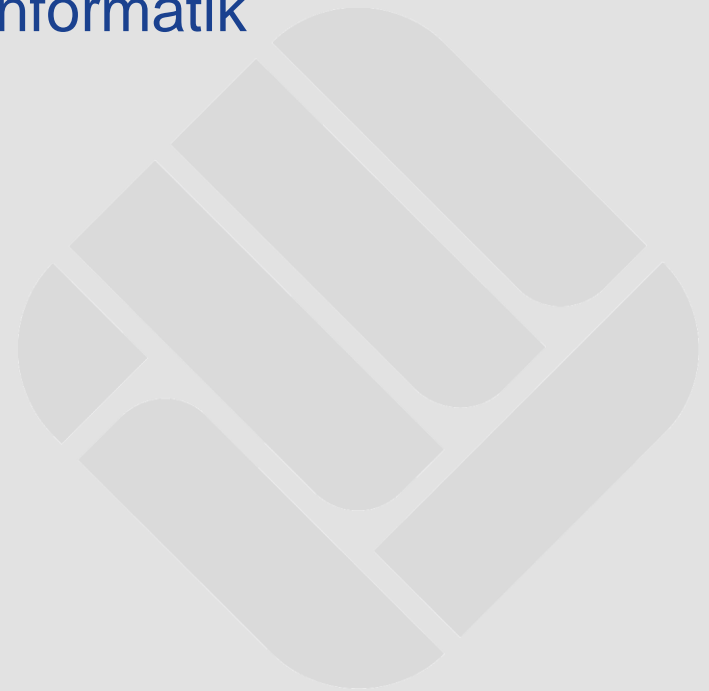


HTML5 und CSS3

Studiengang: Bachelor Wirtschaftsinformatik
Studienfach: Web-Engineering



Referenten: Jonas Brandhorst, Antonios Kouklidis, Andre Eenhuis, Carsten Schober



Gliederung

- HTML5 im Überblick
- Neue und veränderte Funktionen in HTML5
- CSS im Überblick
- Veränderungen und Neuerungen in CSS3
- Responsive Design
- Ausblick in die Zukunft

HTML5 im Überblick

- › Auszeichnungssprache
- › Nachfolger der Version 4.0.1
- › Es wird noch nicht empfohlen (keine W3C-Recommendation)
 - › Probleme der flächendeckenden Browserunterstützung

- › Alter Standard ist für aktuelle Anwendungen nicht konzipiert
 - › Ursprünglich zum Austausch von Textdokumenten
- › Verhindern von zu großer Plugin-Vielfalt
 - › Kein Nachinstallieren mehr von Flash, Java ...
 - › Einschränken von Browserproblemen
- › Einfacher für Web-Entwickler und Endanwender
- › Neue und erweiterte Funktionalitäten



HTML5 - Implementierung heute

HTML  TEST



- › Besseres Layout
- › Höhere Barrierefreiheit für unterschiedliche Endgeräte
- › Beispiele für neue Tags:
 - › `<article>` (z.B. Blogeintrag)
 - › `<canvas>` (Bitmap-Malfläche)
 - › `<footer>` (Fußzeile eines Dokumentes)
 - › `<time>` (Zeitangabe)

Neue und veränderte Funktionen in HTML5

Beispiele für veränderte Tags und Attribute

- › <a> (href ist jetzt optional)
- › (value als Attribut ergänzt → manuelle Nummerierung bei)
- › class ist jetzt ein globales Attribut
- › <!DOCTYPE>

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional  
//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.  
dtd">
```



```
<!DOCTYPE HTML>
```

- Audio-/Video-Einbettung direkt in HTML5
 - Immer mehr Multimedia-Dateien auf unterschiedlichen Webseiten
 - Nicht auf Plugins wie Adobe Flash angewiesen sein
 - Vermehrter Einsatz von mobilen Endgeräten
 - iPhone/iPad unterstützt kein Flash
 - Kein einheitlicher Codec in den unterschiedlichen Browsern
 - H.264: Aktuell bester Codec, aber kein Open-Source
- Einführung der Medien-Elemente `<audio>` und `<video>`
 - Ressourcenschonender im Vergleich zu Flash
 - Wird von fast jedem aktuellem Webbrowser unterstützt

› Beispiel: <audio>

