

Hochschule Darmstadt

Fachbereich Informatik

Entwicklung webbasierter Anwendungen



h_da

HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

fbi

FACHBEREICH INFORMATIK

Hochschule Darmstadt

Fachbereich Informatik

Evaluation



h_da

HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

fbi

FACHBEREICH INFORMATIK

Multiple Choice

Frage	Durchschnitt	Teilnehmer
Ich kann der Lehrveranstaltung problemlos folgen.	8,54	35
Die Vorlesung ist klar strukturiert.	8,82	34
Die Vorlesung ist mit ausreichend praktischen Beispielen bestückt.	8,78	35
Die Dozenten gehen ausreichend auf Fragen ein.	9,82	34
Die Anforderungen der Vorlesung sind angemessen.	8,09	33
Die Anforderungen des Praktikums sind angemessen.	8,07	28

Teilnehmer: 35

Bewertungsschema: 1 - 10

Multiple Choice

Frage	Durchschnitt	Teilnehmer
Die Bewertung des Praktikums ist angemessen.	9,08	25
Ich hatte im Bereich EWA vor der Veranstaltung bereits ausreichend Vorkenntnisse	4,35	35
Die Dozenten vermitteln Begeisterung für das Fach	8,52	35
Die Dozenten sind fachlich kompetent und gut vorbereitet	9,31	35
Das Skript ist gut hinsichtlich Umfang, Aktualität, Inhalt, Gliederung	8,4	32

Teilnehmer: 35

Bewertungsschema: 1 = stimme nicht zu - 10 = stimme zu

“Gute Live-Beispiele”

“Mehr Beispiele”

“Beispiele hochladen”

“Mehr Beispiele im Skript”

“Zu viel Wiederholung”

“Verwirrender Wechsel zwischen HTML5 / XHTML: Was darf man verwenden?”

→ *Beides! Achten Sie bei HTML5 auf eine Dokumentenstruktur gemäß XHTML (Sauberkeit), d.h. schließen Sie Ihre Tags.*

Hochschule Darmstadt

Fachbereich Informatik

2.2 CSS



h_da

HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

fbi

FACHBEREICH INFORMATIK

Beispiel Fachbereichsseite Informatik Hochschule Darmstadt

8 Entwicklung webbasierter Anwendungen, WS2015/2016, Christopher Dörge / David Müller

2.2 CSS

Ohne CSS

Fachbereich Informatik



- [Deutsch](#)
- [English](#)
- [Login](#)

[Passwort vergessen?](#)

- [Backend](#)
- [Suche](#)
- [Sitemap](#)
- [RSS](#)

Fachbereich Informatik

- [fbi](#)
- [Organisation](#)
- [Studieninteressierte](#)
- [Personen](#)
- [Studium](#)
- [Labore](#)
- [Verschiedenes](#)
- [International](#)
- [Forschung & Partner](#)
- [OBS](#)
- [Intern](#)

[Suche](#) [Login](#) [Home](#)

- [fbi](#)
- [Organisation](#)
- [Studieninteressierte](#)
- [Personen](#)
- [Studium](#)
- [Labore](#)
- [Verschiedenes](#)
- [International](#)
- [Forschung & Partner](#)
- [OBS](#)
- [Intern](#)



Nachrichten des Fachbereichs Informatik

[Interesse an Korea?](#)

Liebe Studierende,

sind Sie interessiert an einem Studium in Korea? Dann haben Sie am

Montag, 30.11. von 16.00-17.00 in Raum D14/0.13

die einmalige Möglichkeit die verantwortlichen Ansprechpartner unserer drei Partnerhochschulen in Hanyang, Soongsil und Kyungpook kennenzulernen. Alle Verantwortlichen werden ihre eigenen Hochschule kurz vorstellen, und anschließend besteht die Möglichkeit auf offene Fragen einzugehen.

