

CSS

Dr. Günter Kolousek

2012-08-18

Übersicht

1 Allgemeines

2 CSS

3 CSS3

Prinzipien

- Cascading Style Sheet
- Position, Größe, Schriften, Farben, Hintergrund, Rahmen,...
- Trennung von Inhalt und Darstellung
- Verschiedene Arten von Stylesheets
 - Browser-Stylesheet
 - Benutzer-Stylesheet
 - Autoren-Stylesheet
- Verkleinern: <http://developer.yahoo.com/yui/compressor/>

Einbinden von CSS

- Kodierung, am besten: UTF-8

- Zentral

- mittels `link`, z.B. `<link href="css/main.css" \ rel="stylesheet" type="text/css">`

- mittels `style` - Element

```
1 <style>
2     h1 { color:red; }
3 </style>
```

- Direkt, mittels `style` - Attribut

CSS Regel

- Selector
 - ungültige Selektoren werden ignoriert!
- Declaration block
 - { Declarations }
 - Declarations: beliebige # an Deklarationen durch ; getrennt
- Declaration
 - Property name : property value [!important]
 - !important überschreibt gleiche Angaben ohne !important
 - Ungültige Namen und Werte werden ignoriert
- Kommentare mittels /* und */

Werte 1

- Zahlen
- absolute Längenangaben
 - Zentimeter, Millimeter, z.B. `1cm` oder `3mm`
 - Inch ($1\text{in}=2.54\text{cm}$), z.B.: `2in`
 - Point ($1\text{pt}=1/72\text{in}$), z.B.: `3pt`
 - Pixel ($1\text{px}=0.75\text{pt}$)
- relative Längenangaben
 - `px`: CSS Pixel
 - abhängig von Pixeldichte
 - d.h. 1 CSS `px` muss nicht nur 1 (Screen-)Pixel sein (z.B. Drucker)
 - Definition über Entfernung Auge über Armlänge (28in) bei 96dpi ($72 \cdot 4/3$, $1\text{px}=0.26\text{mm}$): 0.0213°
 - `ex`: x-Höhe des Fonts, z.B.: `3ex`
 - `em`: `font-size`, z.B. `2em`

Werte 2

- Prozentwerte, z.B.: 100%
- URLs, z.B.:
`url(http://www.htlwrn.ac.at/logo.png)`
- Farben
 - Namen, z.B.: `black`
 - RGB Werte, z.B.: `#FF0000`, `rgb(255, 0, 0)` ,
`rgb(100%, 0%, 0%)`
 - RGBA Werte (Alpha-Kanal \leadsto Transparenz), z.B.
`rgba(255, 0, 0, 0.7)`
 - 0 ... unsichtbar (transparent)
 - 1 ... vollständig sichtbar
- Strings, z.B.: `"News"`, `'News'`
- `inherit` ... vom Elternelement erzwingen

Selektoren - 1

■ Universal-Selector

- `*.warning { background-color:orange; }`
- `* { color:black; }`

■ Type-Selector

- `h1 { font-size:36px; }`

■ Class-Selector

- `class="important"`
- `.important { color:red; }`

■ Id-Selector

- `id="nav", z.B. #nav { font-weight:bold; }`

Selektoren - 2

■ Attribut-Selectors

- `audio[controls]` ... wenn ein `controls` Attribut vorhanden
- `input[type="checkbox"]` ... exakt gleich
- `a[rel~="copyright"]` ... in "Blank-Liste"
- `*[lang="en"]` ... gleich oder beginnt mit en-

■ Pseudo-Klassen

- `a:link` ... noch nicht besucht
- `a:external:visited` ... schon besucht
- `a:hover` ... Maus "darüber" (nach `:link` bzw. `:visited`)
- `a:active` ... z.B. angeklickt und noch nicht losgelassen
- `a:focus` ... hat Fokus (z.B. mittels TAB), d.h. empfängt Eingaben
- `html:lang(de)` ... `lang` Attribut mit Sprache
- `div > p:first-child { text-indent: 0 }`

