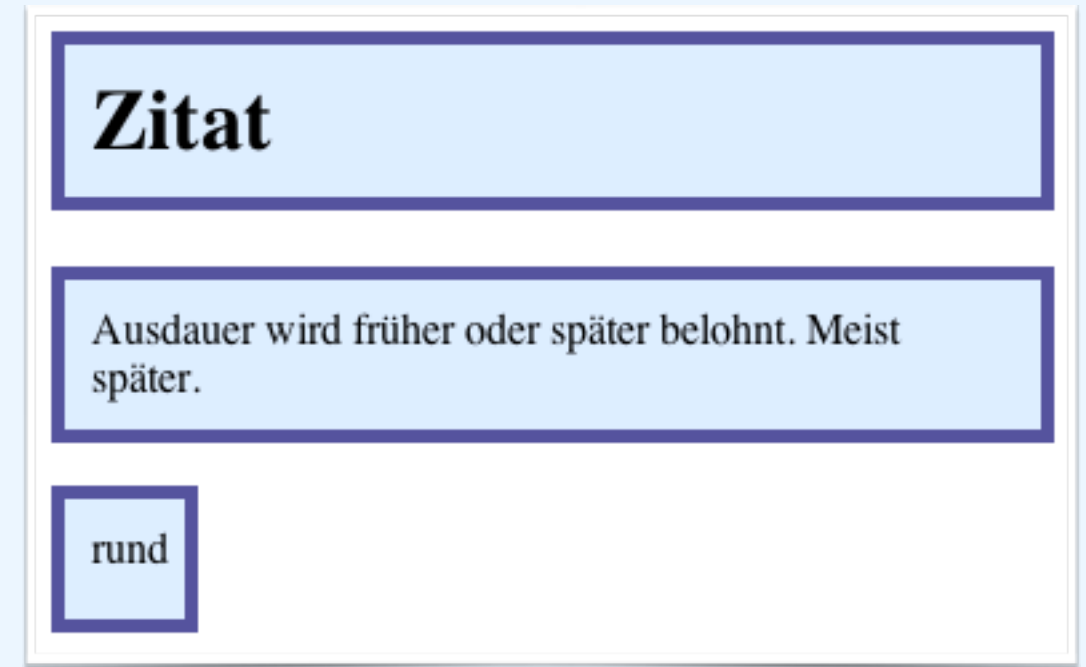
- Der Rahmen von Block-Elementen ist immer rechteckig
- Mit der Eigenschaft border-radius können die Kanten abgerundet werden
 - ▶ der Wert gibt den Kreistradius für die Ecken an
- Die Ecken können auch einzeln gesetzt werden, z.B.:
`border-top-left-radius: 10px;`
- Mit passenden Werten können auch Kreise erzeugt werden

IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Chrome for Android
		31						
		33						
8		35	5.1				4.1	
9	31	36	7		7.1		4.3	
10	32	37	7.1		8		4.4	
11	33	38	8	25	8.1	8	4.4.4	38
	34	39		26			37	
	35	40		27				
	36	41						