```
<audio src="test.mp3" controls>  
  <a href="test.mp3">download test.mp3</a>  
</audio>
```

Erhalt der Steuerbuttons

› Beispiel <video>

```
<video src="test.mp4" width="427" height="240" controls  
  poster="statischesBild.jpg">  
  <a href="test.mp4">download test.mp4</a>  
</video>
```

Fenstergröße

Standbild

Weiteres mögliches Attribut: media

› JavaScript-API:

- › Individuelle Gestaltung durch Entwickler
- › Hinzufügen von zusätzlichen Funktionen/Informationen
- › Ermitteln der vom Browser unterstützten Mediendateien

› Beispiele:

- › Abfragen von Statuswerten
- › Verändern der Abspielgeschwindigkeiten
- › Fortschrittsanzeige
- › Erstellen eigener Steuerungselemente

➤ Beispiel: Erstellen eigener Steuerungselemente

```
<p>  
    <button id="start "  
        onclick="start ()">Starten/Pausieren</button>  
</p>
```

```
\\Starten/Pausieren  
function start () {  
    var video = document.getElementById( 'video ' );  
    if ( video.paused ) {  
        video.play ();  
    }  
    else {  
        video.pause ();  
    }  
}
```

› Drag&Drop-API

- › Attribut draggable
- › Erfunden von Microsoft, jedoch beste Umsetzung in Firefox
- › Alle Elemente in HTML5 sind draggable (mit Maus)
- › Problem: Barrierefreiheit?

› File-API

- › Hochladen von mehreren Dateien zusammen (z.B. Ordner aus Betriebssystem)
- › Zerteilen von Dateien für höhere Performance möglich
- › Gut kombinierbar mit Drag&Drop-API

- › Früher aufwendige JavaScript-Konstrukte notwendig
- › Neue Input-Typen und Funktionen für Eingabe-Elemente
- › Problem: kein Browser unterstützt aktuell alle Funktionen perfekt
 - › Opera ist Vorreiter
 - › Bei Nichtunterstützung werden neue Funktionen ignoriert
- › Eingabefelder auch außerhalb von `<form id="test">` möglich
 - › Zuweisung dann mittels Attribut `form="test"`
- › Validierungsregeln angepasst → auch ohne Fremdsoftware
 - › Neue Pseudoklassen für CSS zur Anzeige von Live-Validierungsmeldungen

› Beispiele:

- › `<input type="search">`
 - › Layout von Browser abhängig
 - › Unterscheidet sich auch optisch von Textfeldern
 - › `<input type="url">`
 - › `<input type="email">`
- } Validierung auf Gültigkeit
- › `<input type="datetime">` → mit Zeitzone
 - › Beispiel: "2014-06-23T10:59z" → z = UTC-Zeitzone
 - › `<input type="datetime-local">` → ohne Zeitzone
 - › Beispiel: "2014-06-23T11:59"

› Weitere Beispiele:

› <keygen>

- › Erzeugt Schlüsselpaare
- › Erstellt clientseitige Sicherheitszertifikate
- › RSA-Verschlüsselung wird genutzt

› Bei Formularen ist im Standard Auto-Vervollständigung aktiviert

› <datalist>

- › Alternative zu Drop-Down-Feld mit JavaScript
- › Vorschlagsliste bei Eingabefeldern der Auto-Vervollständigung (list-Attribut)

