

# Webanwendungen

Vorlesung - Hochschule Mannheim

## ► Cascading Style Sheets

# Inhaltsverzeichnis

- ▶ [CSS Grundlagen](#)
- ▶ [CSS-Selektoren](#)
- ▶ [Maßeinheiten](#)
- ▶ [CSS-Attribute](#)
- ▶ [CSS-Layout](#)
- ▶ [Kaskadierung und Vererbung](#)

# CSS Grundlagen

# Herausforderungen beim Layout

- ▶ HTML zeichnet Text logisch aus
  - ▶ Fließtext mit dynamischem Umbruch
  - ▶ keine feste Platzierung von Elementen
  - ▶ Darstellung folgt der Reihenfolge der Tags in der HTML-Datei
  - ▶ keine Überdeckung von Objekten
  - ▶ Browser stellt Inhalt passend für das Gerät des Benutzers dar
- ▶ Ziel: Layout muss sich dynamisch an die Nutzung (PC, Tablet, Smartphone) anpassen können

# Beispiel: CSS Zen ohne CSS

## **css Zen Garden**

### **The Beauty of CSS Design**

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.

Download the sample [html file](#) and [css file](#)

### **The Road to Enlightenment**

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

### **So What is This About?**


There is clearly a need for CSS to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external .css file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. To date, most examples of neat tricks and hacks have been demonstrated by structurists and coders. Designers have yet to make their mark. This needs to change.

# Beispiel: CSS Zen mit CSS



# Beispiel: CSS Zen mit CSS




A DEMONSTRATION  
OF WHAT CAN BE ACCOMPLISHED  
VISUALLY THROUGH CSS-BASED  
DESIGN. SELECT ANY STYLE  
SHEET FROM THE SUPPLIES LIST  
TO LOAD IT INTO THIS PAGE.

## ZEN ARMY

### AN ONSLAUGHT OF CSS DESIGN

Laden Sie die Beispieldateien für **HTML** und **CSS** herunter



#### SUPPLIES:

- spring360  
von Rene Hornig
- Withering Beauty  
von William Duffy
- Tiny Blue  
von Timo Virtanen
- Retro Theater

#### ★ THE ROAD TO VICTORY

Wir blicken zurück auf den dunklen Weg vergangener Relikte wie  
browserspezifischen Tags, inkompatiblen DOMs und einer brüchigen CSS  
Unterstützung.

Heute müssen wir den Kopf von den alten Praktiken befreien. Die Erleuchtung im  
Web wurde zuletzt dank der unermüdlichen Anstrengungen von Leuten wie  
dem W3C, WaSP sowie den großen Browserherstellern erreicht.

Der css Zen Garden lädt Sie ein zu entspannen und über die wesentlichen Lektionen  
der Meister zu meditieren. Beginnen Sie klar zu sehen. Lernen Sie die (letzten Endes)  
Zeit gewinnenden Techniken in einer neuen und anregenden Weise anzuwenden.



# Beispiel: CSS Zen mit CSS

Wir blicken zurück auf den dunklen Weg vergangener Relikte wie browserspezifischen Tags, inkompatiblen DOMs und einer brüchigen CSS Unterstützung.

Heute müssen wir den Kopf von den alten Praktiken befreien. Die Erleuchtung im Web wurde nicht zuletzt dank der unermüdlichen Anstrengungen von Leuten wie dem W3C, W3P, sowie den großen Browserherstellern erreicht.

Der css Zen Garden lädt Sie ein zu entspannen und über die wesentlichen Lektionen der Meister zu meditieren. Beginnen Sie klar zu sehen. Lernen Sie die (letzten Endes) Zeit gewinnenden Techniken in einer neuen und anregenden Weise anzuwenden. Werden Sie Eins mit dem Web.

Laden Sie die Beispieldateien für HTML und CSS herunter

## BENEFITS

Warum teilnehmen? Wegen der Anerkennung, Inspiration und einer Quelle auf die wir alle verweisen können, wenn es um den Anwendungsfall von CSS basierendem Design geht. Dies ist dringend erforderlich, selbst heutzutage. Immer mehr bedeutende Seiten machen den Sprung, aber bislang zu wenige. Eines Tages wird diese Galerie eine historische Kuriosität sein, dieser Tag

## REQUIREMENTS

Wir möchten so viel CSS1 sehen wie möglich. CSS2 sollte auf die am weitesten unterstützten Elemente beschränkt werden. Beim css Zen Garden geht es um funktionales, praktisches CSS und nicht die allerheiligsten, letzten Tricks die von 2% der browsenden Öffentlichkeit gesehen werden können. Die einzig wirkliche Voraussetzung die wir haben ist, dass Ihr CSS validiert.

## SO WHAT IS THIS ABOUT?

Es ist klar erforderlich, dass CSS von Grafikdesignern erst genommen wird. Der Zen Garden strebt danach zu begeistern, zu inspirieren und zur Teilnahme zu ermutigen. Betrachten Sie sich zunächst einige der existierenden Designs aus der Liste. Wenn Sie eines davon anklicken wird es in diese Seite geladen. Der Code bleibt der selbe, das einzige, dass sich geändert hat ist die externe CSS Datei. Ja, wirklich.

CSS erlaubt vollständige und totale Steuerung über die Gestaltung eines Hypertext-Dokuments. Der einzige Weg es auf eine Weise zu illustrieren welche die Leute begeistert, besteht darin zu demonstrieren wie es wirklich sein kann, wenn man die Zügel denjenigen in die Hand gibt die in der Lage sind, Schönheit aus Struktur zu schaffen. Bis heute wurden die meisten Beispiele der geschickten Tricks und Hacks von Strukturisten und Programmierern vorgeführt. Designer müssen ihre Spuren erst noch hinterlassen. Das muss sich ändern.

## PARTICIPATION

Bitte nur Grafikdesigner. Sie modifizieren diese Seite, daher sind erhebliche Kenntnisse in CSS erforderlich, jedoch sind die Beispieldateien ausreichend kommentiert, damit sie auch von CSS-Anfängern als Ausgangspunkt verwendet werden können. Bitte schauen Sie in den Quellenführer für CSS nach fortgeschrittenen Tutorials und Tipps zur Arbeit mit CSS.

Sie können das Stylesheet nach Ihren Wünschen modifizieren, aber nicht das HTML. Das mag zunächst entmutigend klingen, wenn Sie vorher noch nie auf diese Weise gearbeitet haben, folgen Sie den aufgeführten Links um mehr darüber zu lernen und verwenden Sie

## ZEN GARDEN

The Beauty of CSS Design

A DEMONSTRATION OF WHAT CAN BE ACCOMPLISHED VISUALLY THROUGH CSS-BASED DESIGN. SELECT ANY STYLE SHEET FROM THE LIST TO LOAD IT INTO THIS PAGE.