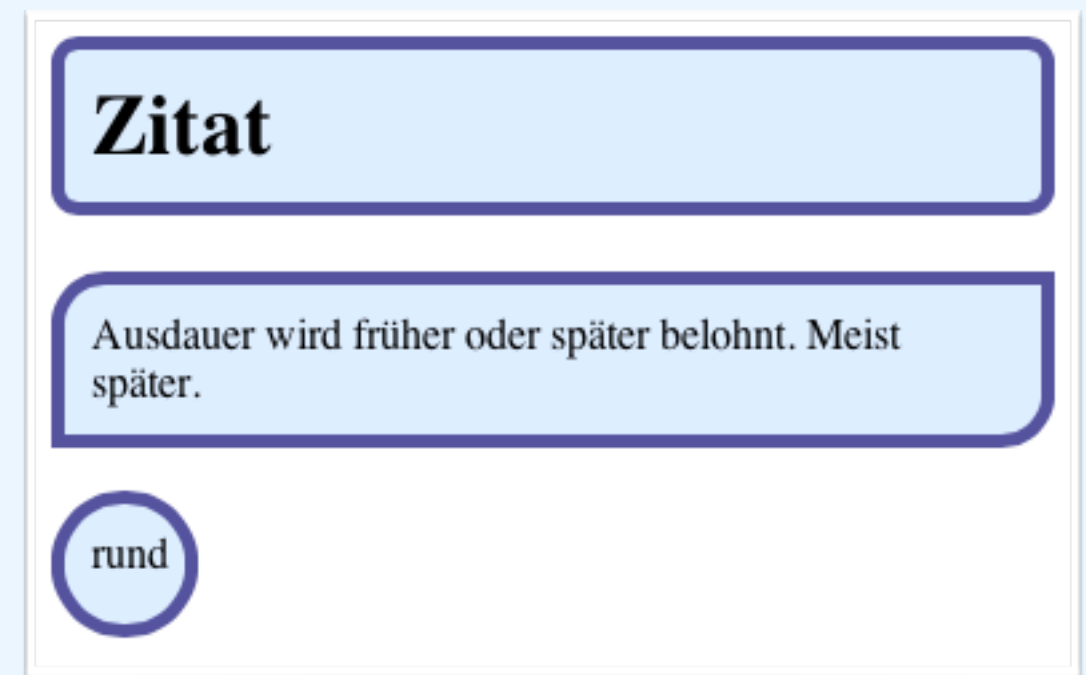
abgerundete Rahmen

```
<html>
<head>
<style type="text/css">
  h1,p,div {
    background-color : #def;
    padding          : 10px;
    border-color      : #559;
    border-width      : 5px;
    border-style      : solid;
  }
  h1 {
    border-radius      : 10px;
  }
  p {
    border-top-left-radius : 20px;
    border-bottom-right-radius : 20px;
  }
  div {
    width: 25px;
    height:25px;
    border-radius:30px;
  }
</style>
</head>
<body>
  <h1>Zitat</h1>
  <p>Ausdauer wird früher oder später
belohnt. Meist später.</p>
  <div>rund</div>
</body>
</html>
```

ohne border-radius



mit border-radius



Schatten

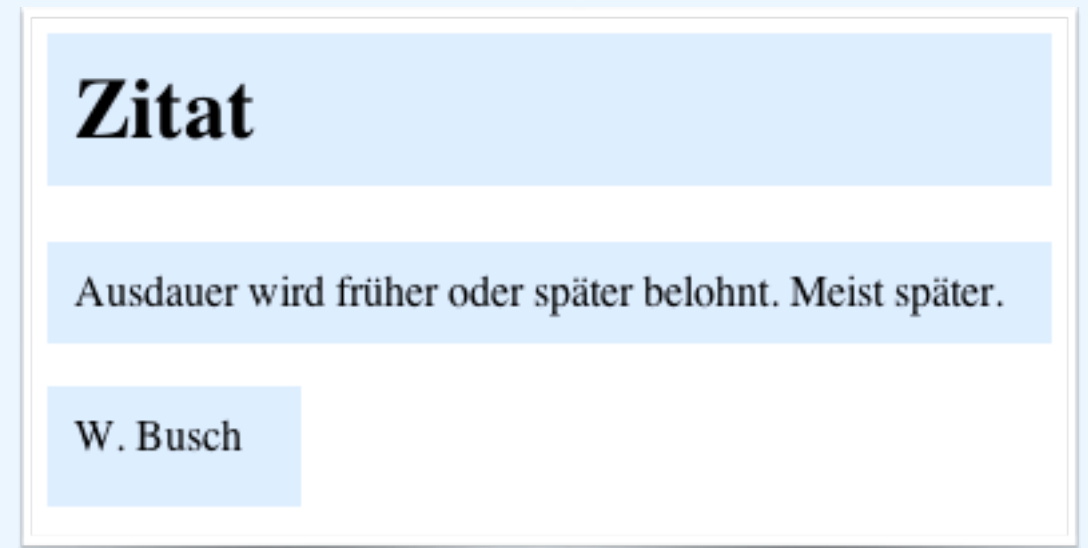
- Mit CSS3 können Block-Elemente mit einem Schatten versehen werden
- Die Eigenschaft `box-shadow` kann folgende Zuweisung erhalten:
`<offset-x> <offset-y> <blur-radius>? <color>?`
 - ▶ horizontale Verschiebung des Schattens zum Block-Element
 - ▶ vertikale Verschiebung des Schattens zum Block-Element
 - ▶ Größe des Schatten
 - ▶ Farbe des Schatten