Selektoren - 3

■ Pseudo-Elemente

■ `:first-letter`

- `p:first-letter { font-size:2em; }` ... **erstes Zeichen in Absatz**

■ `:first-line`

- `p:first-line { color:brown }` ... **erste (angezeigte) Zeile in Absatz**

■ `:before ...` **einfügen** (detto: `:after`)

- `p.special:before {content: "Special! "}`
- siehe <http://www.w3.org/TR/CSS21/generate.html>

Selektoren - 4

- Grouping

- `h1, h2 { background:orange; color:black }`

- Descendant (beliebiger Nachfolger)

- `section p { color:grey; }`

- Child (direkter Nachfolger)

- `section > header { font-size:36px; }`

- Adjacent sibling (direkt nachkommend; gleiche Ebene)

- `h1 + p { margin-top:0px; }`

@-Regel

■ 2. Art von Regeln

- `@import "lists.css"`

- `@media print { @import "print.css"; body { font-size: 10pt } }`

- `@charset "utf-8"`

Box-Modell

- **Content:** `width, height`
- **Innenabstand:** `padding, padding-top | right | bottom | left`
- **Rahmen:**
 - `border,`
 - `border-top,...`
 - `border-width, border-top-width,...`
 - `border-color, border-top-color,...`
 - `border-style, border-top-style,...`
- **Außenabstand:** `margin...`
- **CSS3:** `box-sizing` (in Firefox mit `-moz-`)
 - `content-box:` **default, wie oben**
 - `border-box:` `width` und `height` **enthalten Rahmen und Innenabstand**

Position und Anzeige

- Typ der Positionierung: `position`
 - `static`: Default; normale Positionierung im Fluss; kein top,...
 - `absolute`: relativ zum umgebenden Element (wenn nicht `static`) oder zum Dokument; nicht im Fluss!
 - `fixed`: relativ zum Dokument; scrollt nicht mit! nicht im Fluss!
 - `relative`: ursprünglich gemäß Fluss; dann relativ zu dieser Position; ursprünglicher Platz bleibt frei
- Ort: `top`, `left` (seltener: `bottom`, `right`)
- `visibility`: `visible`, `hidden`
- `display`: hauptsächlich `inline`, `block`

Fonts - 1

- `font-family`
 - 3 wichtige Arten
 - **Serifenlose Schrift:** Arial, Helvetica, Verdana
 - **Serifenschrift:** Times, Palatino
 - **mit fester Breite:** Courier new, Andale Mono, Inconsolata
 - **Generisch:** serif, sans-serif, monospace, cursive (wie Schreibschrift), fantasy (ungewöhnlich)
 - **Beispiele:**
 - `font-family:Verdana, Geneva, sans-serif;`
 - `font-family:"Times New Roman", Times, serif;`
 - `font-family:"Courier New", Courier, monospace;`
- `font-style:normal, italic (kursiv), oblique (schräggestellt)`
- `font-variant:normal, small-caps`

Fonts - 2

- `font-size:`
 - numerischer Wert
 - auch mit Zeilenhöhe: `font-size: 1.2em/1.5em`
 - Symbol
 - `xx-small, x-small, small, medium, normal, large, x-large, xx-large`
 - `smaller, larger`
- `font-weight:`
 - Symbol
 - `normal, bold, bolder, lighter`
 - numerisch: 100(extra dünn), 200, 300, 400(normal), 500(medium), 600, 700(fett), 800, 900(extra fett)
- `font`

Ablauf der Kaskade

- 1 Ermittlung aller selektierten Elemente (→ Stylesheets)
- 2 Sortierung nach Ursprung und Priorität
 - 1 Benutzer-Stylesheet mit `!important`
 - 2 Autoren-Stylesheet mit `!important`
 - 3 Autoren-Stylesheet
 - 4 Benutzer-Stylesheet
 - 5 Browser-Stylesheet
- 3 Spezifität: Gewichtung der Selektoren
 - 1 `style` Attribut
 - 2 Anzahl der `id` Attribute
 - 3 Anzahl der Klassen und Pseudoklassen
 - 4 Anzahl der Elemente und Pseudoelemente

→ Zählen, absteigend sortieren, zuerst 1, dann 2,...
- 4 Reihenfolge des Vorkommens: später → überschreiben

