

CSS

Syntax

Crossmedia  
Publishing

Paged Media Module

HTML

Renderer

Workflows

# Printlayouts mit CSS3

19. April 2013

Pubkon 2013

Kontakt: [gregor.fellenz@publishingx.de](mailto:gregor.fellenz@publishingx.de)

Folien: <http://www.publishingx.de/dokumente>

# Old School Publishing

Ein Content → Ein Layout → Ein Medium

## Seitenbasiertes arbeiten vs. lineare Darstellung



- Wenn mehrere Formate bedient werden müssen, **vervielfacht** sich der Aufwand.
- z.B. **Produktklasse Buch**:
  - Holzbuch
  - E-Book
  - Webseiten/Leseproben
  - Apps?
- Arbeit fällt doppelt an, aber es werden kaum mehr Einheiten verkauft?!?

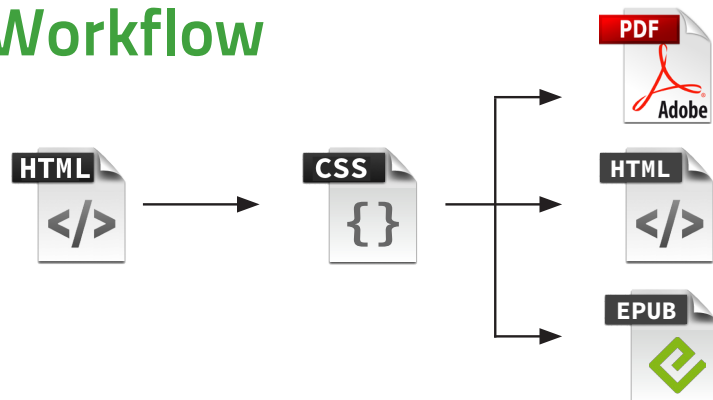
# Crossmedia Publishing

Medienneutrale Workflows für unterschiedliche Ausgabeformate.

## Chancen

- Ein Inhalt für mehrere Medien.
- Wiederverwendbarkeit von Struktur-Informationen.
- Einheitliche Datenspeicherung.
- Trennung von Content und Layout.

## Workflow



## Mögliche Varianten

### ■ XML + Satzsystem

Automatisierte Verarbeitung von XML-Daten in InDesign, 3B2, Miles Oasys ...

- Programmierkenntnisse nötig.

### ■ XSL-FO

Seitenbeschreibungssprache in XML, gerendert wird meist PDF.

- Wird (leider) nicht mehr richtig weiter entwickelt ...
- Datentransformation notwendig.

### ■ HTML/CSS

Sprache des Web, einfacher.

*Nellie McKesson, O'Reilly*: 3 mal so schnell wie XSL-FO

- Vorteil: Viele können CSS und/oder HTML.
- Direkte Weiterverwendung von Webdaten.

# HTML, CSS, WTH?



Quelle: <http://www.collegehumor.com/images/memeposter/the-internet.jpg>



<DIV>Q: HOW DO YOU ANNOY A WEB DEVELOPER?</SPAN>

## Hypertext Markup Language in 2 Minuten

Die Inhaltsdaten/Content von Webseiten wird in HTML abgespeichert.

- HTML Grundgerüst Deklaration, <html>, <head>, <body> macht der Editor...
- Wichtige Elemente
  - Bereiche/Strukturen mit <div>
  - Absätze <p>
  - Textbereiche <span>
  - Bilder <img>  
Ein Bild wird über ein Attribut referenziert (Verknüpfung in InDesign)  