IE	Firefox	Chrome	Safari	Opera	iOS Safari*	Opera Mini*	Android Browser*	Chrome for Android
		31						
		33						
8		35	5.1				4.1	
9	31	36	7		7.1		4.3	
10	32	37	7.1		8		4.4	
11	33	38	8	25	8.1	8	4.4.4	38
	34	39		26			37	
	35	40		27				
	36	41						

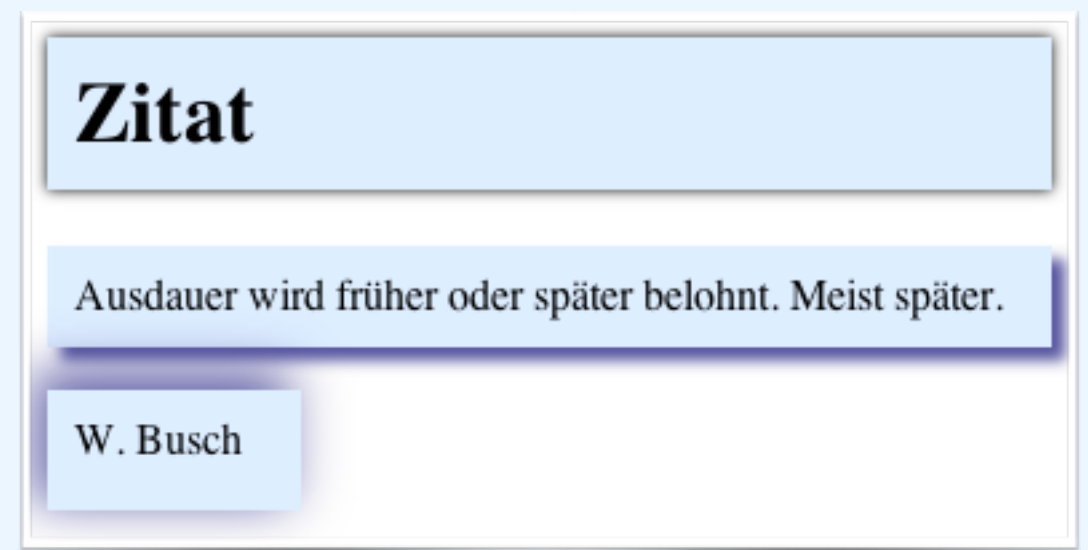
Schatten

```
<html>
<head>
<style type="text/css">
  h1,p,div {
    background-color : #def;
    padding          : 10px;
  }
  h1 {
    box-shadow: 0 0 5px black;
  }
  p {
    box-shadow: 5px 5px 5px #559;
  }
  div {
    width: 75px;
    height:25px;
    box-shadow: -5px -5px 20px #559;
  }
</style>
</head>
<body>
  <h1>Zitat</h1>
  <p>Ausdauer wird früher oder später
belohnt. Meist später.</p>
  <div>W. Busch</div>
</body>
</html>
```

