CSS Paged Media

Create Beautiful PDFs With Ease

Jirka Kosek

Technical Writers' Exchange of Experience Workshop
June 14th, 2018

About me

- consultancy for publishing industry
- evangelizing Web and XML technologies for past 20 years
- core member of DocBook project
- member of many standardization groups – XSLT, XSL/FO, XML, HTML5, CSS, DocBook, OOXML, ODF, …
- organizer of XML Prague conference

Do we still need to target paper when delivering documentation?

- usually yes
- some people still prefer reading from paper
- sometimes it is legal requirement to deliver printed version of docs
- … or PDF/A for archiving purposes
- let's simplify print to PDF for the rest of the presentation

How to obtain PDF from your publishing pipeline

- PDF is the only output format
  - DTP tools (InDesign, Framemaker, Word, LO, …) can be used directly for authoring and rendering
- more output formats are needed
  - single source solution has to be used
  - sources in XML (DocBook, DITA) or other structured format (AsciiDoc, MarkDown)

- PDF is usually obtained by using XSL/FO or TeX
- customization of XSL/FO or TeX outputs is hard
- finding people with XSL/FO and TeX skills is harder and harder

Heresy

“Let's use HTML and CSS for producing PDFs”

- we already have HTML as we have to deliver content in HTML for online docs in browser
- there is no need to learn additional “obscure” technology as CSS knowledge is widespread
- … but is not CSS too limited?

What is missing in CSS support in browsers

- page size and layout
- headers and footers
- generated content, numbering, cross references
- page/line-break control
- hyphenation
- footnotes
- table of contents, index
- bookmarks (outline)

Page dimensions and layout

```
/* Basic page dimensions and orientation */
@page { size: a5 landscape; }

/* Different settings for first, odd and even pages */
@page :first { … }
@page :left { … }
@page :right  { … }

/* Page masters can be named
   so more of them can be used in the document */
@page cover { … }
.coverpage { page: cover; }
```

Headers and footers

```
@page {
  size: a4;

  @top-left {
    content: "Ze života hmyzu";
  }

  @bottom-right {
    content: counter(page);
  }
}
```

Live header/footer

```
<section class="chapter">
  <h3>Title</h3>


section.chapter h3 {
  string-set: chapter contents();
}

…

@top-left {
  content: string(chapter);
}
```

Automatic numbering

```
<section class="chapter">
  <h3>Title</h3>

section.chapter h3 {
  counter-increment: chapter;
}

section.chapter h3::before {
  content: "Chapter " counter(chapter);
}
```

Break control

- document is being typeset automatically without possibility of user intervention
- constraints can be used to make sure that breaks are reasonable
- properties: `page-break-before`, `page-break-after` a `page-break-inside`
- values: `auto`, `always`, `avoid`, `left` a `right`

```
/* Do not break page after heading */
h2 + p { page-break-before: avoid }
```

Table of contents and index

- CSS can't generate these
- ToC and index can be created manually, but usually XSLT or some scripting language is used
- CSS can resolve links and replace them with target page number or content

Cross references

```
<h3 id="last">Conclusions<h3>

…

<a href="#last" class="pageref">Last chapter on page</a>
is really the last one.

a.pageref::after {
  content: " " target-counter(attr(href url), page);
}
```

Hyphenation

```
<html lang="cs">
```

```
html {
  hyphens: auto;
}
```

Bookmarks

- do not confuse with browser bookmarks, this is document outline

```
h1 { bookmark-level: 1; }
h2 { bookmark-level: 2; }
h3 { bookmark-level: 3; }
```

Footnotes

```
<p>Text that contains<span class="note">Footnote</span> and
then continues.</p>

.note { float: footnote; }
```

Leaders

- filling of all available space

```
<ul class="toc">
  <li><a href="#intro">Introduction</a></li>
  <li><a href="#body">Body of document</a></li>
  <li><a href="#conclusion">Conclusion</a></li>

/* Adding dots to ToC entries */
.toc a::after {
 content: leader(dotted) " " target-counter(attr(href, url),
page);
}
```

Implementations

- Antenna House Formatter
- Prince
- PDFreactor
- Vivliostyle
- Weasy Print
- Oxygen PDF Chemistry

Big users

- O'Reilly publisher
  - HTMLBook – HTML extensions for writing books
  - Atlas – publication platform based on HTMLBook
- Hachette Livre publisher
  - each year they publish several thousand books with CSS

Issues

- slow standardization
- there is not yet good open-source implementation available
- does not support everything what XSL-FO can do

Resources

- CSS Generated Content for Paged Media Module
- CSS Paged Media Module Level 3
- Examples and comparision of various CSS PDF engines
- Introduction to CSS for Paged Media
- Slides: http://www.kosek.cz/xml/2018tww/ (PDF)