Hochschule Darmstadt

Fachbereich Informatik

2.2.1 CSS Grundlagen



h_da

HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

fbi

FACHBEREICH INFORMATIK

Cascading Style Sheets

CSS Version	
1.0	1996
2.0	März 1998
2.1	Juni 2011
3.0	Draft 2012

- Definition physischer Attributsätze für
 - ⇒ vordefinierte logische Formate (Überschrift 2, Aufzählung, ...)
 - ⇒ selbstdefinierte Format-Klassen
 - ⇒ einzelne (Text-)Blöcke
- vielfältige physische Attribute
 - ⇒ Schriftart, -größe, -stil, Zeichen- und Zeilenabstand, Einrückung
 - ⇒ Text- und Hintergrundfarbe, Rahmen
- Animationen
- freie Platzierung und Überlappung von Objekten
 - ⇒ vgl. Autorensysteme

Nutzung der CSS

- exakte Bestimmung des Erscheinungsbilds
 - ⇒ die variable Bildschirmauflösung bleibt ein Problem
- Optimierung für verschiedene Ausgabemedien
 - ⇒ Bildschirm, Drucker, TV, Palmtop, Screenreader...
- Grundlage für Barrierefreies Webdesign
 - ⇒ Blindenschriftfähige Ausgabegeräte
 - ⇒ Blindenschriftfähige Drucker

Potentiale der CSS

- verbesserte Exportierbarkeit aus anderen Tools
 - ⇒ Erstellung von Webseiten mit Grafiktools
 - ⇒ kein Fachwissen erforderlich zur Erstellung von Drafts
- vermehrte Nutzung von Frameworks
 - ⇒ Bootstrap
 - ⇒ Foundation

Arbeitsteilung mit CSS

- saubere Trennung zwischen Inhalt und Form
 - ⇒ Inhalt logisch formatiert in HTML
 - ⇒ Physisches Format und Fein-Layout separat in CSS
- Arbeitsteilung Programmierer / Designer wird möglich
 - ⇒ einheitliche Layouts für große Projekte
- Corporate Design kann übernommen werden
 - ⇒ hierarchischer Aufbau (Kaskadierung)
 - ⇒ Übernahme und Abwandlung einer Designvorgabe



Einbindung von CSS in HTML (1)

■ "extern" in eigener CSS-Datei

- ⇒ kann von mehreren HTML-Dateien genutzt werden

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

Normalfall

■ "eingebettet" im HTML-Code

- ⇒ gilt nur für diese eine HTML-Datei

```
<style>
```

```
/* ... Style-Sheet-Definitionen ... */
```

```
</style>
```

gehört in den
HTML-`<head>`

CSS-Kommentar

Einbindung von CSS in HTML (2)

■ "inline" in jedem HTML-Tag

- ⇒ gilt nur für dieses eine Objekt

```
<p style="color:red; font-size:36pt;">  
großer roter Text</p>
```

- ⇒ HTML Inline-Tag `` zur Markierung eines Teilbereich eines Objekts

primärer Zweck

```
<p>Normaler Textabsatz mit  
  <span style="font-style:italic; color:red;">  
    rot-kursivem Text  
  </span> und wieder normal  
</p>
```


Unterstützung verschiedener Ausgabemedien

- verschiedene Bereiche innerhalb eines CSS

```
@media screen {  
    /* Style-Sheet-Definitionen für den Bildschirm */  
}  
@media print {  
    /* Style-Sheet-Definitionen zum Drucken */  
}
```

- verschiedene CSS-Dateien in HTML einbinden

```
<link rel="stylesheet" media="screen"  
      href="Bildschirm.css" />  
<link rel="stylesheet" media="print"  
      href="Drucker.css" />  
<link rel="stylesheet" media="speech"  
      href="Screenreader.css" />
```