ohne box-shadow



mit box-shadow



Transformationen

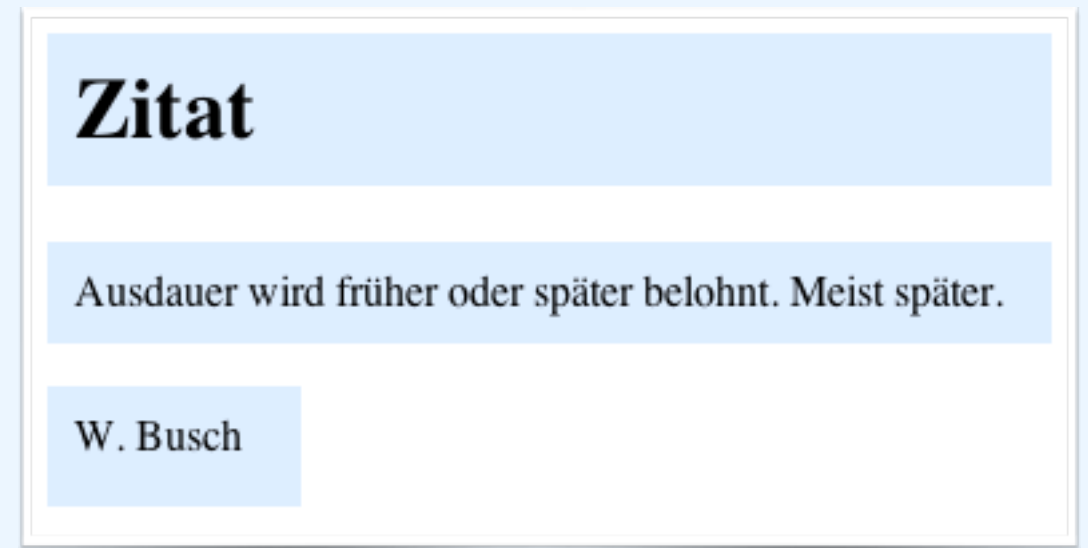
- CSS3 bietet die Möglichkeit, jedes Block-Element mit der Eigenschaft `transform` zu verändern
- `translateX(5px)` → horizontale Verschiebung
- `translateY(-5px)` → vertikale Verschiebung
- `scaleX(2)` → horizontale Skalierung
- `scaleY(2)` → vertikale Skalierung
- `rotate(45deg)` → Rotation
- Achtung: herstellerspezifische Prefixe notwendig
 - `-ms-transform:scaleX(2);`
 - `-moz-transform:scaleX(2);`
 - `-o-transform:scaleX(2);`

IE	Firefox	Chrome	Safari	Opera	iOS Safari*	Opera Mini*	Android Browser*	Chrome for Android
		31						
		33						
8		35	5.1				4.1	
9	31	36	7		7.1		4.3	
10	32	37	7.1		8		4.4	
11	33	38	8	25	8.1	8	4.4.4	38
	34	39		26			37	
	35	40		27				
	36	41						

Transformationen

```
<html>
<head>
<style type="text/css">
  h1,p,div {
    background-color : #def;
    padding          : 10px;
  }
  h1 {
    -webkit-transform: translateX(50px);
  }
  p {
    -webkit-transform: scaleX(0.5);
  }
  div {
    width: 75px;
    height:25px;
    -webkit-transform: rotate(45deg);
  }
</style>
</head>
<body>
  <h1>Zitat</h1>
  <p>Ausdauer wird früher oder später
belohnt. Meist später.</p>
  <div>W. Busch</div>
</body>
</html>
```

ohne transform



mit transform



Gradienten

- Wir können Block-Elementen eine Hintergrundfarbe zuweisen
 - ▶ background-color
- Oder in CSS3 einen Farbverlauf als Hintergrund definieren
- Es gibt zwei Arten:
 - ▶ linearer Farbverlauf
 - ▶ radialer Farbverlauf

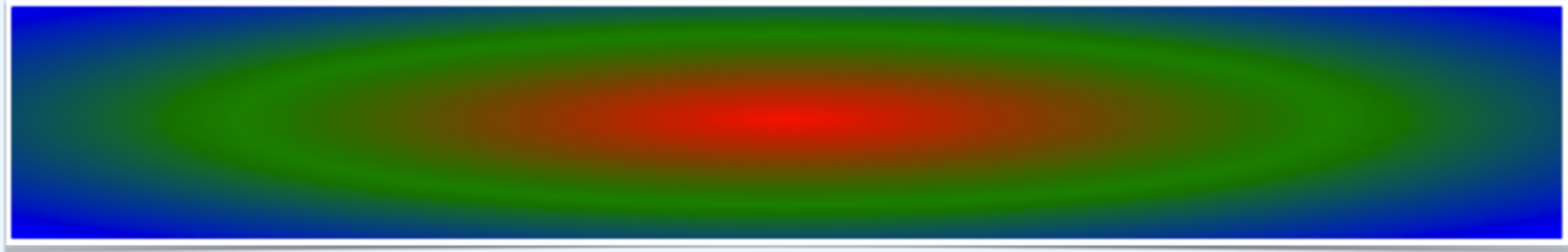
IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Chrome for Android
		31						
		33						
8		35	5.1				4.1	
9	31	36	7		7.1		4.3	
10	32	37	7.1		8		4.4	
11	33	38	8	25	8.1	8	4.4.4	38
	34	39		26			3.7	
	35	40		27				
	36	41						