<p>Der Text im Absatz!</p>

← Element



← Attribut

Zum nachlesen empfehle ich: <http://jendryschik.de/wsdev/einfuehrung/xhtml1>

# Cascading Style Sheets in 4 Minuten

## ■ Anatomie einer CSS-Anweisung

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

## ■ Ein paar Eigenschaften:

### ■ z.B. Kasten

```
border: solid;
```

### ■ oder Absatz

```
font-family: sans-serif;
```

### ■ und hervorgehobener Text

```
font-weight: bold;
```

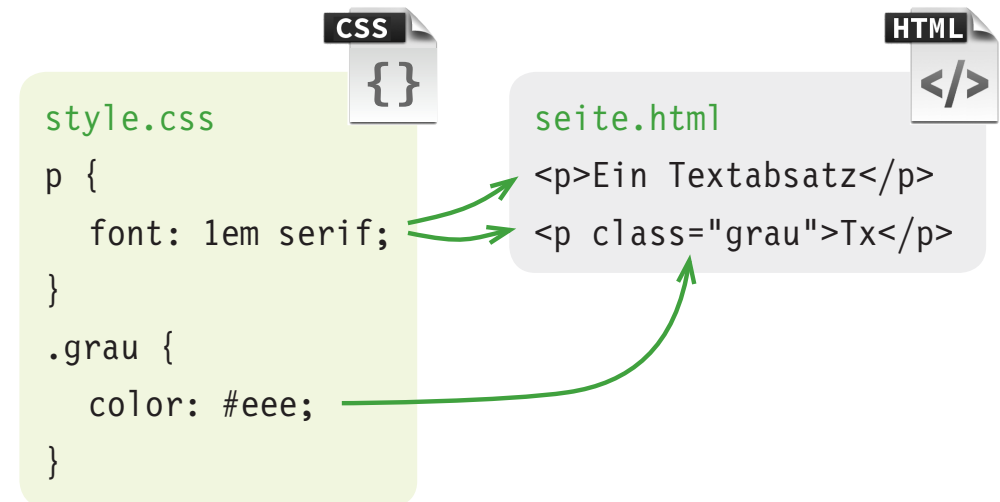
## ■ ... und dann noch das **Cascading**

### ■ vereinfacht gesagt geht es um **Vererbung**

z.B. Angabe der Schriftart in einem `<div>` das mehrere `<p>` Elemente umschließt.

### ■ Quelle großer Verwirrung :-)

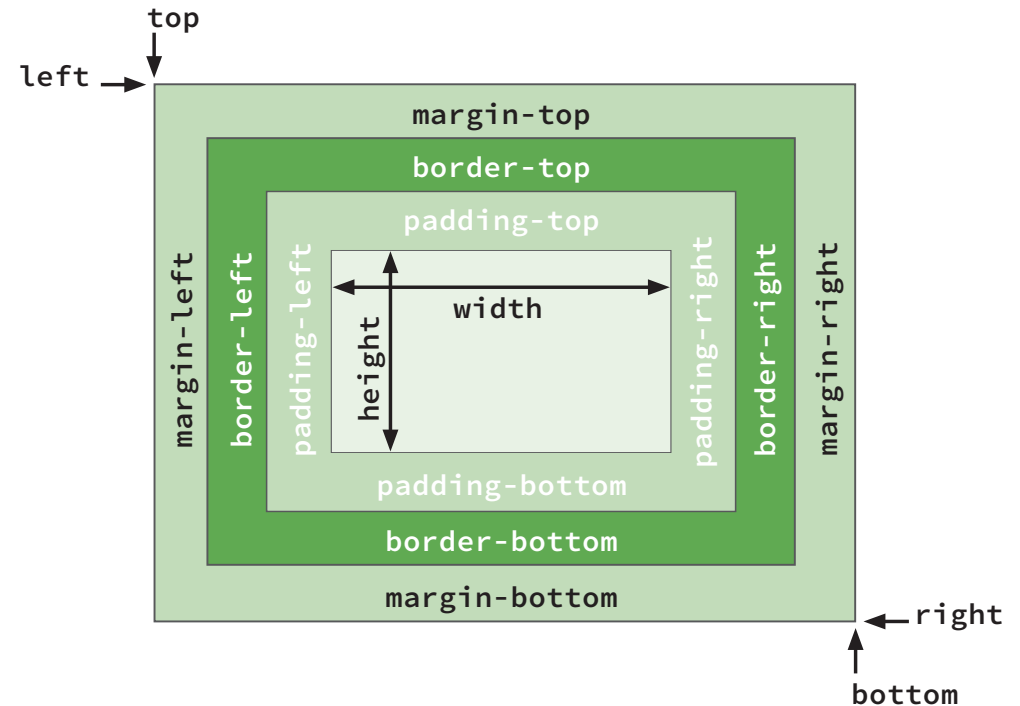
<http://i.imgur.com/Q3cUg29.gif>





## Boxmodell

- ist nicht ganz unwichtig...
- Die wichtigsten Eigenschaften sind `margin` und `padding` (`border-width`)
- Wertevergabe im Uhrzeigersinn:  
`margin: oben rechts unten links;`



## Weiterlesen

<http://jendryschik.de/wsdev/einfuehrung/>

## oder weiterklicken

<http://www.codecademy.com/de/courses/html-one-o-one/0/1>

# So sieht das aus ...

Demo im Browser ...

Demo mit PDF ...

# Paged Media Module

Neben den „normalen“ CSS für Webseiten gibt es das sogenannte **Paged Media Module**, das sich speziell mit den Bedürfnissen von gedruckten Medien beschäftigt.