# Cascading Style Sheets

## ▶ *Cascading Sytle Sheets (CSS)*

- ▶ Definiert physische Formatierung von HTML-Seiten
- ▶ Formatierung kann definiert werden für
  - ▶ vorhandene logische Formatierungen (Überschriften, Listen, ...)
  - ▶ eigene Formatklassen
  - ▶ einzelne Textblöcke
  - ▶ unterschiedliche Formate für Druck, Bildschirm, ...
- ▶ CSS Versionen
  - ▶ 1.0      1996
  - ▶ 2.0      1998
  - ▶ 2.1      2011
  - ▶ 3.0      2012 (Draft)

# Cascading Style Sheets

- ▶ Vielseitige Gestaltung der physischen Darstellung
  - ▶ Schriftart, Schriftgröße, Schriftstil
  - ▶ Zeichen- und Zeilenabstand, Einrückung
  - ▶ Text- und Hintergrundfarbe
  - ▶ Rahmen
  - ▶ Farben
  - ▶ freie Platzierung und Überlappung von Objekten

# Vorteile von CSS

- ▶ Exakte Bestimmung des Erscheinungsbilds
- ▶ Optimierung für bestimmte **Ausgabemedien**
- ▶ Wichtig für **barrierefreies** Webdesign
- ▶ **Strikte Trennung** zwischen Inhalt und Darstellung
- ▶ Inhalt wird logisch in HTML formatiert
- ▶ Darstellung erfolgt separat (und parallel) in CSS
- ▶ Arbeitsteilung Autor / Designer wird möglich
- ▶ **Wiederverwendung** durch Kaskadierung

# Vorbereitung für CSS-Formatierung

- ▶ HTML-Seite als logische Sequenz von Blöcken
  - ▶ jeder Block wird mit einem Tag gekennzeichnet
  - ▶ mit einem passenden Standard-Tag z.B. `<h1>...</h1>`
  - ▶ oder sonst mit `<div>...</div>`
    - ▶ Elemente, die speziell formatiert werden sollen, aber keinen Block erzeugen sollen, werden durch `<span>...</span>` gekennzeichnet
    - ▶ Reihenfolge entspricht der logischen Reihenfolge
- ▶ Blöcke werden dann mit CSS formatiert, positioniert und ausgerichtet

# Einbinden von CSS: Eigene Datei

- ▶ Eigene CSS-Datei
  - ▶ Normalfall in vielen Web-Projekten
  - ▶ Unterstützt Wiederverwendung (kann von mehreren HTML-Dateien genutzt werden)
  - ▶ Einbinden über `<link>`-Element im Header der Datei
  - ▶ Stylesheets können sich über `@import url(...);` gegenseitig einbinden

```
<link rel="stylesheet"  
      type="text/css"  
      href="css/waw.css"/>
```

```
@import url(main.css);
```

# Einbinden von CSS: Eingebettet

- ▶ Eingebettet in HTML-Datei
  - ▶ Stylesheet nur in dieser Datei gültig
  - ▶ Keine Wiederverwendung
  - ▶ Über `<style>`-Element im Header der Datei

```
<style type="text/css">
  <!--
    /* ... Style-Sheet-Definitionen ... */
    h2 { font-style: italic; }
  -->
</style>
```



# Einbinden von CSS: Inline

- ▶ CSS-Befehle direkt beim HTML-Element
  - ▶ Bei jedem Element über `style`-Attribut möglich
  - ▶ Element `<span>`, um Bereiche für das Styling zu definieren
  - ▶ Verletzt Trennung zwischen Inhalt und Darstellung

```
<p style="color: red;">
  Hallo dies ist ein Absatz. Mit einem
  <span style="font-weight: bold;">fetten</span>
  Stil mittendrin.
</p>
```

# CSS-Selektoren

# Aufbau einer CSS-Regel

- ▶ Regel besteht aus *Selektor* und *Deklarationen*
  - ▶ Der *Selektor* legt fest, für welche Elemente die Regel gilt
  - ▶ Die *Deklarationen* bestimmen die Darstellung der Elemente
  - ▶ Eine Deklaration weist einer *Eigenschaft* einen *Wert* zu (getrennt durch einen Doppelpunkt)
  - ▶ Deklarationen werden durch Semikolon getrennt

```
Selektor { Eigenschaft:Wert; Eigenschaft:Wert; }
```

The diagram illustrates the components of a CSS rule. It shows the text "Selektor { Eigenschaft:Wert; Eigenschaft:Wert; }". Below the opening curly brace, a bracket groups the first "Eigenschaft:Wert;" and is labeled "Deklaration". A second bracket groups the second "Eigenschaft:Wert;" and is also labeled "Deklaration". A large bracket at the bottom groups the entire content within the curly braces (the declarations) and is labeled "Regel".

# Unterschiedliche Selektoren

- ▶ *Universal* (\*) - Alle Elemente
- ▶ *Typ* - Alle Elemente eines Typs (z. B. <h1>)
- ▶ *ID* (#name) - Element mit einer ID
- ▶ *Klasse* (.name) - Elemente, die einer Klasse über das `class`-Attribut zugeordnet wurden
- ▶ *Gruppen* (a, b, c) - Alle Elemente der Gruppe auf einmal
- ▶ *Nachfahre* (a b) - Nachfahren
- ▶ *Kind* (a > b) - Kinder von einem Element sind
- ▶ *Geschwister* (a + b) - Benachbarte Geschwisterelemente
- ▶ *Pseudoklassen* (:klasse) - Eingebaute Klassen

# Beispiel: HTML-Datei für Selektoren

```
<h1>Überschrift Ebene 1</h1>
```

```
<p>Erster Absatz mit etwas Text</p>
```

```
<p>Zweiter Absatz mit auch noch etwas Text</p>
```

```
<h2>Überschrift Ebene 2</h2>
```

```
<ul>
```

```
  <li>Apfel</li>
```

```
  <li>Birne</li>
```

```
</ul>
```

```
<div id="info">
```

```
  Dieser Text hat die ID "info".
```

```
</div>
```

```
<p>Manche Informationen sind <span class="wichtig">wichtig</span> und müssen daher <em  
class="wichtig">hervorgehoben</em> werden.</p>
```

# Universalselektor (\*)

```
/* Alle Elemente in mittelgrau darstellen */  
* { color: #7f7f7f; }
```

- ▶ Wählt alle Elemente aus

## Überschrift Ebene 1

Erster Absatz mit etwas Text

Zweiter Absatz mit auch noch etwas Text

## Überschrift Ebene 2

- Apfel
- Birne

Dieser Text hat die ID "info".

Manche Informationen sind wichtig und müssen daher *hervorgehoben* werden.



# Typselektor

```
/* Überschriften einfärben */  
h1 { color: olive; }  
h2 { color: blue; }
```

- ▶ Wählt bestimmte HTML-Elemente aus

## Überschrift Ebene 1

Erster Absatz mit etwas Text

Zweiter Absatz mit auch noch etwas Text

## Überschrift Ebene 2

- Apfel
- Birne

Dieser Text hat die ID "info".

Manche Informationen sind wichtig und müssen daher *hervorgehoben* werden.