Linearer Farbverlauf



- Linearer Farbverlauf über verschiedene Farben hinweg
 - ▶ mindestens zwei Farben
- `background: linear-gradient(<winkel>, <farbe1>, ..., <farbeN>);`
- Winkel bestimmt die Richtung des Verlaufs
 - ▶ 0 Grad → von links nach rechts
- Beispiel oben:
`background: linear-gradient(0deg, red, green, blue);`

Radialer Farbverlauf

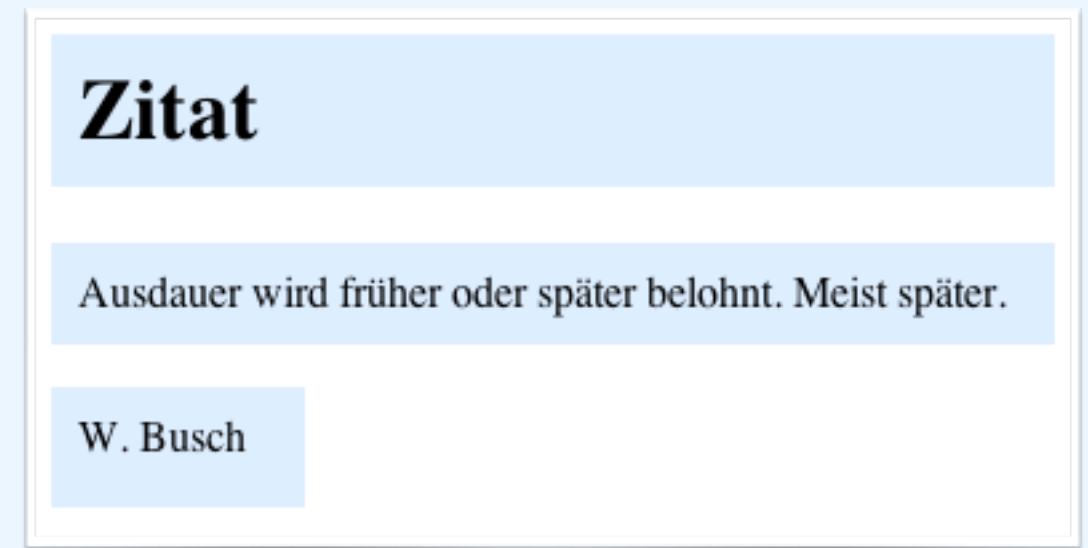


- Kreisförmiger Farbverlauf über mehrere Farben hinweg
- `background: radial-gradient(<farbe1>, ..., <farbeN>);`
- Beispiel oben:
`background: radial-gradient(red, green, blue);`
- Viele optionale Konfigurationsmöglichkeiten
→ <http://www.css3files.com/gradient/>

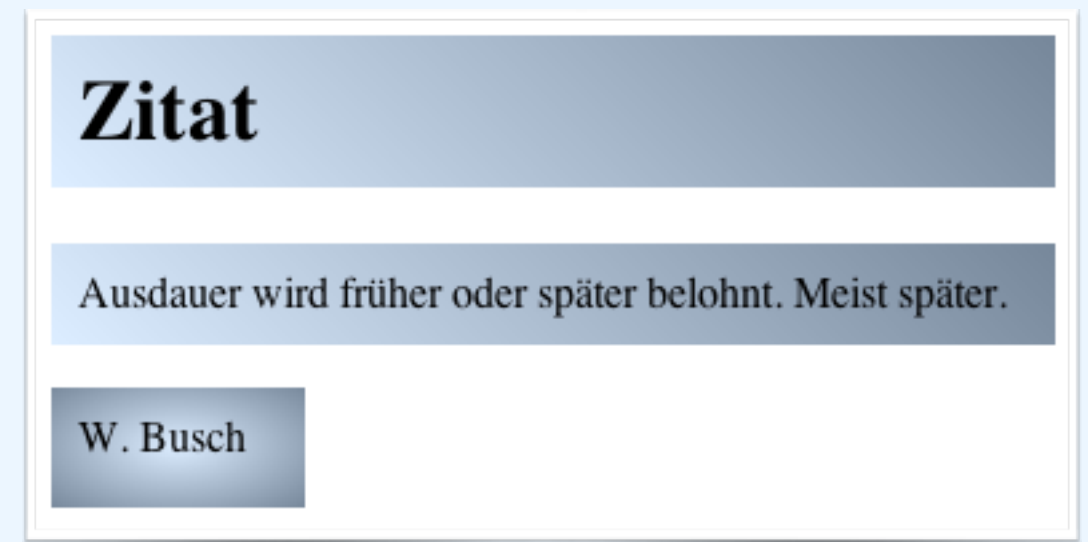
Gradienten

```
<html>
<head>
<style type="text/css">
  h1,p {
    padding : 10px;
    background:
      -webkit-linear-gradient(45deg, #def,#789);
  }
  div {
    width      : 75px;
    height     : 25px;
    padding    : 10px;
    background:
      -webkit-radial-gradient(#def,#789);
  }
</style>
</head>
<body>
  <h1>Zitat</h1>
  <p>Ausdauer wird früher oder später belohnt.
Meist später.</p>
  <div>W. Busch</div>
</body>
</html>
```