```
<input type="text" list="himmelsrichtungen">
<datalist id="himmelsrichtungen">
  <option value="norden"></option>
  <option value="sueden"></option>
  <option value="westen"></option>
  <option value="osten"></option>
</datalist>
```

- › Normale Webanwendungen
 - › Caching von Inhalten
 - › Keine aktive Internetverbindung nötig

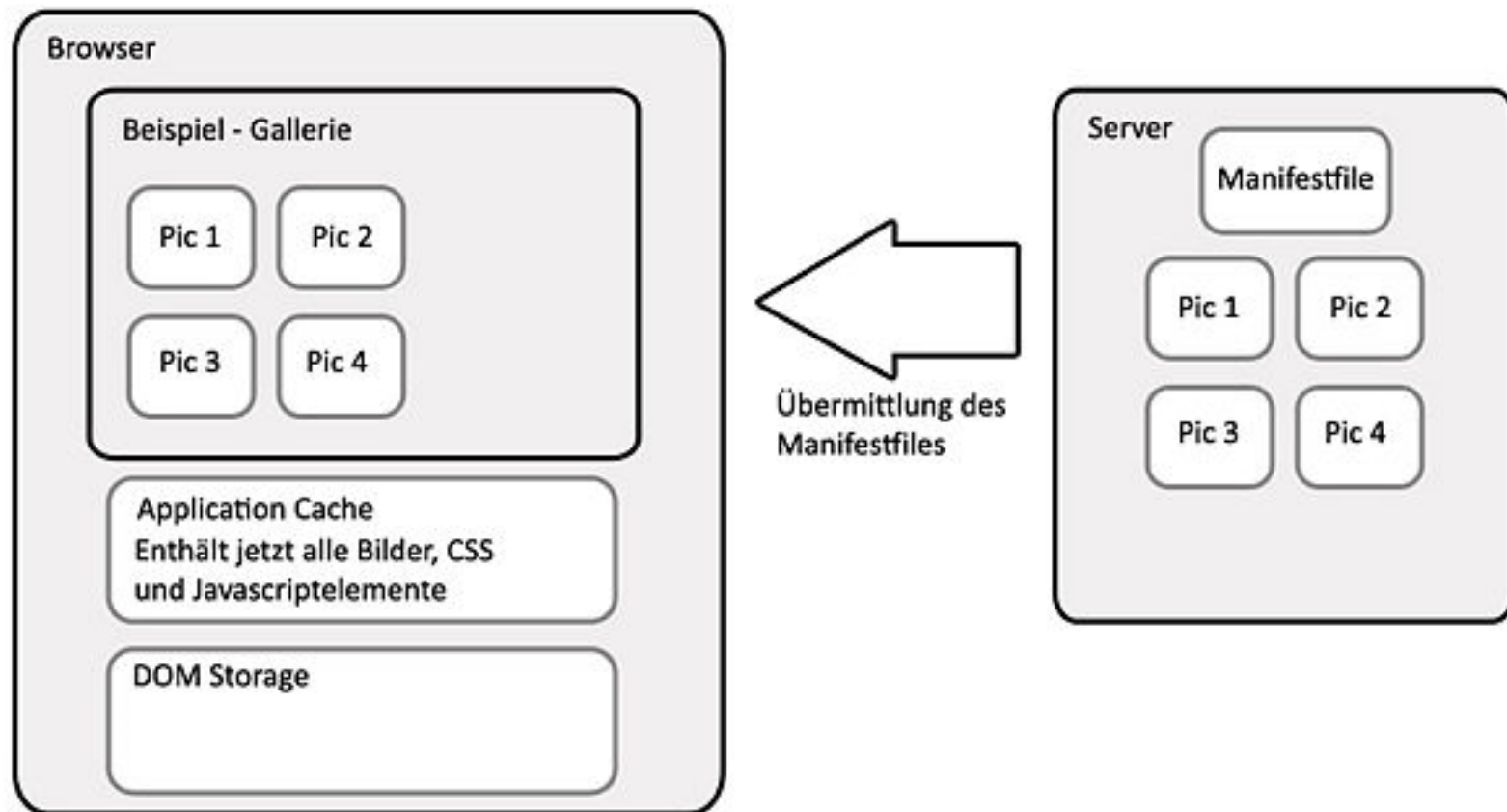