Hochschule Darmstadt

Fachbereich Informatik

2.2.2 CSS - Formate definieren



h_da

HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

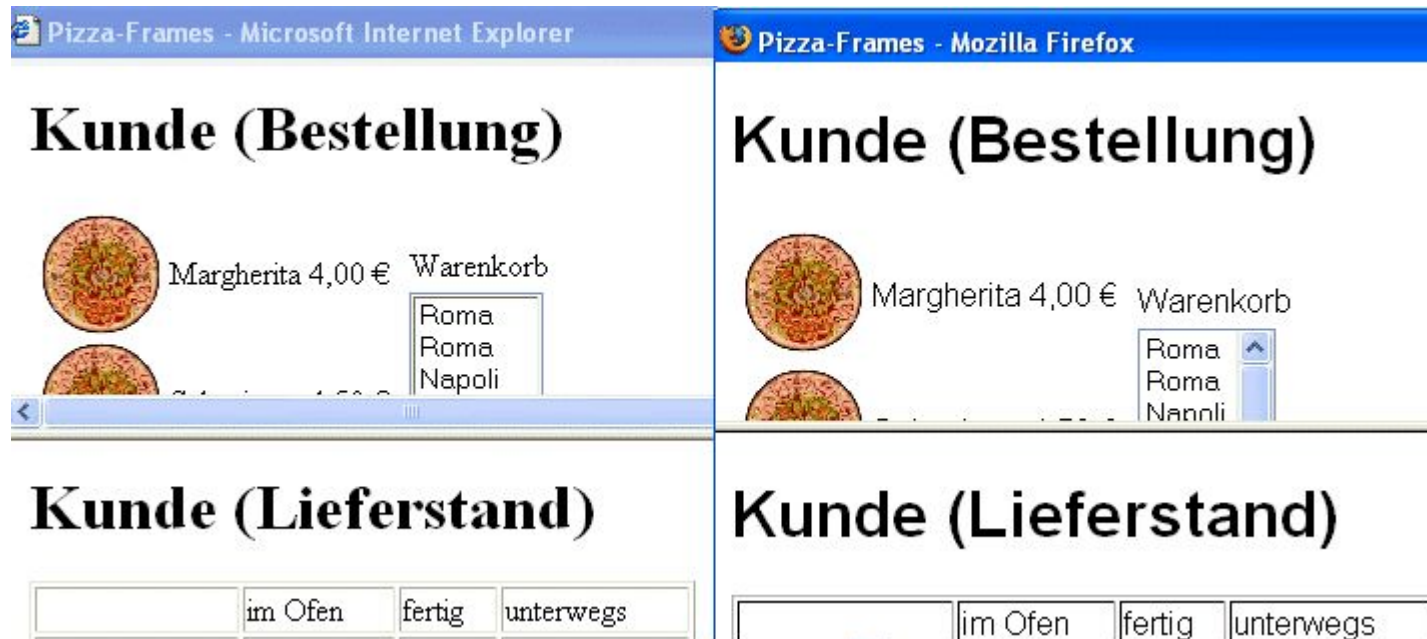
fbi

FACHBEREICH INFORMATIK

Browser-Default-Formatierung im Vergleich - ohne CSS

■ Browser haben unterschiedliche Default-Stile

⇒ hier Internet Explorer 6 und Firefox 2



Um ein definiertes Layout zu erhalten,
muss man die Standard-Formate selbst definieren!

Standard-Formate modifizieren

■ Definition in CSS

vorzugsweise für Ausgestaltung
logischer Formate

```
h3 { text-align: center; color:#33FF00; }  
p  { border: 1px solid black;  
    font-family:Arial, Helvetica; }  
*   { color:green; } /* Universalselektor gilt für alle Tags*/  
ul { list-style:none; } /* verbirgt die Aufzählungspunkte */
```

■ Anwendung in HTML

kein Attribut
in HTML

```
<h3>Überschrift 3. Ebene</h3>  
<p>einfacher Fließtext in einem Absatz</p>
```