ohne Gradienten



mit Gradienten



Kombiniert

```
<html>
<head>
  <style type="text/css">
    p {
      width           : 200px;
      padding         : 50px;
      background:
        -webkit-radial-gradient(#def,#bcd);
    }
    div {
      width           : 70px;
      padding         : 5px;
      border-radius   : 10px;
      box-shadow      : 0 0 10px black;
      -webkit-transform: translateX(-10px)
        translateY(60px) rotate(-45deg);
      background:
        -webkit-linear-gradient(45deg, #afb,#dfe);
    }
  </style>
</head>
<body>
  <div>kostenlos!</div>
  <p>
    Ein tolles Angebot!<br/>
    Greifen Sie zu!
  </p>
</body>
</html>
```

ohne CSS3 Eigenschaften

kostenlos!

Ein tolles Angebot!
Greifen Sie zu!

mit CSS3 Eigenschaften

kostenlos!

Ein tolles Angebot!
Greifen Sie zu!

Exkurs: eigene Schriftarten im Web

Web Fonts

Web Fonts

- Schriftarten werden (in der Theorie) schon lange unterstützt
 - ▶ in HTML: `Andere Schriftart`
 - ▶ in CSS: `p { font-family: "Verdana"; }`
- Problem: Schrift muss im Client (Browser) bereits installiert sein
- Microsoft Projekt - Core fonts for the Web
Andale Mono, Arial, Arial Black, Comic Sans MS, Courier New, Georgia, Impact, Times New Roman, Trebuchet MS, Verdana, Webdings
 - ▶ mehr ist nicht verlässlich verfügbar
- Alternativen mit Flash, Grafiken und SVG sind nicht gut integriert

Web Fonts

- WOFF (Web Open File Format) seit 2010 vom W3C propagiert
 - Container für Schriftarten (TrueType, Open Font, OpenType)
- WOFF-Schriftarten können auf dem Server abgelegt werden
 - URL wird in der CSS angegeben
- CSS3 "Selektor" @font-face mit Eigenschaften:
 - font-family → Name der Schrift
 - src → URL der WOFF-Datei
- Nach der Einbindung kann die Schrift wie eine lokal installierte verwendet werden

IE	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
		31						
		33						
8		35	5.1				4.1	
9	31	36	7		7.1		4.3	
10	32	37	7.1		8		4.4	
11	33	38	8	25	8.1	8	4.4.4	38
	34	39		26			3.7	
	35	40		27				
	36	41						

