



A Data Usability Company

ANTENNA HOUSE

CSS to XSL-FO, XSL-FO to CSS

Tony Graham

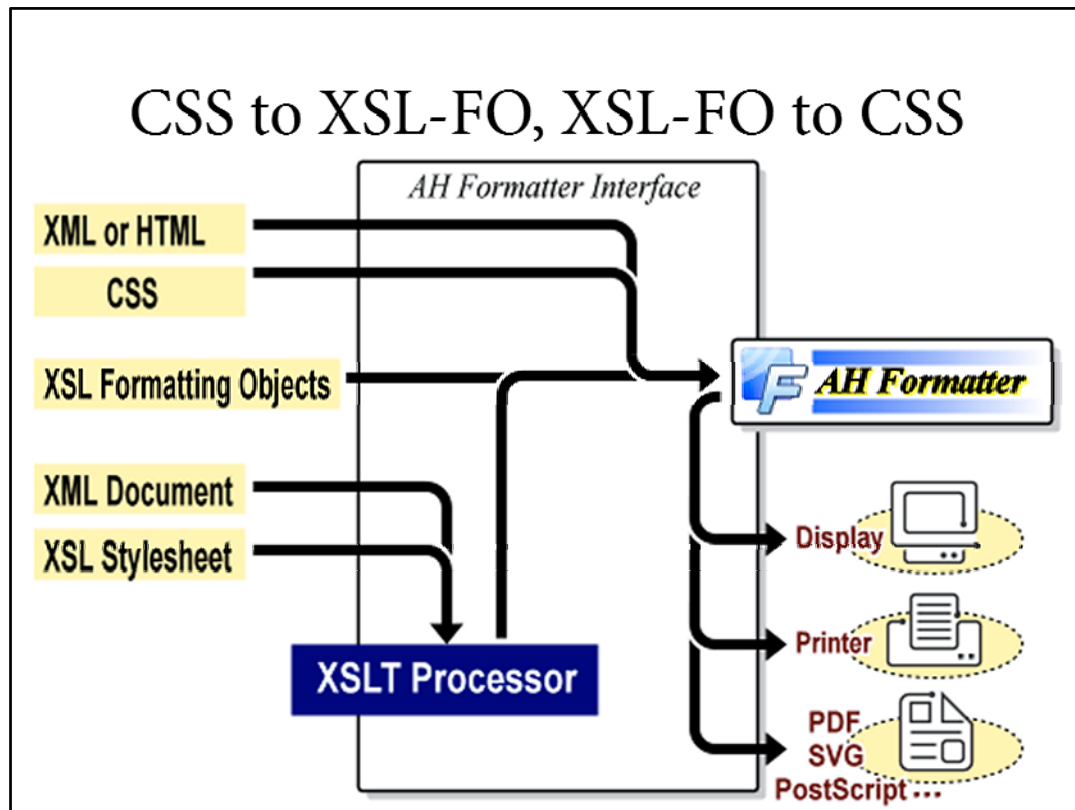
XML Division

Antenna House, Inc.

tgraham@antenna.co.jp

tony@antennahouse.com

@tgraham_antenna



I'm talking today about features in the Antenna House AH Formatter. Firstly because I work for Antenna House, but secondly because it's the only formatter that I know of that implements both XSL-FO and CSS.

When I was preparing this talk, I was thinking that in 20 years of presenting at markup conferences, this is the first time that I've presented about a piece of commercial software that I'm working on.



I had to think to myself, "Is this going to be a marketing talk?" No. Firstly because it's not my field, and, secondly, because I'd be dragged off the stage if this was too much of marketing talk.



The other consideration, of course, is how technical should it be? Should I go into detail about how the cogs of Antenna House AH Formatter fit together? That's a bit too much for half an hour, plus I'd also be stopped if I gave away too many details.

Spread produced with XSL-FO

Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean. A small river named Duden flows by their place and supplies it with the necessary regalia. It is a paradisiac country, in which roasted parts of sentences fly into your mouth. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life.

thead
1
2
3
4
5

2 of 3

6
7
8
9
10
tfoot

One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there were thousands of bad Commas, wild Question Marks and devious Semikoli, but the Little Blind Text didn't listen. She packed her seven versalia, put her initial into the belt and made herself on the way.

壹	One
貳	Two
參	Three
四	Four

When she reached the first hills of the Italic Mountains¹, she had a last view back on the skyline of her hometown Bookmarksgrove, the headline of Alphabet Village and the subline of her own road, the Line Lane. Pityful a rethoric question ran over her cheek, then she continued her way.