- › Synchronisation bei Wiederverbindung
 - › Jedoch nur für Hybriden

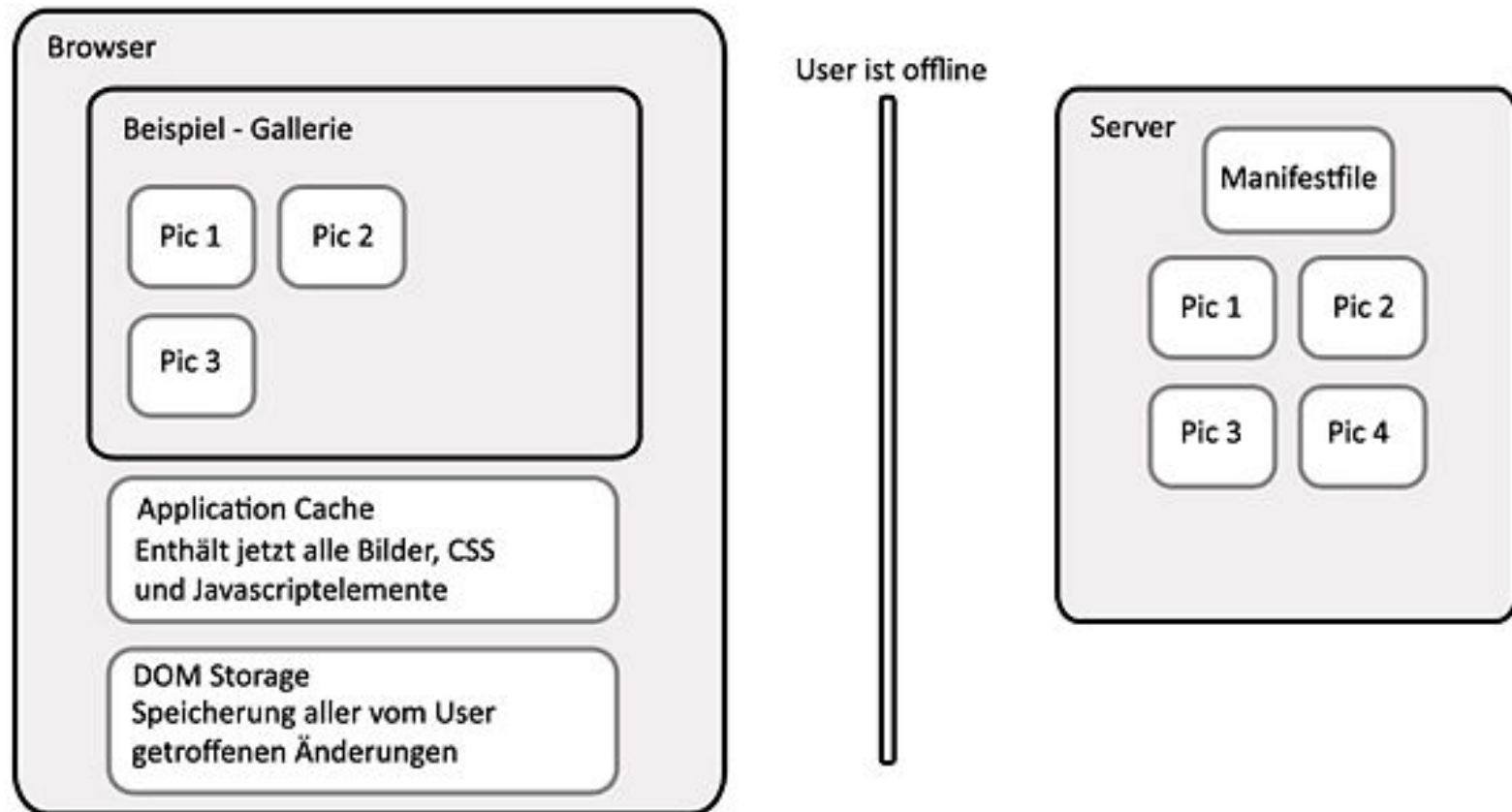
› Application Cache

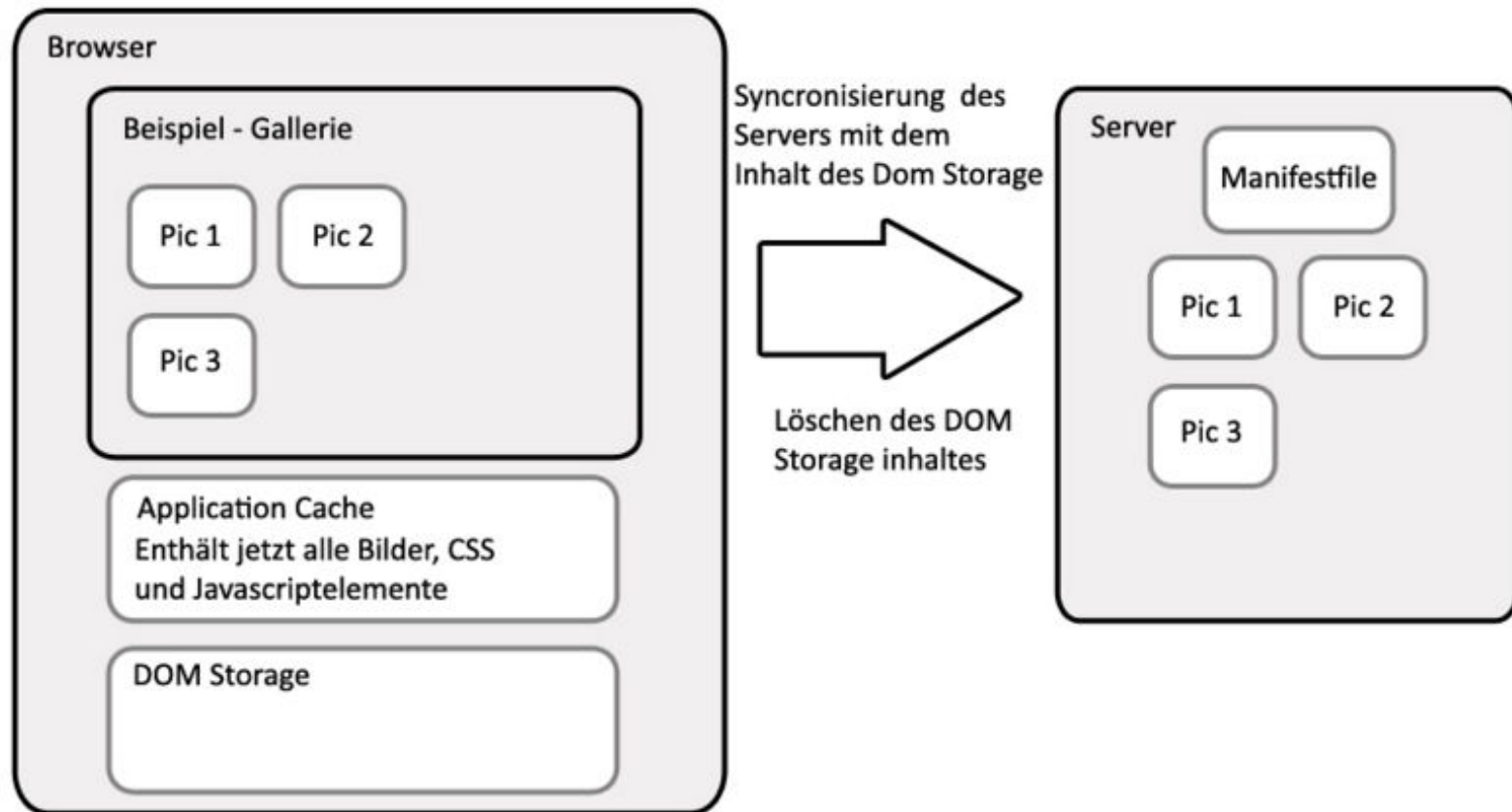
- › Durch Webseite steuerbar
- › Automatisch synchronisiert
- › Cacheinhalte → Manifest-File

› DOM – Storage

- › Haltbar
- › Große Kapazität
- › Session Storage ↔ Local Storage







- › Ansatz Performance steigern
 - › Durch Multithreading
- › Ursprünglich Teil der HTML5 - Spezifikation
 - › WHATWG
 - › W3C

- › Browser muss Web Workers unterstützen

- › Beispiel

- › Mehrere DOM-Baum-Updates
- › Komplexe JavaScript-Berechnung

- › Folge

- › Blockierung des UI-Threads

› Lösung

› JavaScript-Berechnungen / DOM-Baum-Updates

› Parallele Threads zum UI-Thread

Website.html