# ID-Selektor

- ▶ Wählt ein Element anhand seiner ID aus

```
/* Element mit ID "info" mit Rahmen versehen */  
#info { border-style: solid;  
        border-width: 3px; }
```

## Überschrift Ebene 1

Erster Absatz mit etwas Text

Zweiter Absatz mit auch noch etwas Text

## Überschrift Ebene 2

- Apfel
- Birne

Dieser Text hat die ID "info".

Manche Informationen sind wichtig und müssen daher *hervorgehoben* werden.

# Klassenselektor

- ▶ Wählt alle Elemente einer CSS-Klasse aus

```
/* Alle Elemente der Klasse "wichtig" einfärben. */  
.wichtig { color: red; font-weight: bold; }
```

## Überschrift Ebene 1

Erster Absatz mit etwas Text

Zweiter Absatz mit auch noch etwas Text

## Überschrift Ebene 2

- Apfel
- Birne

Dieser Text hat die ID "info".

Manche Informationen sind **wichtig** und müssen daher *hervorgehoben* werden.

# Klassenselektor (kontextabhängig)

- ▶ Bestimmte Elemente, die eine CSS-Klasse haben

```
/* Nur "wichtig" bei em einfärben. */  
em.wichtig { color: red; font-weight: bold; }
```

## Überschrift Ebene 1

Erster Absatz mit etwas Text

Zweiter Absatz mit auch noch etwas Text

## Überschrift Ebene 2

- Apfel
- Birne

Dieser Text hat die ID "info".

Manche Informationen sind wichtig und müssen daher ***hervorgehoben*** werden.

# Gruppenselektor

- ▶ Aufzählung aller Selektoren für die die Regel gilt

```
/* Alle Überschriften und wichtigen Elemente in olive */  
h1, h2, .wichtig { color: olive; }
```

## Überschrift Ebene 1

Erster Absatz mit etwas Text

Zweiter Absatz mit auch noch etwas Text

## Überschrift Ebene 2

- Apfel
- Birne

Dieser Text hat die ID "info".

Manche Informationen sind **wichtig** und müssen daher *hervorgehoben* werden.

# Beispiel: HTML-Datei für Selektoren

`<h1>Überschrift Ebene 1</h1>`

`<p>Ein Absatz, der ein <q>Zitat enthält, in dem es eine <em>Hervorhebung</em> gibt.</q>`  
und danach ganz normal weiterläuft. Huch, noch eine `<em>`Hervorhebung</em>, jetzt aber  
außerhalb des Zitates.</p>

`<q>Hier ein Zitat ohne Absatz.</q>`



# Nachfahren-Selektor

- ▶ Formatiert Elemente, die in einem anderen vorkommen

```
/* Zitate in Absätzen grau */
```

```
p q { color: grey; }
```

```
/* Hervorhebungen in Zitaten in Absätzen unterstreichen */
```

```
p q em { text-decoration:underline; }
```

## Überschrift Ebene 1

Ein Absatz, der ein "Zitat enthält, in dem es eine Hervorhebung gibt." und danach ganz normal weiterläuft. Huch, noch eine *Hervorhebung*, jetzt aber außerhalb des Zitates.

"Hier ein Zitat ohne Absatz."

# Nachfahren-Selektor

- ▶ Formatiert Elemente, die in einem anderen vorkommen

```
/* Zitate in Absätzen grau */
```

```
p q { color: grey; }
```

```
/* Hervorhebungen in Zitaten in Absätzen unterstreichen */
```

```
p em { text-decoration:underline; }
```

## Überschrift Ebene 1

Ein Absatz, der ein "Zitat enthält, in dem es eine *Hervorhebung* gibt." und danach ganz normal weiterläuft. Huch, noch eine *Hervorhebung*, jetzt aber außerhalb des Zitates.

"Hier ein Zitat ohne Absatz."

# Kindselektor

- Formatiert direkte Kinder eines Elements

```
/* Hervorhebungen nur direkt in Absätzen unterstreichen */  
p > em { text-decoration:underline; }
```

```
/* Hervorhebungen in Zitaten olive */  
q > em { color: olive; }
```

## Überschrift Ebene 1

Ein Absatz, der ein "Zitat enthält, in dem es eine *Hervorhebung* gibt." und danach ganz normal weiterläuft. Huch, noch eine Hervorhebung, jetzt aber außerhalb des Zitates.

"Hier ein Zitat ohne Absatz."

# Geschwister-Selektor

- Formatiert benachbarte Elemente auf gleicher Ebene

```
/* Ersten Absatz nach Überschrift einfärben */  
h1 + p { color: olive; }
```

## Überschrift Ebene 1

Ein Absatz, der ein "Zitat enthält, in dem es eine *Hervorhebung* gibt." und danach ganz normal weiterläuft. Huch, noch eine *Hervorhebung*, jetzt aber außerhalb des Zitates.

"Hier ein Zitat ohne Absatz."

# Pseudoklassen

- *Pseudoklassen* sind Selektoren zur Auswahl von Elementen, die nicht eigenständig im HTML ausgezeichnet sind

Klasse	Beschreibung
:link	Link im originalen Zustand von <a>
:visited	Link nachdem er angeklickt wurde
:hover	Link, wenn die Maus über dem Link steht
:active	Link, wenn er gerade angeklickt wird
:focus	Element, dass gerade den Fokus hat (Formulare, Links)

# Pseudoklassen

```
/* Links stylen */  
a:link { color: olive; }  
a:visited { color: grey; }  
a:active { color: red; }  
a:hover { font-weight: bold; }
```

Eine ganze Reihe von Links: Link1, **Link2**, Link3, Link4.



# Maßeinheiten

# Regeln für Maßeinheiten

- ▶ Kein Leerzeichen zwischen Zahl und der Maßeinheit  
`9px` nicht `9 px`
- ▶ Komma wird als Punkt geschrieben  
`1.2em` nicht `1,2em`
- ▶ Bei Dezimalzahlen darf eine führende Null entfallen  
`font-size:.6em;`
- ▶ Werte von 0 werden ohne Maßeinheit notiert  
`border:0;`
- ▶ Negative Zahlen werden durch einen Bindestrich markiert  
`-17px`

# Maßeinheiten

## ▶ *Relative Maßeinheiten*

- ▶ orientieren sich an anderen Werten
- ▶ gut geeignet für Bildschirmdarstellung
- ▶ Beispiel: Pixel (px), Prozent (%)

## ▶ *Absolute Maßeinheiten*

- ▶ stehen für sich
- ▶ gut geeignet für Druckausgaben
- ▶ Beispiel: Millimeter (mm) oder Punkt (pt)