- Die Spezifikation findet man hier: <http://www.w3.org/TR/css3-page/>

## Eine Seite

- **Zentrale Anweisung:** `@page`

```
@page {  
    size: breite hoehe;  
}
```

- margin Möglich
- Ebenfalls Boxmodell

## Pseudo Klassen für ...

- ... Erste Seite

```
@page :first {  
    padding: 0;  
}
```

- ... Doppelseiten/Druckbögen:

```
@page :left {  
    padding-left: 0.5cm;  
}
```

```
@page :right {  
    padding-right: 0.5cm;  
}
```

## Musterseiten

- Zuweisung eines Bereichs über Eigenschaft page

```
div.section {  
    page: teil;  
}
```

- Definition der Seite mit @page

```
@page teil {  
    margin: 3.5cm 2.5cm 3cm 2.5cm;  
}
```

## Seitenbereiche/ Header und Footer

- Es sind insgesamt 17 Bereiche definiert.

```
@page teil:right {  
    @top-center {  
        content: counter(page) ;  
    }  
}
```

- Gute Übersicht <http://dev.w3.org/csswg/css-page/#margin-boxes>

## Mehrspaltige Layouts

- Mehrspaltige Bereiche sind möglich

```
div.zweiSpalten {  
    column-count: 2;  
    column-gap: 4mm;  
}
```

top-left-corner	top-left	top-center	top-right	top-right-corner
left-top	main page area			right-top
left-middle				right-middle
left-bottom				right-bottom
bottom-left-corner	bottom-left	bottom-center	bottom-right	bottom-right-corner

## Seitenzahlen/Counter

- Seitenzahl

`content: counter(page);`

- Kapitelnummern

Ausgeben:

```
h1:before {  
    counter-increment: KapEbene1;  
    content: counter(KapEbene1);  
}
```

Initialisieren

```
body {  
    counter-reset: KapEbene1;  
}
```

- Mehrere Counter möglich (Bilder, Tabellen, etc.)

- Lebende Kolumnen mit `content: string(kol)` und `string-set: kol content()`



## Cross References

### Aufbau von dynamischen Querverweisen

- Seitenverweis `target-counter()`  

```
a.xrefCounter:after {  
  content: " (Seite " target-counter(attr(href), page) ")";  
}
```
- Kapitel bzw. Inhaltsverweis `target-content()`  

```
a.xrefContent {  
  content: "Siehe " target-content(attr(href));  
}
```
- Kombinationen sind möglich



# Umbrüche/Breaks

## ■ Seitenumbruch

```
div.chapter {  
    page-break-after: always;  
}
```

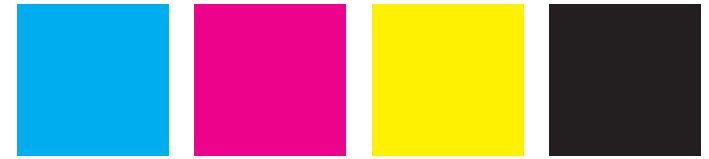
## ■ Umbruchoptionen

Hurenkinder und Schusterjungen

```
p {  
    widows: 2;  
    orphans: 2;  
}
```

## ■ Blocksatz

```
p {  
    hyphens: auto;  
    text-align: justify;  
}
```



# Farben

## CMYK

- kann für Hintergründe und Farben definiert werden

```
p.blau {  
  color: device-cmyk(1.0, 0.5, 0.0, 0.3);  
}
```

## Bildworkflow

- Direkt druckfähige Bilder einbinden
- Color Management via Rendering Engine denkbar/möglich

# Rendering Systeme

- **Prince XML**  
<http://www.princexml.com>  
Kosten: \$495
- **Antenna House**  
<http://www.antennahouse.com/>  
Kosten: \$1,750 for a single user license
- **wkhtmltopdf**  
<https://github.com/antialize/wkhtmltopdf>  
Basiert auf webkit
- **xhtml2pdf**  
<https://github.com/chrisglass/xhtml2pdf>  
Python based CSS Engine Pisa

# Wie geht es weiter

## Grenzen der Technologie im Bereich Print

- Kein WYSIWYG  
Bildplatzierung im Code anstrengend
- Daten für Print meist nicht in HTML
- Farben Colormanagement
- Trennungen und Wörterbücher

## Ausblick

- HTML und CSS werden wichtiger
- Digital First Workflows
- Automatisierung
- Entwicklungsschub und Open Source Tools

## Reading List

- <http://alistapart.com/article/building-books-with-css3>
- <http://drublic.de/blog/printing-the-web/>
- <http://www.w3.org/TR/2013/WD-css3-page-20130314/>
- <http://alistapart.com/article/boom>
- <http://www.princexml.com/doc/8.1/>

## Tools

- <https://github.com/antialize/wkhtmltopdf>
- <http://www.princexml.com/>
- <http://www.antennahouse.com/>
- <http://www.xhtml2pdf.com/>
- <http://weasyprint.org/>
- <http://www.codeproject.com/Articles/568136/Porting-and-Extending-PDFJS-to-NodeJS>
- <https://github.com/mozilla/pdf.js/>



**Vielen Dank für Ihre Aufmerksamkeit!**

**Fragen und Anregungen?**

Folien: <http://www.publishingx.de/dokumente>

E-Mail: [gregor.fellenz@publishingx.de](mailto:gregor.fellenz@publishingx.de)

Twitter: [grefel](#)

Blog: <http://www.indesignjs.de>