```
<script>
function webWorker(){
    var worker = new Worker('workerTask.js');
    var result = worker.postMessage('Aufruf');
}
</script>
```

workerTask.js

```
...
...
self.postMessage('Antwort');
}
```

Neue und veränderte Funktionen in HTML5

Shared Workers

- › Sonderform der Web Workers
 - › Nicht weit verbreitet
- › Kommunikation der Informationen über Ports

Website.html

```
<script>  
    var worker = new SharedWorker('sharedWorkerTask.js');  
    worker.port.onmessage = function(e)  
        var result = e.data;  
</script>
```

sharedWorkerTask.js

```
onconnect = function(e) {  
    var port = e.ports[0];  
    port.postMessage('Result');  
}
```


CSS im Überblick

- › Cascading Style Sheets
- › Deklarative Sprache
- › Publiziert Dezember 1996 vom W3C
- › Entwickelt von Håkon Wium Lie und Bert Bos

› Individualisierung von Attributen

- › Gestaltung von Schriftfarben und -arten, Tabellengrößen, Zeilenabständen u.v.m.
- › Anpassungen können zentral vorgenommen werden → einheitlicher „Look“

```
div {  
  background-color: red;  
}  
  
h1, p {  
  color: yellow;  
}
```

```
<html>  
  <head>  
    <link rel="stylesheet" type="text/css" href="stylesheet.css">  
  <body>  
    <div>  
      <h1>Hallo Welt!</h1>  
      <p>Kleines Beispiel</p>  
    </div>  
  </body>  
</html>
```

- › HTML-Dokument referenziert Stylesheet → CSS-Datei für Formatierung
- › CSS-Datei enthält Selektoren, die Elemente ansteuern und anpassen können

› Vorteile CSS

- › Einheitliche Webpräsenz
- › Einfachere und schnellere Gestaltung von Webseiten
- › Kürzere Ladezeiten durch weniger Code und Caching
- › Gestaltung für unterschiedliche Medien

› Nachteile CSS

- › Uneinheitliche Unterstützung durch Browser

- › Komplette abwärtskompatibel
- › Modularer Aufbau
 - › Teiltechniken (Selektoren, Sprachausgabe) können unabhängig voneinander entwickelt werden
- › Anpassung an Elementen
 - › Neue Funktionen für Ränder, Hintergründe, Texteffekte, Attribut-Selektor
- › Weitere Module
 - › Mehrspaltige Layouts, speech

- › Anwendung der neuen Funktion des Attribut-Selektors
 - › Kann gezielt den Textinhalt eines Attributes ansteuern
- › Syntax:
 - › `h1[title*=„onf“]` berücksichtigt alle Zeichenketten mit dem Schlüsselwort `title`, die ein **onf** **enthalten**.
 - › `p[title$=„Zeile“]` berücksichtigt alle Zeichenketten mit dem Schlüsselwort `title`, die mit **Zeile** **enden**.
 - › `p[title^=„Zeile“]` berücksichtigt alle Zeichenketten mit dem Schlüsselwort `title`, die mit **Zeile** **beginnen**.



Veränderungen und Neuerungen in CSS3

Beispiel: Attributselektor

```
div[title*="onf"] {  
  background-color: red;  
}
```

```
h1[title$="Zeile"], p[title^="Zeile"] {  
  color: yellow;  
}
```

```
<html>  
  
  <head>  
  
    <link rel="stylesheet" type="text/css" href="stylesheet.css">  
  
  <body>  
  
    <div title="config">  
      <h1 title="ersteZeile">Hallo Welt!</h1>  
      <p title="zweiteZeile">Kleines Beispiel</p>  
    </div>  
  