# Maßeinheiten - Überblick

Maßeinheit	Beschreibung
px	Pixel. Größe abhängig von der Bildschirmauflösung
em	Höhe des Großen „M“ der gewählten Schriftart
ex	Höhe des kleinen „x“ der gewählten Schriftart
%	Prozent, abhängig vom übergeordneten Element

relativ

Maßeinheit	Beschreibung	in mm
mm	Millimeter	1,000
cm	Zentimeter	10,000
in	Inch (Zoll)	25,399
pt	Pica Point	0,351
pc	Pica	4,217

absolut

# Maßeinheiten

- ▶ **px**
  - ▶ für Darstellung am Bildschirm
  - ▶ verbreitet für Bilder und eingebettete Objekte
  - ▶ bezogen auf die Auflösung des Geräts
- ▶ **pt, pc, cm, mm, in**
  - ▶ für Druckausgabe mit fester Papiergröße
  - ▶ pt üblich für Schriften
  - ▶ cm, mm, in für Positionen und Abstände

# Maßeinheiten

- ▶ Prozent (%) bezogen auf
  - ▶ `font-size` - Schriftgröße des umschließenden Blocks
  - ▶ `width`, `height` - Breite/Höhe des umschließenden Blocks
  - ▶ `margin`, `padding` - Breite/Höhe des eigenen Blocks
- ▶ `em`, `ex` bezogen auf
  - ▶ `font-size` - Schriftgröße des umschließenden Blocks
  - ▶ andere Attribute - Schriftgröße des Elements

# Empfehlung

- ▶ Schriftgröße in % oder **em** (einstellbar im Browser)
- ▶ Ränder, Außen- und Innenabstände in **em** (passend zur Schrift)
- ▶ Bilder in **em** (passend zur Schrift)
- ▶ Größe von Blöcken
- ▶ fensterabhängig in % (falls Umbruch möglich) oder
  - ▶ Position von Blöcken
- ▶ vorzugsweise fließend
- ▶ abwägen: fensterabhängig in % oder schriftabhängig in **em**
- ▶ möglichst nicht absolut positionieren

# Angabe von Farbwerten

- ▶ Notationen für Farbwerte
  - ▶ `rgb(255,140,0)` - Farbanteile rot, grün, blau im Bereich 0...255
  - ▶ `rgb(100%,55%,0%)` - Farbanteile im Bereich 0%...100%
  - ▶ `#FF8C00` - Farbanteile hexadezimal (00...FF)
  - ▶ benannte Farben: black, white, olive, ...
- ▶ Erweiterungen durch CSS3
  - ▶ `hsl(0, 100%, 50%)` - HSL (hue saturation lightness) Farbraum
  - ▶ Angabe eines Alpha-Kanals als vierten Wert



# Benannte CSS-Grundfarben

Named	Numeric	Color name	Hex rgb	Decimal
		black	#000000	0,0,0
		silver	#C0C0C0	192,192,192
		gray	#808080	128,128,128
		white	#FFFFFF	255,255,255
		maroon	#800000	128,0,0
		red	#FF0000	255,0,0
		purple	#800080	128,0,128
		fuchsia	#FF00FF	255,0,255
		green	#008000	0,128,0
		lime	#00FF00	0,255,0
		olive	#808000	128,128,0
		yellow	#FFFF00	255,255,0
		navy	#000080	0,0,128
		blue	#0000FF	0,0,255
		teal	#008080	0,128,128
		aqua	#00FFFF	0,255,255

Quelle: <http://www.w3.org/TR/css3-color/#html4>

# CSS-Attribute

# Übersicht CSS-Attribute

Attributart	Attribute
Schriftformatierung	Schriftart, -stil, -größe, -gewicht, -farbe ...
Absatzformatierung	Texteinrückung und -ausrichtung, Zeilenhöhe und -umbruch
Außenrandabstand	Außenabstand Elementrand zu Nachbarelementen
Innenabstand	Innenabstand Elementinhalt zu Elementrand
Rahmen	Rahmentyp, -dicke und -farbe
Hintergrund	Hintergrundfarben und -bilder
Listenformatierung	Listentyp, -symbol und Einrückung
Tabellenformatierung	Tabellengröße, -rahmen, -zellen
Positionierung	Position, Größe und Anzeigeart von Elementen

# Block- und Inline-Elemente

## ▶ *Block-Elemente*

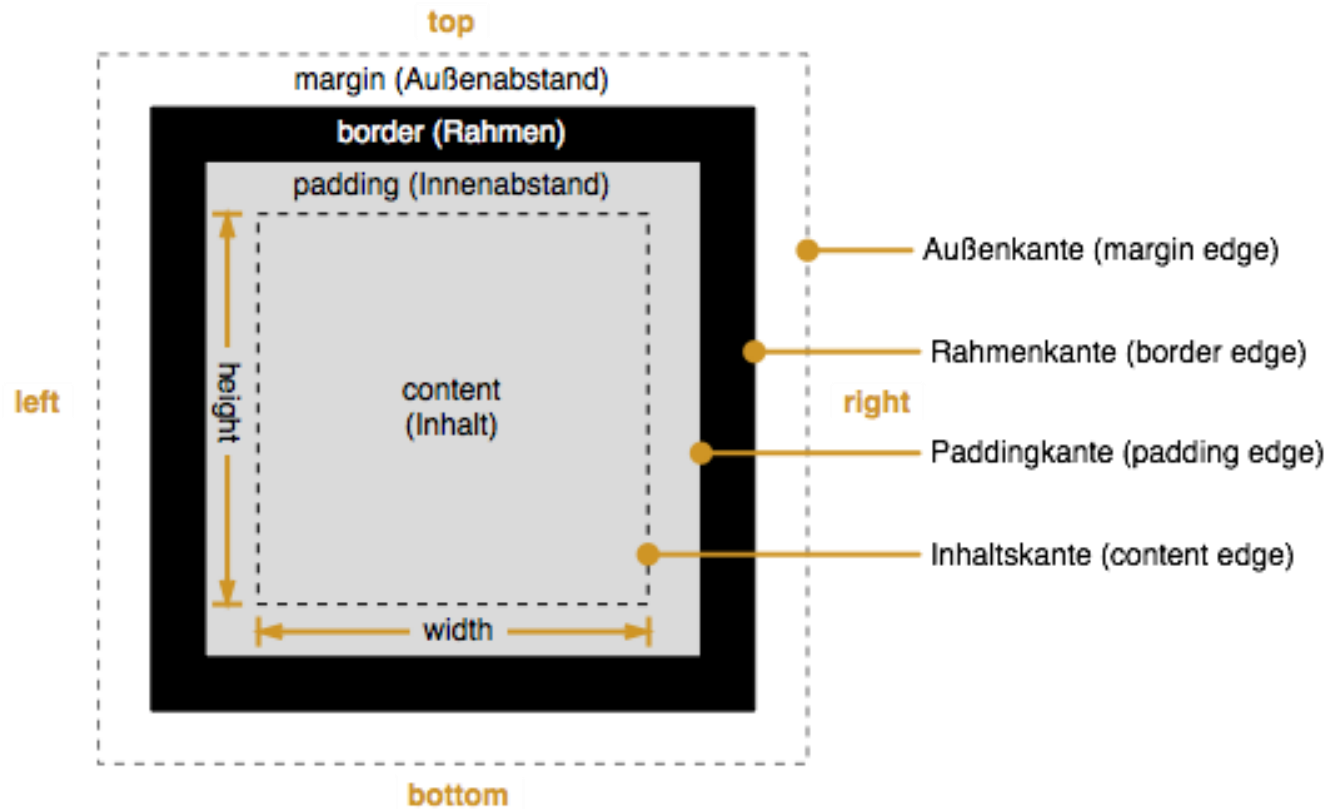
- ▶ Erzeugen einen Absatz
- ▶ Haben eine Box
- ▶ Beispiele: address, blockquote, div, dl, fieldset, form, h1-h6, hr, p, pre, table, ul

