

CSS

Syntax

Crossmedia
Publishing

Paged Media Module

HTML

Workflows

Renderer

Crossmedia Publishing

~~Printlayouts~~ mit HTML 5 & CSS 3

28. Februar 2015

#eBookCamp München

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

Tobias Fischer



- Studium »*Druck- und Medientechnologie*«
an der Hochschule der Medien in Stuttgart
 - Schwerpunkte: Cross-Media-Publishing, XML
 - Abschluss 2011: »*Einbindung von Multimedia-Inhalten in crossmediale XML-Workflows*«
- Freier XSLT- und Print-CSS-Entwickler
- Festanstellung beim Verlagsdienstleister und XML-Spezialisten *pagina GmbH* in Tübingen
 - XML- und XSLT-Entwicklung
 - E-Book-Automation
 - XML-Workflows
 - Produktmanagement „parsX“ parsx.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 ist kaum ein Mehrwert ersichtlich?!?

Crossmedia Publishing

Medienn neutrale Workflows für unterschiedliche Ausgabeformate

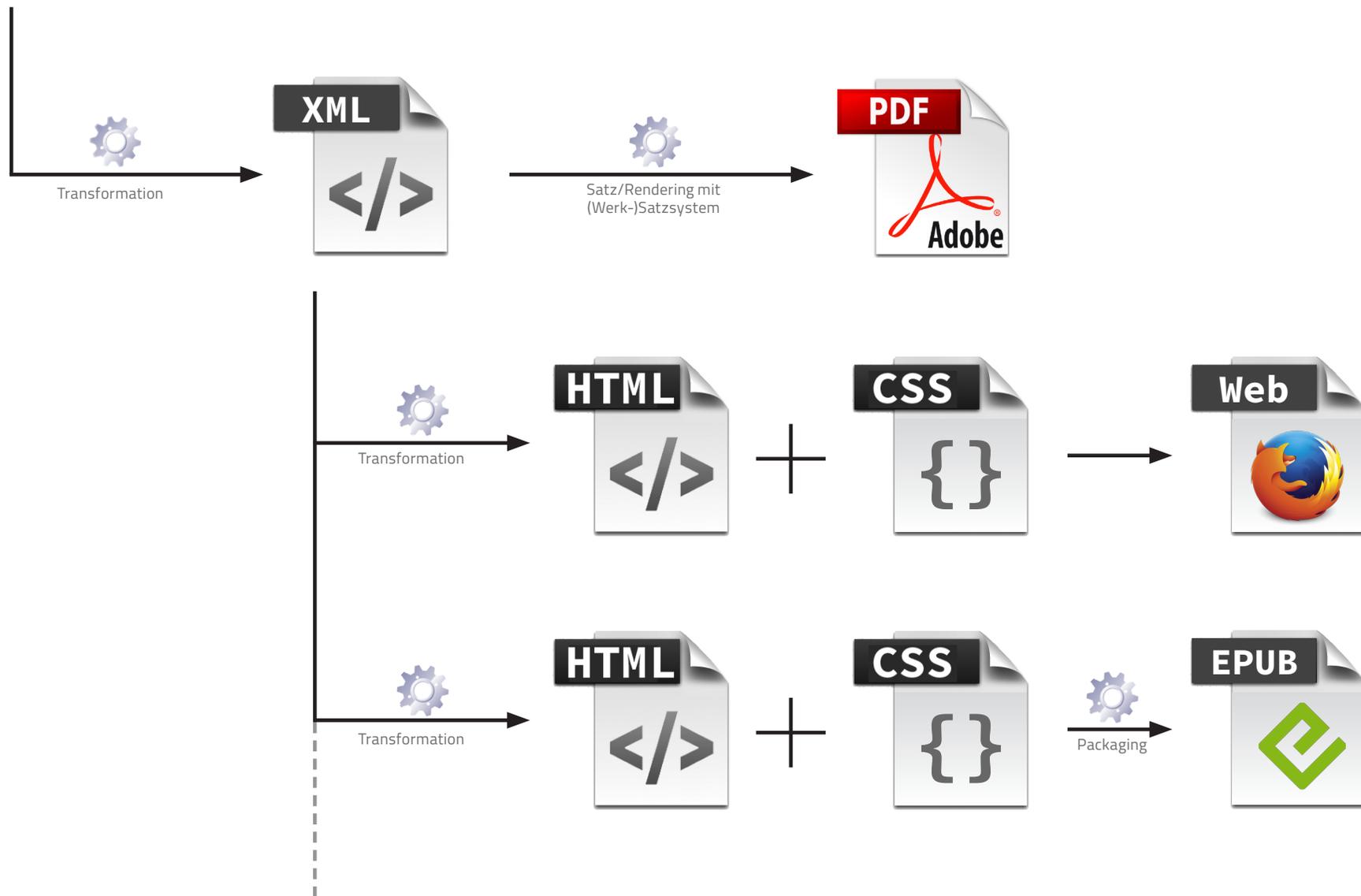
Chancen

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



Typischer XML-Workflow in der Verlagsindustrie

(stark vereinfachte Darstellung)