Web Fonts

```
<html>
<head>
<style type="text/css">
  @font-face {
    font-family : "DIN Schrift";
    src: url('dinfont.woff') format('woff');
  }
  * {
    font-family: "DIN Schrift";
  }
</style>
</head>
<body>
  <h1>Zitat</h1>
  <p>Ausdauer wird früher oder später belohnt.
Meist später.</p>
</body>
</html>
```

ohne @font-face

Zitat

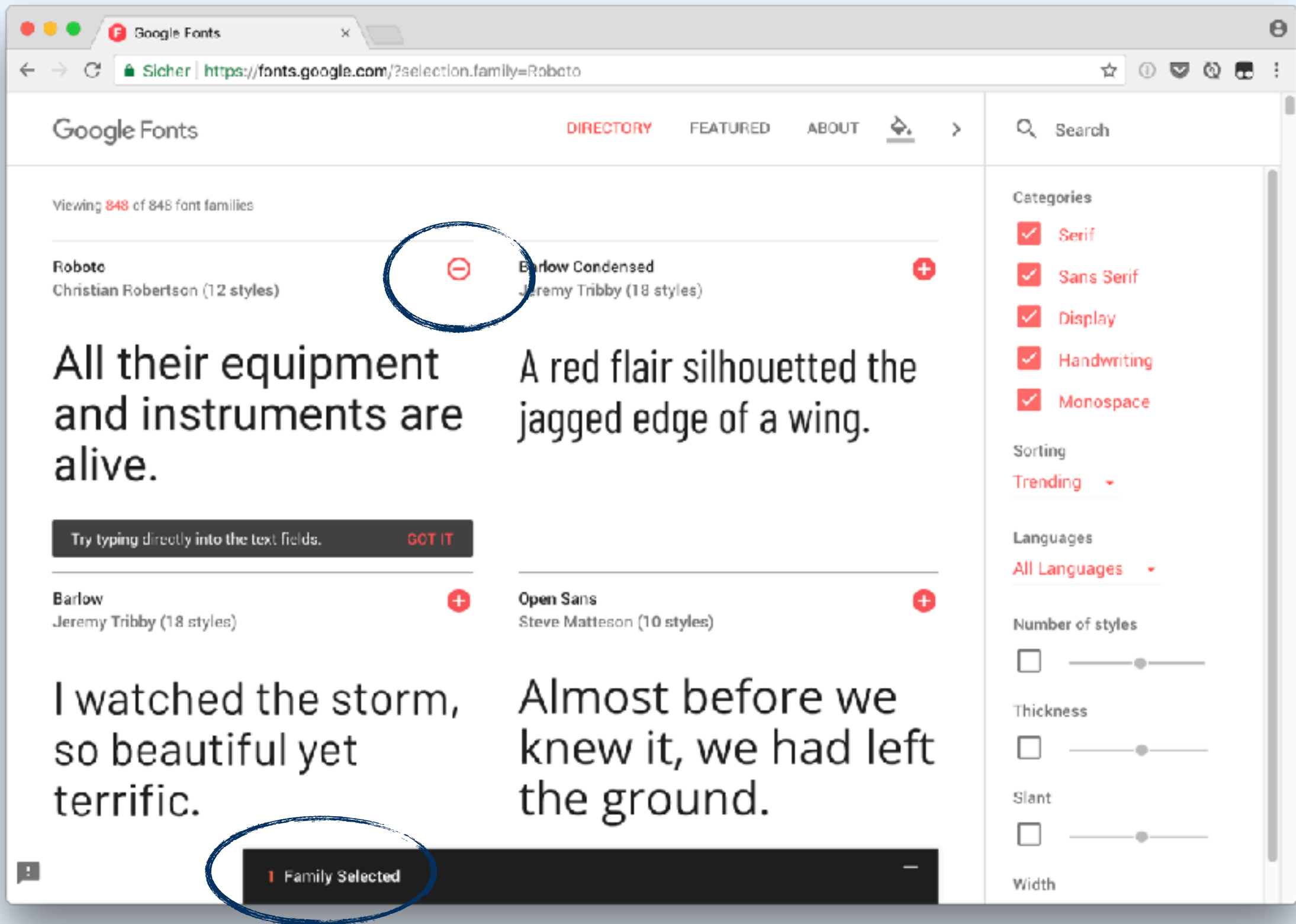
Ausdauer wird früher oder später belohnt. Meist später.

mit @font-face

Zitat

Ausdauer wird früher oder später belohnt. Meist später.