## ▶ *Inline-Elemente*

- ▶ Erzeugen keinen Absatz, fließen im Text mit
- ▶ Haben keine Box
- ▶ Beispiele: a, abbr, acronym, b, big, br, button, cite, code, dfn, em, i, img, input, kbd, label, map, object, q, samp, script, select, small, span, strong, sub, sup, textarea, tt, var

# Box-Modell

- CSS organisiert die Block-Elemente mithilfe von Boxen



# Randangaben für die Box

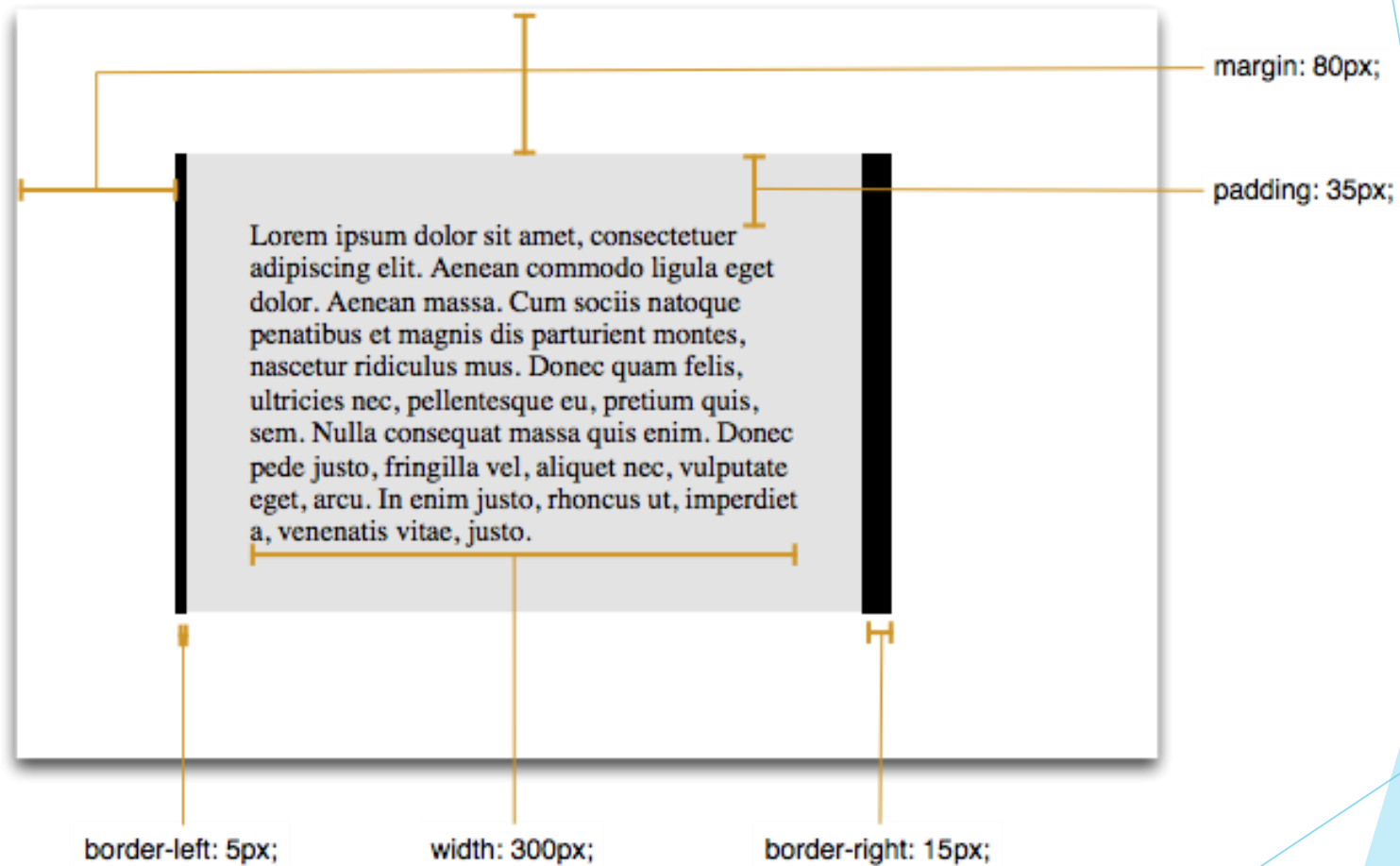
- ▶ Ränder für die Box können per CSS festgelegt werden
  - ▶ `padding` - Innenabstand
  - ▶ `border` - Breite des Rahmens
  - ▶ `margin` - Außenabstand
- ▶ Die vier Seiten können einzeln adressiert werden
  - ▶ `-top` - oben
  - ▶ `-right` - rechts
  - ▶ `-bottom` - unten
  - ▶ `-left` - links

# Beispiel: Box

```
p.boxed {  
  width: 300px;  
  padding: 35px;  
  margin: 80px;  
  border: #353d2d;  
  border-right: 15px solid;  
  border-left: 5px solid;  
  background-color: #e3e3e3;  
}
```

<p class="boxed">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo.</p>

# Beispiel: Box





# Kurzschrift für Randangaben

- ▶ Statt die vier Seiten der Box explizit anzusprechen, kann man auch eine Kurzform verwenden
  - ▶ Angaben werden beginnend bei top im Uhrzeigersinn notiert
  - ▶ zwei Angaben: top und bottom
  - ▶ drei Angaben: für top, left/right und bottom

```
padding-top: 5px;  
padding-right: 10px;  
padding-bottom: 15px;  
padding-left: 25px;
```



```
padding: 5px 10px 15px 25px;
```

```
padding-top: 20px;  
padding-right: 5px;  
padding-bottom: 20px;  
padding-left: 5px;
```



```
padding: 20px 5px;
```

# Elementgröße

- ▶ Angabe der gewünschten Größe
- ▶ width - Breite
- ▶ height - Höhe
- ▶ Angabe der maximalen und minimalen Größe
- ▶ min-width, max-width - minimale und maximale Breite
- ▶ min-height, max-height - minimale und maximale Höhe
- ▶ overflow - Was passiert, wenn Inhalt nicht zur Größe passt
- ▶ visible - Inhalt ragt über den Rand hinaus
- ▶ hidden - Inhalt wird am Rand abgeschnitten
- ▶ scroll - Inhalt wird verborgen, es gibt aber Scrollbalken
- ▶ auto - Browser entscheidet, wo Scrollbalken benötigt werden