Mögliche Varianten für Print-Output

■ 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 zwingend notwendig

■ HTML + CSS

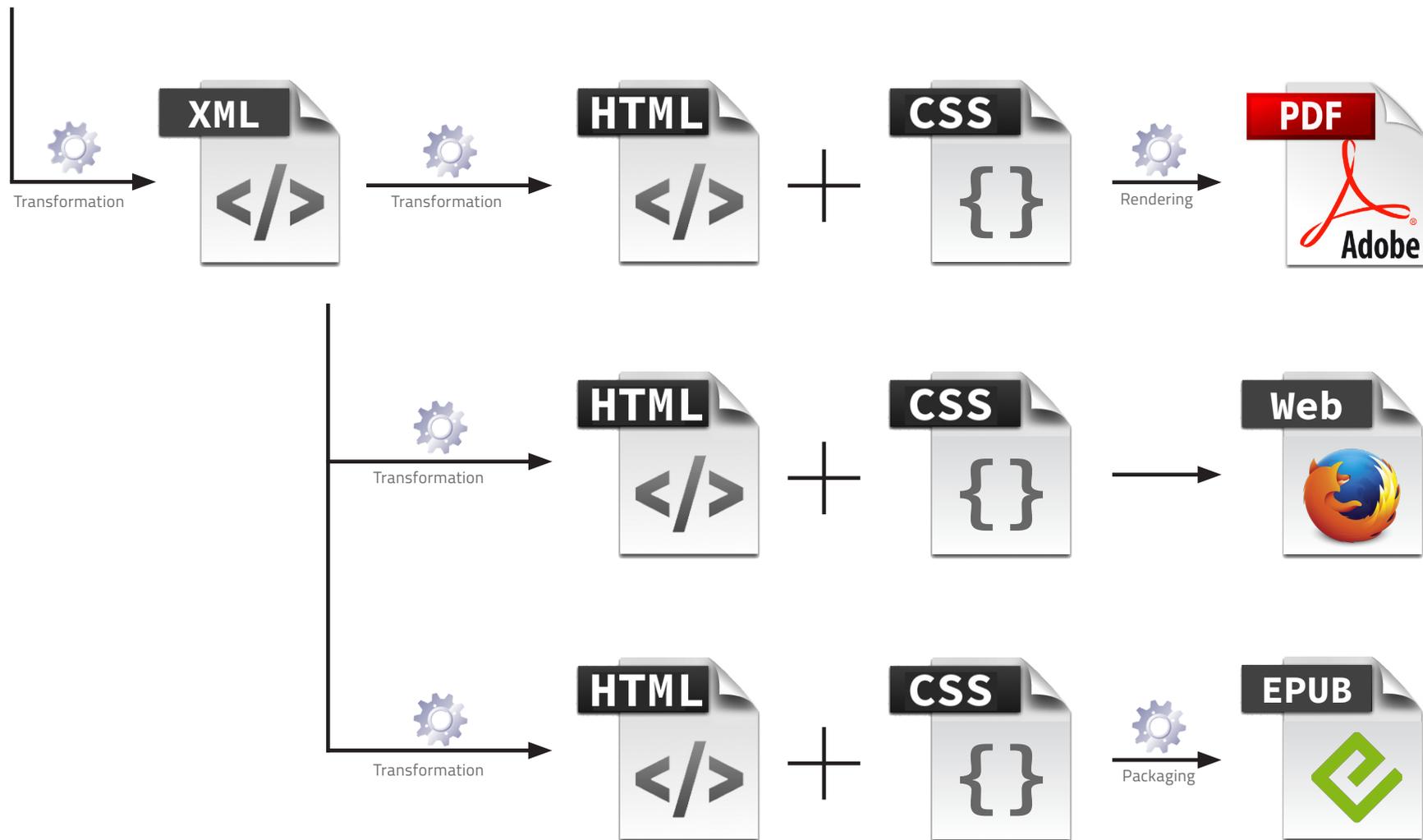
Sprache des Web, einfacher in der Anwendung.