  </body>  
  
</html>
```

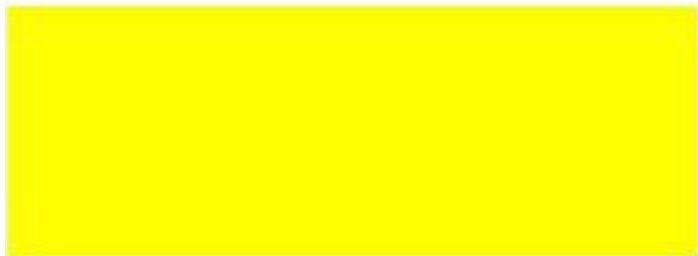
- › border-color: Ränder farbig und Farbverlauf
- › border-image: Grafiken als Randmarkierung
- › border-radius: Abgerundete und flache Ecken/Kanten
- › box-shadow: Boxen mit 3D-Effekt durch Schattenwurf

› border-radius:

Die Border-Radius-Eigenschaft erlaubt es,
runde Ecken dem Element hinzuzufügen

› border-radius: 25px;

› box-shadow:



› box-shadow: 10px 10px 5px #888888;

- › background-origin/background-clip:
 - › Berechnung für die Position eines Box-Hintergrundes
- › background-size:
 - › Hintergrundbilder mit bestimmter Größe
 - › entweder in Pixeln oder Prozenten
- › multiple backgrounds:
 - › Verwendung mehrerer Hintergrundbilder

› HSL colors

- › Nutzung von HSL-Farben → Hue, Saturation, Lightness (Farbton, Sättigung, Helligkeit)

› HSLA/RGBA colors

- › So wie HSL/RGB, aber mit A für Alpha (Durchsichtigkeitsfaktor)

› Opacity

- › Undurchsichtigkeit, Unterschied von RGBA ist die Vererbung

› text-shadow

- › Texte können Schatten bekommen

› word-wrap

- › Zeilenumbruch, falls ein Wort innerhalb der Box zu lang ist

Veränderungen und Neuerungen in CSS3

Texteffekte

› text-shadow:

Text-shadow effect

› text-shadow: 2px 2px #FF0000;

› word-wrap:

This paragraph contains a very long word:
thisisaveryveryveryveryveryverylongword. The long word will break and wrap to the next line.

This paragraph contains a very long word:
thisisaveryveryveryveryveryverylongword.
The long word will break and wrap to the next line.

› word-wrap: break-word;

Veränderungen und Neuerungen in CSS3

Mehrspaltiges Layout

Note: Internet Explorer 9, and earlier versions, does not support the column-count property.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat voluptat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo	consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum	soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum claritatem. Investigationes demonstraverunt lectores legere me lius quod ii legunt saepius.
---	---	--

- Für Firefox:
→ -moz-column-count: 3;
- Für Chrome und Safari:
→ -webkit-column-count: 3;

- › column-count: 3;
- › column-width: 100px;

› Media Queries

- › Erweiterung von Media Types mit Eigenschaften
- › Anpassung für Medien wie z.B. Handhelds

› Eigenschaften für Media Queries:

- **width**
- **height**
- **device-width**
- **device-height**
- **orientation**
- **aspect-ratio**
- **device-aspect-ratio**
- **color**
- **color-index**
- **monochrome**
- **resolution**
- **scan**
- **grid**

- › all
 - › Gilt für alle Medien
- › braille
 - › Für Medien mit Blindenschriftfeedback
- › embossed
 - › Für Blindenschriftdrucker
- › handheld
 - › Für Handheld-Geräte
- › print
 - › Für Druckversionen

- › projection
 - › Für Beamer
- › screen
 - › Für Monitore
- › speech
 - › Für Sprachcomputer
- › tv
 - › Für Fernseher

@media handheld and (min-width 80px) and (max-width 400px)
{...}

@media screen and (device-aspect-ratio: 16/9)
{...}

Ausblick in die Zukunft

„...the technology is not versioned and instead we just have a living document that defines the technology as it evolves.“

- Ian Hickson (Einer der Autoren der HTML-Spezifikation)

„There is no such thing as CSS4.“

- Tab Atkins Jr. (Mitglied der CSS Working Group)



HTML 5 und CSS3

Vielen Dank für eure Aufmerksamkeit!

Literaturverzeichnis

- › Peter Kröner. HTML5 Webseiten innovativ und zukunftssicher. 2. Auflage, 2011
- › Matthias Kirchner Rodion Sidorencov. Offline webanwendungen unter HTML5, 2011
- › Eric Bidelman. Web worker-grundlagen, 2010
- › Brian P. Hogan. HTML5 & CSS3 Webentwicklung mit den Standards von morgen. 1. Auflage, 2011
- › Michael Jendryschik. Einführung in XHTML, CSS und Web-Design:Standardkonforme, moderne und barrierefreie Websites erstellen, 2008
- › http://www.w3schools.com/cssref/css3_pr_border-radius.asp, Seitenbesuch am 05.06.2014
- › <http://www.mediaevent.de/css/css-selektor-attributselektor.html>, Hässler April 2013
- › <http://www.css3.info/preview/>, Seitenbesuch am 06.06.2014