Google Web Fonts



Elemente bewegen ohne JavaScript

Animationen

Animationen

- Auf Mobilgeräten wird (häufig) kein Flash unterstützt
 - ▶ gerade dieses wird aber häufig für Animationen verwendet
- Lösung soll HTML5 sein (gemeint ist aber CSS3)
- Grundsätzliche Idee:
 - ▶ verschiedene Zustände von HTML-Elementen mit CSS definieren
 - ▶ Übergang von einem zum anderen Zustand animieren
- Animation kann mit zwei Techniken realisiert werden:
 - ▶ JavaScript (für komplexe Animationen)
 - ▶ mit der CSS3-Eigenschaft transition

Transition

- Transitionen definieren, wie der Übergang zwischen zwei Werten berechnet werden soll
- Wichtig: CSS3 kann nur Eigenschaften animieren, deren Werte sich überblenden lassen:
 - ▶ Farben, Abstände, Breiten → Zahlenwerte
 - ▶ aber nicht font-family, position, text-align
→ Eigenschaften mit definierten Werten (Zeichenketten)
- Zwei Eigenschaften definieren die Animation:
 - ▶ `transition-property` → Angabe der Eigenschaft(en), die animiert werden soll (Standard ist all)
 - ▶ `transition-duration` → Dauer der Animation

Transition

- Einfaches Beispiel ohne Animation:
 - ▶ Hintergrundfarbe ändert sich mittels :hover Pseudoklasse

```
<html>
<head>
<style type="text/css">
  h1 {
    background-color: #55f;
  }
  h1:hover {
    background-color: #0f0;
  }
</style>
</head>
<body>
  <h1>Zitat</h1>
  <p>Ausdauer wird früher oder später belohnt.
Meist später.</p>
</body>
</html>
```

Überschrift ohne Mausberührung

Zitat

Ausdauer wird früher oder später belohnt. Meist später.

Überschrift mit Mausberührung

Zitat

Ausdauer wird früher oder später belohnt. Meist später.

externer Link

Transition

- Einfaches Beispiel mit Animation:
 - ▶ Transition-Eigenschaften eingefügt

```
<html>
<head>
<style type="text/css">
  h1 {
    background-color: #55f;
    -webkit-transition-property: background-color;
    -webkit-transition-duration: 2s;
  }
  h1:hover {
    background-color: #0f0;
  }
</style>
</head>
<body>
  <h1>Zitat</h1>
  <p>Ausdauer wird früher oder später belohnt.
Meist später.</p>
</body>
</html>
```

Überschrift ohne Mausberührung

Zitat

Ausdauer wird früher oder später belohnt. Meist später.

Überschrift mit Mausberührung

Zitat

Ausdauer wird früher oder später belohnt. Meist später.

externer Link

Transition

- Transitionen werden üblicherweise mit der :hover Pseudoklasse kombiniert
 - ▶ es wird nur etwas animiert, wenn die Maus ein Element berührt
- Der Browser sorgt für ein korrektes Aufräumen der Animation
 - ▶ wenn der Mauszeiger ein Element verlässt, wird der Ursprungszustand wieder hergestellt

IE	Firefox	Chrome	Safari	Opera	iOS Safari*	Opera Mini*	Android Browser*	Chrome for Android
		31						
		33						
8		35	5.1				4.1	
9	31	36	7		7.1		4.3	
10	32	37	7.1		8		4.4	
11	33	38	8	25	8.1	8	4.4.4	38
	34	39		25			3.7	
	35	40		27				
	36	41						

Transition-Transformation

- Animation über mehrere Eigenschaften:
Schatten und Transformation

```
<html>
<head>
<style type="text/css">
  ul {
    list-style: none;
    font-size: 30px;
  }
  li {
    -webkit-transition-property: all;
    -webkit-transition-duration: 1s;
  }
  li:hover {
    text-shadow: 5px 5px 5px #777;
    -webkit-transform: translateX(10px);
  }
</style>
</head>
<body>
  <ul>
    <li>Hauptseite</li>
    <li>Nachrichten</li>
    <li>Hilfe</li>
  </ul>
</body>
</html>
```

Text ohne Mausberührung

Hauptseite
Nachrichten
Hilfe

Text mit Mausberührung

Hauptseite
Nachrichten
Hilfe

externer Link



ZUSAMMENFASSUNG

Zusammenfassung

- Layout
 - ▶ Boxmodell
 - ▶ Fluss der Elemente
 - ▶ absolute und relative Positionierung
- CSS3 Selektoren
- Rahmen, Schatten, Transformationen
- Web Fonts
- Animationen

DANKE