NELLIE MCKESSON, O'REILLY: „*3 mal so schnell wie XSL-FO*“ [Anm.: Lernkurve bzw. Layout]

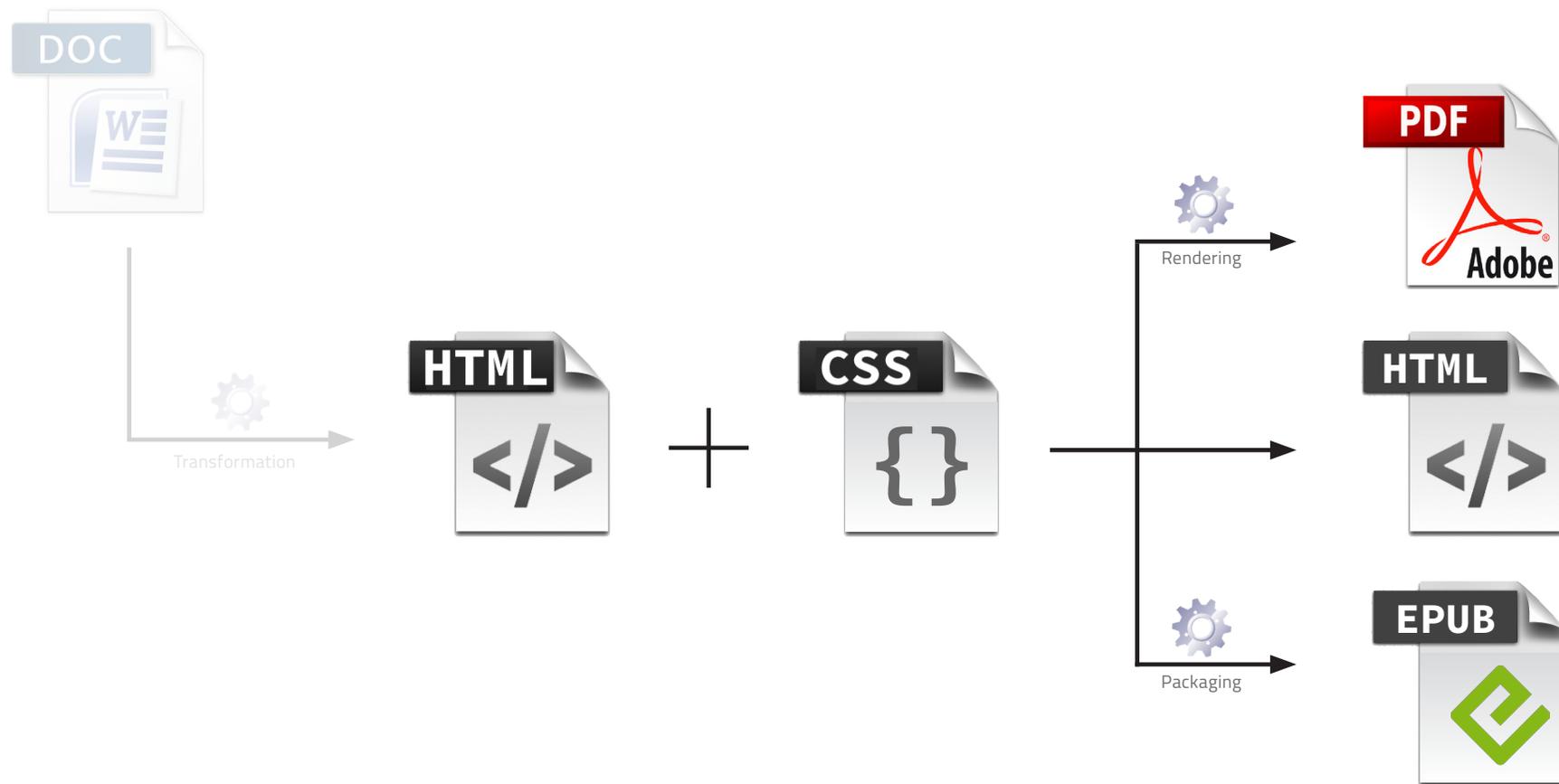
- Vorteil: Viele beherrschen CSS und HTML
- Direkte Weiterverwendung von Webdaten (Idealfall)



Möglicher XML-Workflow mit HTML + CSS



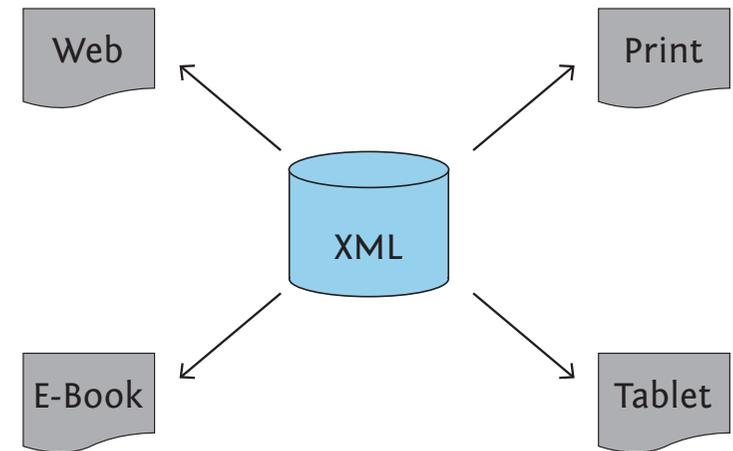
Idealer HTML+ CSS-Workflow



**Brauchen wir dann
überhaupt noch XML?**

Recap:

XML-Crossmedia Publishing



Grafik: Vortrag »Produktion digitaler Bücher«
von Gregor Fellenz, publishingX.de, 2013

Warum XML?