# Beispiel: Behandlung von Overflow

overflow: visible;

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim.

overflow: hidden;

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec

overflow: scroll;

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus

overflow: auto;

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes,

# Anzeigeart

- ▶ **display** - legt fest, wie ein Element angezeigt wird
- ▶ **none** - nicht anzeigen und auch keinen Platz dafür freihalten
- ▶ **inline** - im laufenden Textfluss anzeigen
- ▶ **block** - Block erzwingen, d.h. in neuer Zeile anzeigen
- ▶ **list-item** - wie block, aber mit Aufzählungszeichen davor
- ▶ **run-in** - kontextabhängig als block oder inline darstellen
- ▶ **visibility** - legt fest, ob ein Element angezeigt wird
- ▶ **hidden** - nicht anzeigen und auch keinen Platz dafür freihalten
- ▶ **visible** - anzeigen, Default

# Beispiel: display

## **display: inline**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim.

## **display: block**

Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Aenean commodo ligula eget dolor. Aenean massa.  
Cum sociis natoque penatibus et magnis dis parturient montes,  
nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque  
eu, pretium quis, sem. Nulla consequat massa quis enim.

## **display: none**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cum sociis  
natoque penatibus et magnis dis parturient montes, nascetur  
ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu,  
pretium quis, sem. Nulla consequat massa quis enim.

# Rahmen-Design

- ▶ **border-color** - Rahmenfarbe
- ▶ Einschränkung auf Seite über **border-top-color**, **border-left-color** ...
- ▶ **border-style** - Rahmenstil
- ▶ **none** - kein Rahmen
- ▶ **dotted** - gepunktet
- ▶ **dashed** - gestrichelt
- ▶ **solid** - durchgezogen
- ▶ **double** - doppelt durchgezogen
- ▶ **inset** - 3D-Effekt eingedrückt
- ▶ **outset** - 3D-Effekt hervorstehend
- ▶ Einschränkung auf Seite über **border-top-style**, **border-left-style** ...



# Schriftformatierung

- ▶ **font-family** - Liste (kommasetrennt) von Zeichensätzen
- ▶ Angabe einer konkreten Schrift (z.B. Helvetica, Times New Roman)
- ▶ oder einer Schriftfamilie (z.B. serif, sans-serif, monospace)
- ▶ mehrere Angaben können gemacht werden, passende wird gewählt
- ▶ **font-style** - Schriftstil
- ▶ **normal** - normal
- ▶ **italic** - *kursiv*
- ▶ **oblique** - *schräggestellt*
- ▶ **font-size** - Schriftgröße
- ▶ **color** - Textfarbe

# Schriftformatierung

- ▶ **font-weight** - Schriftgewicht
  - ▶ lighter - fein
  - ▶ normal - normal
  - ▶ bold - **fett**
  - ▶ bolder - extra fett
- ▶ **text-decoration** - Textdekoration
  - ▶ none - keine
  - ▶ underline - unterstrichen
  - ▶ overline - überstrichen
  - ▶ line-through - durchgestrichen
  - ▶ blink - **blinkend** (bitte nicht!)



# Absatzformatierung

- ▶ **text-indent** - Erste Zeile eines Textes einrücken
- ▶ negative Werte definieren eine Ausrückung
- ▶ **text-align** - Horizontale Textausrichtung
- ▶ **left** - linksbündig
- ▶ **center** - zentriert
- ▶ **right** - rechtsbündig
- ▶ **justify** - Blocksatz

**Linksbündig:** Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

**Zentriert:** Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

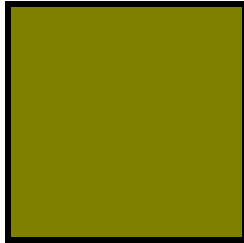
**Rechtsbündig:** Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

**Blocksatz:** Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

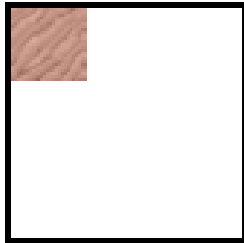
# Hintergrund

- ▶ `background-color` - Hintergrundfarbe
- ▶ `background-image` - Hintergrundbild
- ▶ `background-repeat` - Wiederholen des Hintergrundbilds
- ▶ `repeat` - Wiederholen in X- und Y-Richtung (Standardwert)
- ▶ `repeat-x` - Wiederholen nur in X-Richtung
- ▶ `repeat-y` - Wiederholen nur in Y-Richtung
- ▶ `no-repeat` - Keine Wiederholung
  - ▶ `background-position` - Position des Bildes (bei `no-repeat`)
- ▶ horizontale Ausrichtung: `top`, `bottom`, `center`
- ▶ vertikale Ausrichtung: `left`, `right`, `center`

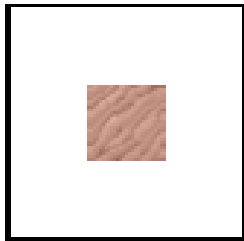
# Hintergrund



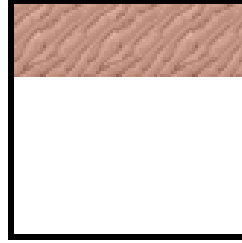
`background-color: olive`



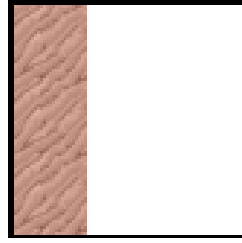
`background-image: url(bilder/sand.png);`  
`background-repeat: no-repeat;`



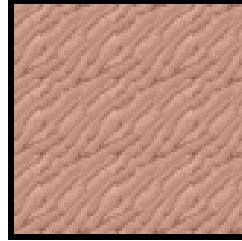
`background-image: url(bilder/sand.png);`  
`background-repeat: no-repeat;`  
`background-position:center center;`



`background-image: url(bilder/sand.png);`  
`background-repeat: repeat-x;`



`background-image: url(bilder/sand.png);`  
`background-repeat: repeat-y;`



`background-image: url(bilder/sand.png);`  
`background-repeat: repeat;`

# Listenformatierung

- ▶ **list-style-type** - Darstellung für geordnete Listen (**ol**)
  - ▶ decimal - arabische Numerierung: 1., 2., 3., 4. ...
  - ▶ lower-roman - römische Kleinbuchstaben: i., ii., iii., iv. ...
  - ▶ upper-roman - römische Großbuchstaben: I., II., III., IV. ...
  - ▶ lower-alpha - Kleinbuchstaben: a., b., c., d. ...
  - ▶ upper-alpha - Großbuchstaben: A., B., C., D. ...
  - ▶ ... - weitere Stile für andere Sprachen
- ▶ **list-style-type** - Darstellung für ungeordnete Listen (**ul**)
  - ▶ disc - gefüllter Kreis •
  - ▶ circle - hohler Kreis ○
  - ▶ square - Quadrat ■
  - ▶ none - kein Zeichen