⇒ ohne CSS zeigt der Browser die "schlichte" Version

Standard-Formate kontextabhängig

■ Definition in CSS

```
h1 { color:red; }  
h1 i { color:blue; font-weight:normal; }
```

d.h. Italic geschachtelt in Header 1

dieses Format gilt nur dort

■ Anwendung in HTML

```
<h1>Eine Überschrift mit <i>Style-Sheets</i></h1>  
<p>Ein Fließtext mit <i>Style-Sheets</i></p>
```

nicht hier

Eine Überschrift mit *Style-Sheets*

Ein Fließtext mit *Style-Sheets*

Eigene Format-Klassen

■ Definition in CSS

- ⇒ Unterklassen für Standard Formate

```
p.Hinweis { color:red; }  
p.Fussnote { color:black; }
```

- ⇒ allgemein verwendbar

```
.Warnung { color:#DC0081 }  
.Zitat { color:#00DFCA }
```

aber keine Klasse mit
Ableitung etc. wie in OO

■ Anwendung in HTML

```
<p class="Hinweis">beachten Sie bitte</p>  
<p class="Fussnote">das nur am Rande</p>  
<p class="Warnung">Achtung! Aufpassen!</p>  
<blockquote class="Zitat">des Pudels Kern  
</blockquote>
```

HTML Attribut `class` stellt den Bezug her

Individuelle Objekt Formate

■ Definition in CSS

```
#Block1 { font-weight:bold; font-style:italic; }  
#Hotw3 { text-decoration:underline; }
```

■ Anwendung in HTML

⇒ jedes Format und jede **id** nur einmal !

Eindeutige Block-IDs
kann man auch für
JavaScript brauchen

```
<p id="Block1">Extra-Formatierung</p>  
<p>Einfacher Text mit <em id="Hotw3">Hotword</em></p>
```

"Hotw3" ergänzt das Format von ; im Konfliktfall mit Vorrang

Pseudo-Formate

- Sonderfall:
 - ⇒ definieren Eigenschaften, die keine Attribute von HTML-Blöcken sind
- Darstellung von Hyperlinks - Festlegung in CSS
- ```
a:link { color:blue;} /* normaler Link */
a:visited { color:green;} /* bereits besucht */
a:active { color:red;} /* gerade angeklickt */
a:hover { color:yellow;} /* unter dem Mauszeiger */
a:focus { color:black;} /* mit Tastatur angewählt */
```
- ```
p:first-line { font-weight:bold; }
p:first-letter { font-size:36pt; color:red;}
```

Man kann nur Brücken schlagen zwischen Ufern die man auseinanderhält. Denn wo es keine Gräben gibt, da gibt es auch keine Unterschiede, und wo es keine Unterschiede gibt, da ist kein Leben.

Weitere Selektoren (eine kleine Auswahl)

■ Flexible Möglichkeit, um Tags auszuwählen:

- ⇒ `[target]` Wählt alle Tags mit einem Attribut target
- ⇒ `[target=xyz]` Wählt alle Tags mit einem Attribut target, das den Wert xyz hat

■ Auswahl mit Parametern

- ⇒ `p:nth-child(2)` Wählt alle p-Tags, die das 2-te Kind von irgendeinem Tag sind
- ⇒ `p:nth-child(3n+1)` Wählt alle p-Tags, die das 1-te, 4-te, 7-te,.... Kind von irgendeinem Tag sind
- ⇒ Anwendung für "gestreifte" Tabellen (`even` und `odd` sind vordefiniert):
 - `tr:nth-child(even) { background-color: LightGrey; }`
 - `tr:nth-child(odd) { background-color: white; }`

Es gibt noch viele weitere Selektoren!