- **Datenchaos** beenden
- **Mehrfachverwertung** der Daten
- **Automatisierte Produktion** und Standardisierung im Publishing
- Kein DTP-Programm notwendig

XML ist Dateiformat

- Sprache zur Beschreibung von Dokumentstrukturen
→ Metasprache, die eine abstrakte Syntax für Auszeichnungssprachen definiert
- Exportierbar aus anderen (Redaktions-/Content-Management-) Systemen

HTML statt XML?

Gerne! Aber bitte XHTML 5!

- XHTML 5 ist XML!
 - sieht bereits eigene semantische Elemente vor (<section>, <hgroup>, <figure>, etc...)
 - Auch geeignet für E-Book-Multimedia-Inhalte (<audio>, <video>, <math>, <svg>, <canvas>,...)
- Es entfällt ein großer Transformationsschritt (XML -> XHTML)
- Neutrales, weit verbreitetes Format mit großer Tool-Unterstützung
- Für viele Anwendungsfälle keine speziellen Editoren notwendig
-> haufenweise WYSIWYG-Editoren für's Web
- Viel Know-How und Erfahrung in der (Web-) Community
- O'Reilly: HTMLBook – »Let's write books in HTML!«

<https://github.com/oreillymedia/HTMLBook>

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

- Definition der Seite mit @page

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

- Zuweisung eines Bereichs über Eigenschaft page

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

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;  
}
```

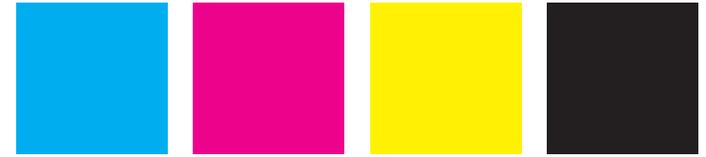
- Umbruchoptionen

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

PDF-Renderer

- **Prince XML**
<http://www.princexml.com>
Kosten: \$495
- **Antenna House**
<http://www.antennahouse.com>
Kosten: ab \$1,250 (single user license)
- **PDFreactor**
<http://www.realobjects.com/products/pdfreactor/>
Kosten: ab \$2,950
- **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
- Output nur nach PDF/A möglich
- JavaScript-Unterstützung

Nachteile

- 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
- Eigenständige CSS-Lizenz notwendig

Kleine Details

- Parst offenbar auch `@media screen` und „verschluckt“ sich an Screen-spezifischen CSS-Eigenschaften wie z.B. der Farbangabe `rgba()`

bisher noch ungetestet, sorry...

PDFreactor

- Nur Server-Lizenzen
Kostenlos für private, nicht-kommerzielle Nutzung
- Gute Unterstützung der CSS3-Standards
- Output nur nach PDF/A möglich?
- Unterstützung für HTML-Daten mit JavaScript-Inhalten (Charts, etc...)
- Kann Tagged-PDF erstellen
- Digital signierte Dokumente möglich

Grenzen der Technologie

im Bereich Print

- Kein WYSIWYG-Editor vorhanden
 - Bildplatzierung im Code anstrengend
 - Debugging nur durch erneutes Rendering
- CSS 3-PMM (und Renderer) noch nicht so leistungsfähig wie XSL-FO, InDesign, 3B2, etc...
 - Kolummentitel
 - Marginalspalten
 - Layoutsatz
 - etc...
- Daten für Print meist (noch) nicht in HTML
- Color Management
- Typographische Feinjustierung
- Teilweise immernoch XSLT- oder JavaScript-Eingriffe notwendig
 - Generierte Inhalte (Index, Inhaltsverzeichnis)
- 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: 13.05.2014 (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>
- <http://www.smashingmagazine.com/2015/01/07/designing-for-print-with-css/>
- http://digitalpub.abcd.harvard.edu/files/digitalpub/files/html5_harvard_presentation.pdf

Vielen Dank für Ihre Aufmerksamkeit!

Fragen und Anregungen?

pagina: tobias.fischer@pagina-tuebingen.de
freelance: tobias.fischer@mediaversal.de
Twitter: @tobias_fischer