

CSS

Syntax

Crossmedia  
Publishing

Paged Media Module

HTML

Renderer

Workflows

# Printlayouts mit CSS3

6. November 2013

#tekom13

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

[tobias.fischer@pagina-tuebingen.de](mailto:tobias.fischer@pagina-tuebingen.de)

# Old School Publishing



## 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?!?

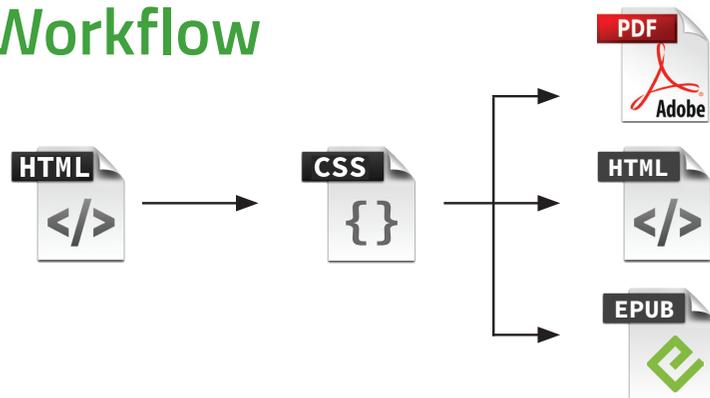
# Crossmedia Publishing

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 notwendig

### ■ XSL-FO

Seitenbeschreibungssprache in XML, gerendert wird meist PDF.

*Liam Quin, W3C:* „The answer is probably to invest in CSS, not XSL-FO, these days“

- 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

# Demo

# 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/>
- In Browsern bisher nur rudimentär unterstützt

## 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>

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

## Mehrspaltige Layouts

- Mehrspaltige Bereiche sind möglich

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

## 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()`

## Fußnoten

- Nicht Teil des Paged Media Module, sondern ein eigener W3C-Draft:  
„CSS Generated Content for Paged Media Module“ ( <http://www.w3.org/TR/css3-gcpm/> )

- Fußnoten werden gefloatet

```
.fn { float: footnote; }
```

- Neuer Seitenbereich `@footnotes` der die Fußnoten-Floats aufnimmt

- Reservierter Counter „footnote“

- Stylingmöglichkeiten ähnlich wie für Listen

- Eigene Pseudo-Elemente für Fußnotenzähler und -marker

```
.fn::footnote-call { content: "[" counter(footnote) "]" ; }
```

```
.fn::footnote-marker { font-weight: bold; }
```

- Keine Block-Level-Elemente in Fußnote möglich



## Cross References

### Aufbau von dynamischen Querverweisen

- Seitenverweis `target-counter()`  

```
<p>Siehe <a href="#kapitel5" class="verweis">Kapitel 5</a>.</p>  
a.verweis:after {  
  content: " (Seite " target-counter(attr(href), page) ")";  
}
```
- Inhaltsverweis `target-content()`  

```
a.verweis {  
  content: "Siehe " target-content(attr(href));  
}
```