CSS3

- erweitert CSS 2.1
- noch nicht fertig
- modularisiert
 - wichtig derzeit und fertig (d.h. "Recommandation"):
 - Media Queries Level 3
 - CSS Colors Level 3
 - Selectors Level 3
 - interessant und noch **nicht** fertig
 - CSS Fonts Level 3
 - CSS Backgrounds and Borders Level 3
 - CSS Transforms
 - CSS Transitions
 - CSS Animations
 - CSS Flexible Box Layout (aka flexbox)
 - ... (annähernd 50 Spezifikationen!)
- ~> <http://meiert.com/en/indices/css-properties/>

Media Queries

- Ausdrücke mit Medientypen und Medieneigenschaften
- Medientypen aus HTML4 sind:
 - wichtig: `screen`, `print`, `handheld` und `all`
 - andere: `aural` (synthetische Sprachausgabe), `braille`, `projection`, `tty`, `tv`
 - CSS2: `deprecates aural`, fügt `embossed` (Braille-Drucker) und `speech` hinzu
- Medieneigenschaften, z.B.:
 - `(min-width)`, `(width)`, `(max-width)`
 - `(height)`, `(orientation)`, `(color)`,...
- Beispiele
 - `<link media="screen and (color)" rel="stylesheet" href="cscr.css">`
 - `<link media="screen and (min-width:500px) and (orientation: portrait)"...`

Farben

- **RGBA Werte:** RGB + Alpha-Kanal (\leadsto Transparenz)
 - 0 ... unsichtbar (transparent)
 - 1 ... vollständig sichtbar (opaque)
 - z.B. `rgba(255, 0, 0, 0.7)`
- **HSL Werte:** Hue (Farbton), Saturation (Sättigung), Lightness
 - H ... Gradangabe am Farbenkreis: 0=360=rot
 - S ... 0%=Grauschatten, 100%=volle Sättigung
 - L ... 0%=schwarz, 100%=weiß, 50%="normal"
 - z.B.: `hsl(0, 100%, 50%)`
- **HSLA Werte:** HSL + Alpha (\leadsto `hsla`)
- **Attribut `opacity`** (Alphakanal), z.B. `footer { opacity:0.7; }`

CSS3-Selektoren - 1

■ Pseudo-Klassen-Selektoren

- `:enabled`, `:disabled` z.B. bei `input` Elementen
- `:checked`, `:indeterminate`: **Checkbox**, **Radiobox**
- `:last-child`
- `:nth-child()`, `:nth-last-child()`
 - `p:nth-child(2) ...` p, das 2.Element von Elternelement
 - `tr:nth-child(even) ...` jede gerade Zeile (detto: `odd`)
 - `tr:nth-child(2n) ...` wie vorher (detto: $2n+1$)
 - allgemein (aber siehe Sonderfälle vorher):
`[+|-]<x>n[+|-]<y> ...` x und y natürliche Zahlen
- `:nth-of-type()`, `:nth-last-of-type()` von diesem Typ, sonst wie `:nth-child`
 - `dt:nth-of-type(2n) ...` jedes 2. dt Element

CSS3-Selektoren - 2

■ Pseudo-Klassen-Selektoren

- `:first-of-type` (wie `nth-of-type(1)`),
`:last-of-type`
- `:empty`
- `:not()` nicht (einfacher Selektor):
`button:not([disabled])`

■ Attribut-Selektoren

- `a[href^="http://www.orf.at"]` ... beginnt genau mit
- `a[href$="/register"]` ... endet genau mit
- `a[href*="admin"]` ... beinhaltet

■ General sibling (nachkommend; gleiche Ebene)

- `article ~ aside { ... }`

CSS3 Fonts

- `font-stretch: normal, condensed, expandend,...`
- Einbinden von beliebigen Fonts!
- Unterstützte Formate
 - Chrome, Firefox, Opera, Safari: TrueType (`.ttf`), OpenType (`.otf`), Web Open Font Format (`.woff`)
 - IE 9+: Embedded OpenType (`.eot`), `.woff`
- Beispiel:

```
1 @font-face {  
2     font-family: myFont;  
3     src: url('Sansation_Light.ttf'),  
4         url('Sansation_Light.eot'); /* IE9+ */  
5 }  
6  
7 div { font-family:myFont; }
```