Hochschule Darmstadt

Fachbereich Informatik

2.2.3 CSS - Attribute



h_da

HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

fbi

FACHBEREICH INFORMATIK

Farben und Hintergrundbilder

Farbschema entwerfen mit
<http://kuler.adobe.com>

■ Farbattribute

`background-color` Hintergrundfarbe

`color` Textfarbe

`border-color` Rahmenfarbe

`text-shadow` schattierter Text

```
background-color: white;
```

■ Notationen für Farbwerte

`rgb(255,140,0)` Farbanteile für rot, grün, blau im Bereich 0..255

`rgb(100%,55%,0%)` Farbanteile im Bereich 0%..100%

`#FF8C00` Farbanteile hexadezimal

`Darkorange` diverse Farben mit Namen

■ Hintergrundbild

⇒ nicht nur für gesamte Seite, sondern auch für einzelne Blöcke

`background-image:url(bild.gif)`

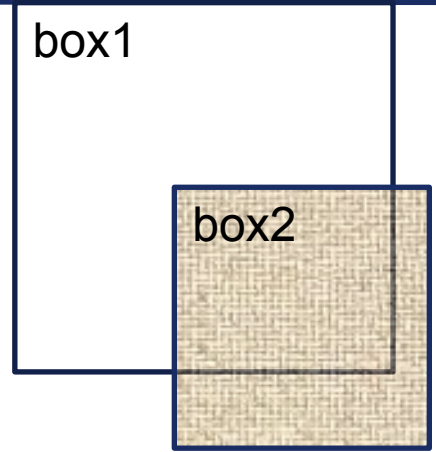
Transparenz

■ Durchsichtige Elemente

- ⇒ `opacity: 0.5;`
- ⇒ Einstellung über eine Zahl zwischen 0 (voll transparent) und 1 (volle Deckkraft)
- ⇒ Macht das gesamte Element mit Rahmen, Hintergrund etc. transparent
- ⇒ Ein- und Ausblendeffekte (mit Javascript)

■ Transparent als „Farbe“

- ⇒ `background-color: transparent;`
- ⇒ Macht nur das jeweilige Element transparent



Schrift

- **font-family:**
 - ⇒ Arial, Helvetica, "Times New Roman"
 - ⇒ serif, sans-serif, cursive, fantasy, monospace
- **font-style:**
 - ⇒ italic, normal
- **font-size:**
 - ⇒ 12pt, 35px, 3em, 1.5cm, large
- **font-weight:**
 - ⇒ bold, bolder, lighter, 100 .. 900
- **font:**
 - ⇒ kompakte Kombination o.g. Attributwerte

Die Bedeutung der verschiedenen Maßeinheiten kommt im nächsten Abschnitt!

Aussen- und Innenabstand

die Standardwerte sind browserabhängig, deshalb vollständig spezifizieren!

- `margin`, `margin-top`, `margin-bottom`, `margin-left`, `margin-right` Aussenabstand in Längenmaß
- `padding`, `padding-top`, `padding-bottom`, `padding-left`, `padding-right` Innenabstand in Längenmaß
- Achtung: `width` und `height` beziehen sich auf den Inhalt !

`margin`: Abstand zur Nachbarbox (transparent)

`border`: Rand (standardmäßig nicht vorhanden)

`padding`: Innenabstand (background-color des Elements)

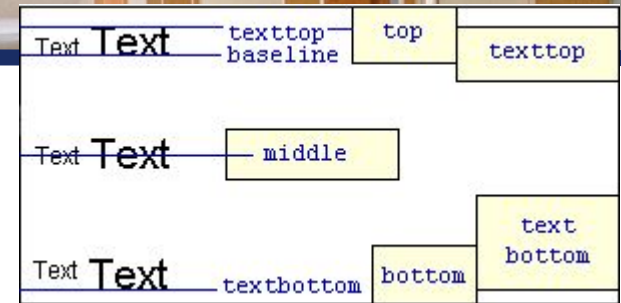
Inhalt des HTML-Elements



Ausrichtung und Rand

■ Ausrichtung

- ⇒ `line-height` Zeilenhöhe in Längenmaß
- ⇒ `text-indent` Texteinrückung in Längenmaß
- ⇒ `text-align: left, center, right, justify` (Blocksatz)
- ⇒ `vertical-align: top, middle, bottom, text-top, text-bottom`