# Umbrüche/Breaks

- Seitenumbruch

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

- Bilder zusammenhalten

```
figure {  
    page-break-inside: avoid;  
}
```

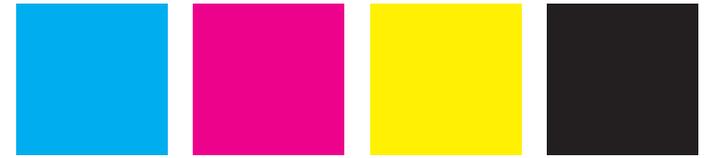
- Umbruchoptionen

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);  
}
```
- Best Practice: Immer in Media Queries definieren.

## Bildworkflow

- Direkt druckfähige CMYK-Bilder einbinden?
- RGB-Workflow und Color Management mit Antenna House möglich  
Valides PDF/X3

# Rendering Systeme

- **Prince XML**  
<http://www.princexml.com>  
Kosten: \$495
- **Antenna House**  
<http://www.antennahouse.com>  
Kosten: ab \$1,250 (single user license)
- **xhtml2pdf**  
<https://github.com/chrisglass/xhtml2pdf>  
Python based, aktive Entwicklung, (fast) nur CSS2.1 (kein paged-media)
- **weasyprint**  
<http://weasyprint.org>  
Aktive Entwicklung, CSS2.1 gut abgedeckt, paged-media nur rudimentär
- **wkhtmltopdf**  
<https://github.com/antialize/wkhtmltopdf>  
Basiert auf webkit, kaum aktiv, (fast) nur CSS2.1 (kein paged-media)

# PrinceXML

- Einzelplatz- und Server-Lizenzen  
Kostenlos für nicht-kommerzielle Nutzung (dezentes Wasserzeichen)
- Gute Unterstützung der CSS3-Standards
- Wenige CSS-Extensions (z.B. für Spotfarben)
- Wörterbücher für Trennungen möglich
- Keine XSL-FO Unterstützung

## Nachteile

- PDF/X-1a nur rudimentär
- Kein Output nach PDF/X (*geplant für 2013?*)
- Keine Colormanagement-Funktionen
  - Keine Dokumentprofile (*RGB only*)
  - Profile an Abbildungen werden entfernt

# Antenna House

- Einzelplatz- und Server-Lizenzen
- Gute Unterstützung der CSS3-Standards
- 200+ CSS-Extensions (meist analog zu FO)
- Output nach PDF/X bzw. PDF/A möglich
- Wörterbücher für Trennungen möglich
- Eigene CSS Lizenz notwendig

## Kleine Details

- Parst offenbar auch `@media screen` und „verschluckt“ sich an Screen-spezifischen CSS-Eigenschaften wie z.B. der Farbangabe `rgba()`
- Acid-Test 2 kann nicht gerendert werden

# Wie geht es weiter

## Grenzen der Technologie im Bereich Print

- Kein WYSIWYG  
Bildplatzierung im Code anstrengend  
Debug nur durch Rendering
- Daten für Print meist nicht in HTML
- Color Management
- Typographische Feinjustierung
- tbd: <http://www.w3.org/Style/2013/paged-media-tasks>

## Ausblick

- HTML und CSS werden wichtiger
- Digital First Workflows
- Automatisierung
- Way to go... Open Source Tools

## Standards

- **Paged Media Module Level 3**  
<http://www.w3.org/TR/css3-page/>  
Stand: 14.03.2013 (WorkingDraft)
- **CSS Generated Content for Paged Media Module**  
<http://www.w3.org/TR/css3-gcpm/>  
Stand: 29.11.2011 (WorkingDraft)

## Reading List

- <http://alistapart.com/article/building-books-with-css3>
- <http://drublic.de/blog/printing-the-web/>
- <http://alistapart.com/article/the-look-that-says-book>
- <http://programming.oreilly.com/2013/09/html5-is-the-future-of-book-authorship.html>
- <http://balisage.net/Proceedings/vol10/html/Kleinfeld01/BalisageVol10-Kleinfeld01.html>
- <http://www.techrepublic.com/blog/web-designer/going-further-with-the-paged-media-module-and-alternatives-for-creating-paged-media/>
- <http://www.w3.org/2012/12/global-publisher/slides/Day2/P1-w3c-paris-hachette.pdf>

**Vielen Dank für Ihre Aufmerksamkeit!**

**Fragen und Anregungen?**

E-Mail: [gregor.fellenz@publishingx.de](mailto:gregor.fellenz@publishingx.de)  
Twitter: [@grefel](https://twitter.com/grefel)

[tobias.fischer@pagina-tuebingen.de](mailto:tobias.fischer@pagina-tuebingen.de)  
[@tobias\\_fischer](https://twitter.com/tobias_fischer)