- Browser-übergreifend, im Web verfügbar, auf `font-face` basierend: <http://www.google.com/webfonts/>

CSS3 Backgrounds - 1

■ CSS2.1:

- background-color, b...-image, b...-repeat, b...-attachment, b...-position
 - z.B.: b...-attachment: fixed (default: scroll)
 - z.B.: b...-position: left center (default.: 0% 0%)
 - z.B.: b...-position: 2em top
- bedeckt: Inhalt, Innenabstand, Rahmen
- background-origin: **zu was** background-position relativ ist
 - padding-box (default, wie in CSS2.1)
 - border-box
 - content-box
- background-clip: **gibt Fläche für Hintergrund an; Werte wie** background-origin
- background-image: **Gradients**, z.B.
-webkit-linear-gradient(white, black)

CSS3 Backgrounds - 2

- Textschatten, jetzt mit CSS3, z.B.: `text-shadow: 4px 2px 2px rgba(216, 150, 150, 0.69);` (siehe `box-shadow`)
 - Gradients und Textschatten, am besten mit <http://css3gen.com>
- `background-size`
 - 2 Werte: `width height`
 - 1 Wert: `height`, default: `auto`
 - Wenn Prozentwerte, dann bzgl. Elternelement
 - `contain`: skaliert im Verhältnis so, dass `height` und `width` maximal und Bild in Content passt.
 - `cover`: skaliert im Verhältnis so, dass `height` und `width` minimal und Bild den Content überdeckt.

```
1  div {  
2      background-image: url(bg.gif);  
3      background-size: 50px 50px;  
4      background-repeat: no-repeat;  
5  }
```

CSS3 Borders

```
1  div {  
2      border: 2px solid;  
3      border-radius: 25px; /* border-radius: top-left tr br bl *  
4      /* Pflicht: h-shadow v-shadow */  
5      /* danach optional: blur spread color inset */  
6      box-shadow: 10px 10px 5px rgba(0,0,0,0.5);  
7  }
```

- border-image **dzt. mit** -moz-, -webkit-, -o- (kein IE)
- blur ... Unschärfe (numerisch, z.B. px)
- spread ... Ausdehnung (numerisch, z.B. px)
- inset ... Einfügung (keyword!), d.h. Schatten nach innen
- ↪ <http://css3gen.com>

CSS3 2D Transforms

■ transform Eigenschaft

- `translate():transform:translate(10px,20px)`

- `rotate():transform:translate(10px)
rotate(45deg);`

- Andere Einheiten: rad, grad (400 je Kreis), turn (1 je Kreis)

- `scale():transform:rotate(45deg) scale(0.5);`
- `=skew()` (schräg stellen)

- `matrix()` ... "Einer für alle!"

■ transform-origin Eigenschaft, gibt Position der "transformierten" Elemente anzugeben

- dzt. jeweils mit: `-moz-`, `-webkit-`, `-o-` und `-ms-`

CSS3 Transitions - 1

- Übergang bei `:hover`
- Eigenschaften
 - `transition-property`, z.B.:
`transition-property:width`
 - `transition-duration`, z.B.:
`transition-duration:2s`
 - Andere Einheit: `ms`
 - `transition-timing-function`, z.B.:
`transition-timing-function:linear` Viele andere:
`ease (default)`, `ease-in`,...
 - `transition-delay`, z.B.: `transition-delay`
 - `transition`: “property” “duration” “timing-function” “delay”
 - dzt. jeweils mit: `-moz-`, `-webkit-`, `-o-` (kein IE!)

CSS3 Transitions - 2

■ Beispiel

```
1  div {
2      width:100px;
3      height:100px;
4      /* -webkit-transition und -webkit-transform,... */
5      transition: width 2s, height 2s, transform 2s;
6  }
7  div:hover {
8      width:200px;
9      height: 200px;
10     transform: rotate(180deg); /* -webkit-transform */
11 }
```