# Beispiele für Listen-Stile

list-style-type: disc

- Äpfel
- Orangen
- Birnen

list-style-type: circle

- Äpfel
- Orangen
- Birnen

list-style-type: square

- Äpfel
- Orangen
- Birnen

list-style-type: none

Äpfel  
Orangen  
Birnen

list-style-type: decimal

1. Äpfel
2. Orangen
3. Birnen

list-style-type: lower-roman

- i. Äpfel
- ii. Orangen
- iii. Birnen

list-style-type: upper-roman

- I. Äpfel
- II. Orangen
- III. Birnen

list-style-type: lower-alpha

- a. Äpfel
- b. Orangen
- c. Birnen

list-style-type: upper-alpha

- A. Äpfel
- B. Orangen
- C. Birnen

list-style-type: lower-greek

- α. Äpfel
- β. Orangen
- γ. Birnen

list-style-type: hebrew

- א. Äpfel
- ב. Orangen
- ג. Birnen

list-style-type: hiragana

- あ. Äpfel
- い. Orangen
- う. Birnen

list-style-type: katakana

- ア. Äpfel
- イ. Orangen
- ウ. Birnen

list-style-type: georgian

- ა. Äpfel
- ბ. Orangen
- გ. Birnen

# CSS-Layout

# Positionierung von Elementen

- ▶ **position** - Positionierung eines Elements
- ▶ **static** - normaler Fluss
- ▶ **relative** - Position relativ zur Umgebung
- ▶ **absolute** - absolute Position
- ▶ **fixed** - Feste Position
- ▶ **Angabe der Position**
- ▶ **top** - Abstand von oben
- ▶ **left** - Abstand von links
- ▶ **right** - Abstand von rechts
- ▶ **bottom** - Abstand von unten

# Positionierung von Elementen

- ▶ **static** - normaler Fluss, der sich aus dem Umbruch ergibt
- ▶ Standardeinstellung
- ▶ Abstandsangaben (**top**, **bottom**, **left**, **right**) werden ignoriert

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim.

**position: static;**

Nam eget dui. Etiam rhoncus. Maecenas tempus, tellus eget condimentum rhoncus, sem quam semper libero, sit amet adipiscing sem neque sed ipsum. Nam quam nunc, blandit vel, luctus pulvinar, hendrerit id, lorem. Maecenas nec odio et ante tincidunt tempus. Donec vitae sapien ut libero venenatis faucibus.



# Positionierung von Elementen

- ▶ relative
- ▶ verändert Position relativ zur Lage, die sich aus normalem Fluss ergibt
- ▶ ursprünglicher Platz für den Container bleibt erhalten

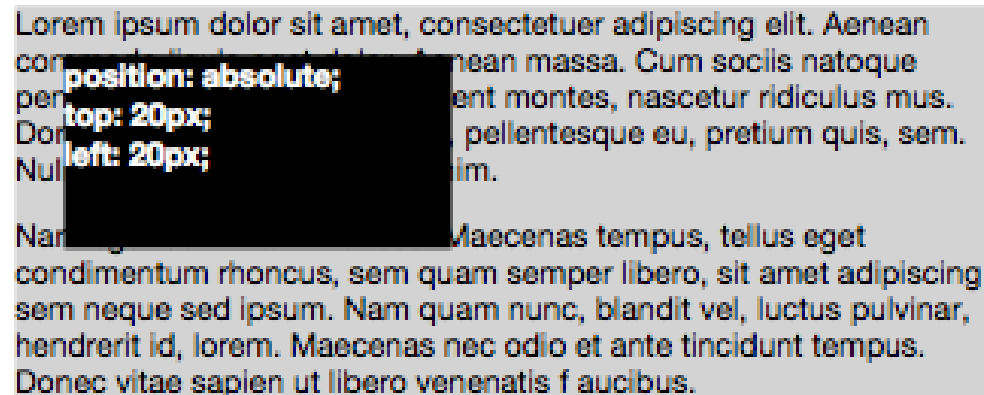
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim.

```
position: relative;  
top: 20px;  
left: 20px;
```

Nam eget dui. Etiam rhoncus. Maecenas tempus, tellus eget condimentum rhoncus, sem quam semper libero, sit amet adipiscing sem neque sed ipsum. Nam quam nunc, blandit vel, luctus pulvinar, hendrerit id, lorem. Maecenas nec odio et ante tincidunt tempus. Donec vitae sapien ut libero venenatis faucibus.

# Positionierung von Elementen

- ▶ absolute
- ▶ gibt eine absolute Position an
- ▶ Position bezieht sich
  - ▶ auf den Rand des umgebenden Elements, wenn dieses selber positioniert wurde (**relative**, **absolute** oder **fixed**) oder
  - ▶ andernfalls auf das Browserfenster
- ▶ ursprünglicher Platz wird freigegeben