Quelle: SelfHTML

■ Rand

- ⇒ `border[-top, -left, -right, -bottom]-width`
(z.B. `border-left-width`, `border-width`)
- ⇒ `border[-top, -left, -right, -bottom]-style:`
hidden, dotted, dashed, solid, double, groove, ridge, inset, outset

■ Abgerundete Ecken

- ⇒ `border[-top, -bottom][-left, -right]-radius: x radius y radius`
- ⇒ z.B. `border-top-left-radius: 3em 2em` oder `border-radius: 5%`

Erscheinungsbild einer Tabelle

Spalte A	Spalte B	Spalte C
Even	Even	Even
Odd	Odd	Odd
Even	Even	Even

- Tabellen werden mit den üblichen Elementen formatiert:

- ⇒ `width, height, padding, border, margin,`
- ⇒ `text-align, vertical-align`

- Üblicherweise werden die Linien angeschaltet

- ⇒ `table, th, td { border: 1px solid black; }`
- ⇒ dann hat aber jede Zelle einen Rahmen, d.h. die Linien sind doppelt
Lösung: `border-collapse: collapse;`

- Beispiel: Gestreifte Tabelle

```
tr:nth-child(even){ background-color: LightGrey; }
tr:nth-child(odd) { background-color: white; }
th {background-color:black; color:white; }
table, th, td { border: 1px solid black;
                  border-collapse: collapse;}
```


Zeigen und Verbergen

Auf- und Zuklappen
von Unterpunkten
im Inhaltsverzeichnis
mit JavaScript

- Anzeige(-art) bzw. Nichtanzeige ohne Platzhalter
(folgende Blöcke verschieben sich)

display:

inline Element wird im laufenden Textfluss angezeigt. Der Text "fließt" in Lücken, welche die anderen Elemente bieten

block Rechteckig begrenztes Element steht alleine in einer Zeile

inline-block Rechteckig begrenztes Element, das als Block im Textfluss bleibt (siehe folgendes Beispiel)

none Element wird nicht angezeigt, folgende Blöcke verschieben sich

- Anzeige bzw. Nichtanzeige mit Platzhalter

visibility:

visible Element wird angezeigt

hidden Element wird versteckt (folgende Blöcke bleiben stehen)

2.2.3 CSS - Attribute

Beispiel: HTML mit leerer CSS-Datei

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <meta charset="UTF-8" />
    <title>CSS-Vorbereitung</title>
  </head>
  <body>
    <header><h1>Kopfzeile</h1></header>
    <nav><ul>
      <li>Menu1</li> <li>Menu2</li>
    </ul></nav>
    <section>
      <article>Inhalt1</article >
      <article>Inhalt2</article >
    </section>
    <footer>Fußzeile</footer>
  </body>
</html>
```

Kopfzeile

- Menu1
- Menu2

Inhalt1

Inhalt2

Fußzeile

Beispiel: Gleiche HTML-Datei mit einfachem CSS

Achtung!
Nicht das Semikolon oder die Klammern vergessen. Sonst funktioniert es nicht!

```
* { padding:0pt; margin:0pt; } /* keine Default Abstände */
body {color:black; background-color: WhiteSmoke;
      font:1em Verdana;}
footer, header {clear: both; text-align:center;
                 color: white; background-color: grey;}
article {display: inline-block; width: 15em;
          border: 1px solid black; margin:5px;}
nav {display: block; margin: 5px; border: 2px solid grey;
     text-align:center; float:left;}
nav li {font-size: 1.5ex; margin: 1px;
        background-color: Lavender;}
ul {list-style:none;
    border: 1pt solid white;
    text-align: left;}
```

Maßeinheiten
sind hier nicht
vorbildlich!