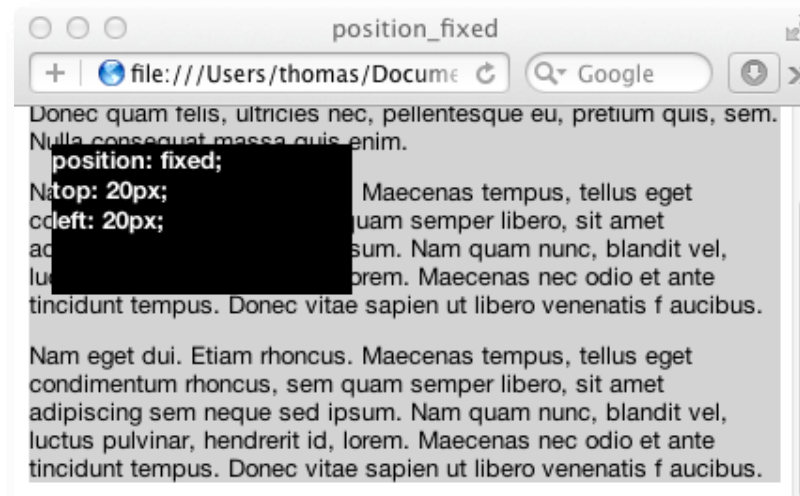
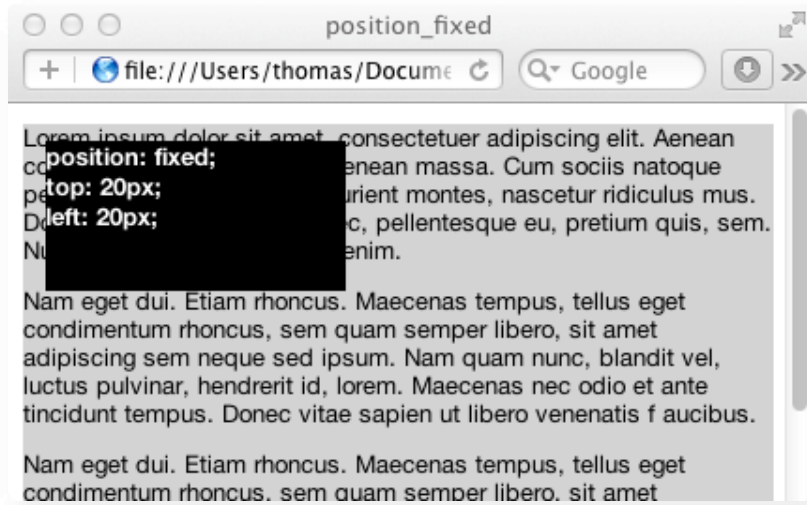


```
position: absolute;  
top: 20px;  
left: 20px;
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean  
consectetur adipiscing elit. Aenean massa. Cum sociis natoque  
penetremt montes, nascetur ridiculus mus.  
Dor pellentesque eu, pretium quis, sem.  
Nullam.  
Nam Maecenas tempus, tellus eget  
condimentum rhoncus, sem quam semper libero, sit amet adipiscing  
sem neque sed ipsum. Nam quam nunc, blandit vel, luctus pulvinar,  
hendrerit id, lorem. Maecenas nec odio et ante tincidunt tempus.  
Donec vitae sapien ut libero venenatis faucibus.

# Positionierung von Elementen

- ▶ fixed
- ▶ gibt eine absolute Position an
- ▶ Position bezieht sich immer auf das Browserfenster
- ▶ Element bleibt auch beim Scrollen stehen



# Positionierung auf der Z-Achse

- ▶ **z-index** - legt Reihenfolge bei Überlappung fest
  - ▶ wird als Ganzzahl angegeben
  - ▶ je größer die Zahl, desto weiter vorne das Element
  - ▶ relevant vor allem bei absolute- und fixed-Positionierung

z-index: 1;

z-index: 2;

z-index: 3;

# Layout mit Floats

- ▶ float - Block aus dem normalen Fluss herausnehmen
  - ▶ left - links positionieren
  - ▶ right - rechts positionieren
  - ▶ none - im normalen Textfluss einbetten (Standard)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim.

**float: left;**  
**width: 160px;** Nam eget dui. Etiam rhoncus. Maecenas tempus, tellus eget condimentum rhoncus, sem quam semper libero, sit amet adipiscing sem neque sed ipsum. Nam quam nunc, blandit vel, luctus pulvinar, hendrerit id, lorem. Maecenas nec odio et ante tincidunt tempus. Donec vitae sapien ut libero venenatis faucibus.

Nam eget dui. Etiam rhoncus. Maecenas tempus, tellus eget condimentum rhoncus, sem quam semper libero, sit amet adipiscing sem neque sed ipsum. Nam quam nunc, blandit vel, luctus pulvinar, hendrerit id, lorem. Maecenas nec odio et ante tincidunt tempus. Donec vitae sapien ut libero venenatis faucibus.

**float: right;**  
**width: 160px;**

**float: none;**  
**width: 160px;**

Nam eget dui. Etiam rhoncus. Maecenas tempus, tellus eget condimentum rhoncus, sem quam semper libero, sit amet adipiscing sem neque sed ipsum. Nam quam nunc, blandit vel, luctus pulvinar, hendrerit id, lorem. Maecenas nec odio et ante tincidunt tempus. Donec vitae sapien ut libero venenatis faucibus.

# Layout mit Floats

- ▶ **clear** - legt Positionierung nach Floats fest
  - ▶ **left** - unterhalb des letzten links ausgerückten Blocks
  - ▶ **right** - unterhalb des letzten rechts ausgerückten Blocks
  - ▶ **both** - unterhalb des letzten ausgerückten Blocks



**clear: both;**

# Dreispaltiges Layout mit Floats

Kopfzeile: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor.		
<b>float: left; width: 15%;</b>  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim.	<p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim.</p> <p> Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus.</p> <p> Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi.</p> <p> Nam eget dui. Etiam rhoncus. Maecenas tempus, tellus eget condimentum rhoncus, sem quam semper libero, sit amet adipiscing sem neque sed ipsum. Nam quam nunc, blandit vel, luctus pulvinar, hendrerit id, lorem. Maecenas nec odio et ante tincidunt tempus. Donec vitae sapien ut libero venenatis faucibus. Nullam quis ante. Etiam sit amet orci eget eros faucibus tincidunt. Duis leo. Sed fringilla mauris sit amet nibh. Donec sodales sagittis magna. Sed consequat, leo eget bibendum sodales, augue velit cursus nunc,</p>	<b>float: right; width: 15%;</b>  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes.
<b>clear: both; Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu.</b>		







# Kaskadierung und Vererbung

# Kaskadierung von Formaten

- ▶ Wenn es für ein Element mehrere Regeln für ein Attribut (z. B. `color`) gibt, dann gilt folgende Rangfolge
  - ▶ kontextabhängig individuell (`p#id`)  
vor individuell (`#id`)  
vor kontextabhängiger Klasse (`p.achtung`)  
vor allgemeiner Klasse (`.warnung`)  
vor redefiniertem Standard Format (`p`)
  - ▶ Innerhalb einer Ebene hat `!important` Vorrang
  - ▶ Reihenfolge in der Datei ist das letzte Kriterium
  - ▶ inline vor embedded vor externem Stylesheet

# Beispiel: Kaskadierung

```
p#hervorhebung { color: white; }  
p { color: olive; }  
.achtung { color: green; }  
p.achtung { color: orange; }  
p#hervorhebung { color: red; }  
#hervorhebung { color: grey; }
```

Dies soll hervorgehoben werden.

Und das hier auch.

Dies soll hervorgehoben werden.

```
<p id="hervorhebung" class="achtung">Dies soll hervorgehoben werden.</p>
```

```
<p id="auch" class="achtung">Und das hier auch.</p>
```

```
<p id="noch" class="achtung" style="color: green;">Dies soll hervorgehoben werden.</p>
```

# Vererbung von Formaten

- ▶ Wenn es für ein Element kein direktes Format gibt, wird das Format des übergeordneten Elements übernommen (*Vererbung von Formaten*)
  - ▶ Vererbung folgt dem Dokumenten-Baum
  - ▶ Manche Attribute werden nicht vererbt (z. B. text-decoration)

# Beispiel: Vererbung

```
body { font-family: monospace; }  
div { font-family: sans-serif; font-size: 90%; }  
div ul { color: olive; text-decoration: none; }
```

`<p>`Ein Absatz mit etwas Text. Darin eine `<em>`Hevorhebung`</em>`.`</p>`

`<div>`Hier ein Abschnitt

`<h1>`Mit Überschrift`</h1>`

```
<ul>  
  <li>Eins <em>Hervorhebung</em></li>  
  <li>Zwei <em>Hervorhebung</em></li>  
</ul>  
</div>
```

Ein Absatz mit etwas Text. Darin eine Hervorhebung.

Hier ein Abschnitt

## Mit Überschrift

- Eins Hervorhebung
- Zwei Hervorhebung