On her way she met a copy. The copy warned the Little Blind Text, that where it came from it would have been rewritten a thousand times and

everything that was left from its origin would be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the copy said could convince her and so it didn't take long until a few insidious Copy Writers ambushed her, made her drunk with Longe and Parole and dragged her into their agency, where they abused her for their projects again and again. And if she hasn't been rewritten, then they are still using her.

¹ Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean. A small river named Duden flows by

their place and supplies it with the necessary regalia. It is a paradisiac country, in which roasted parts of sentences fly into your mouth. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life.

3 of 3

The other thing that I thought about doing is that I can show you a spread that's produced with XSL-FO...

Spread produced with HTML+CSS

Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean. A small river named Duden flows by their place and supplies it with the necessary regalia. It is a paradisematic country, in which roasted parts of sentences fly into your mouth. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life.

thead
1
2
3
4
5

2 of 3

6
7
8
9
10
tfoot

One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there were thousands of bad Commas, wild Question Marks and devious Semikoli, but the Little Blind Text didn't listen. She packed her seven versalia, put her initial into the belt and made herself on the way.

壹、One
貳、Two
参、Three

四、Four

When she reached the first hills of the Italic Mountains¹, she had a last view back on the skyline of her hometown Bookmarksgrove, the headline of Alphabet Village and the subline of her own road, the Line Lane. Pityful a rethoric question ran over her cheek, then she continued her way.

On her way she met a copy. The copy warned the Little Blind Text, that where it came from it would have been re-

written a thousand times and everything that was left from its origin would be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the copy said could convince her and so it didn't take long until a few insidious Copy Writers ambushed her, made her drunk with Longe and Parole and dragged her into their agency, where they abused her for their projects again and again. And if she hasn't been rewritten, then they are still using her.

¹ Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean. A small river named Duden flows by

their place and supplies it with the necessary regalia. It is a paradisematic country, in which roasted parts of sentences fly into your mouth. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life.

3 of 3

...and then I can show you a spread that's produced with HTML and CSS (and if I'd spent more time, they would have been pixel-perfect the same).

I could spend 25 minutes showing you those, but that would be even more boring for you than it would be for me.

CSS to XSL-FO

Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean. A small river named Duden flows by their place and supplies it with the necessary regalia. It is a paradisiac country, in which roasted parts of sentences fly into your mouth. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life.

thead
1
2
3
4
5

2 of 3

6
7
8
9
10
tfoot

One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there were thousands of bad Commas, wild Question Marks and devious Semikoli, but the Little Blind Text didn't listen. She packed her seven versalia, put her initial into the belt and made herself on the way.

壹	One
貳	Two
參	Three
四	Four

When she reached the first hills of the Italic Mountains¹, she had a last view back on the skyline of her hometown Bookmarksgrove, the headline of Alphabet Village and the subline of her own road, the Line Lane. Pityful a rethoric question ran over her cheek, then she continued her way.

On her way she met a copy. The copy warned the Little Blind Text, that where it came from it would have been rewritten a thousand times and

everything that was left from its origin would be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the copy said could convince her and so it didn't take long until a few insidious Copy Writers ambushed her, made her drunk with Longe and Parole and dragged her into their agency, where they abused her for their projects again and again. And if she hasn't been rewritten, then they are still using her.

¹ Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean. A small river named Duden flows by

their place and supplies it with the necessary regalia. It is a paradisiac country, in which roasted parts of sentences fly into your mouth. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life.

3 of 3

So I'll just concentrate on a few features, aspects of CSS formatting that have crossed-over to be extensions in the XSL-FO side of AH Formatter.

I'm going to talk about counter styles, which is a new CSS feature which I think is generally useful for internationalization.

I'm also going to talk about footnotes.

XSL-FO to CSS

Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean. A small river named Duden flows by their place and supplies it with the necessary regalia. It is a paradisiacal country, in which roasted parts of sentences fly into your mouth. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life.

thead
1
2
3
4
5
6

2 of 3

One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there were thousands of bad Commas, wild Question Marks and devious Semikoli, but the Little Blind Text didn't listen. She packed her seven versalia, put her initial into the belt and made herself on the way.

壹、One
貳、Two
参、Three

四、Four

When she reached the first hills of the Italic Mountains¹, she had a last view back on the skyline of her hometown Bookmarksgrove, the headline of Alphabet Village and the subline of her own road, the Line Lane. Pityful a rethoric question ran over her cheek, then she continued her way.

On her way she met a copy. The copy warned the Little Blind Text, that where it came from it would have been re-

written a thousand times and everything that was left from its origin would be the word "and" and the Little Blind Text should turn around and return to its own, safe country. But nothing the copy said could convince her and so it didn't take long until a few insidious Copy Writers ambushed her, made her drunk with Longe and Parole and dragged her into their agency, where they abused her for their projects again and again. And if she hasn't been rewritten, then they are still using her.

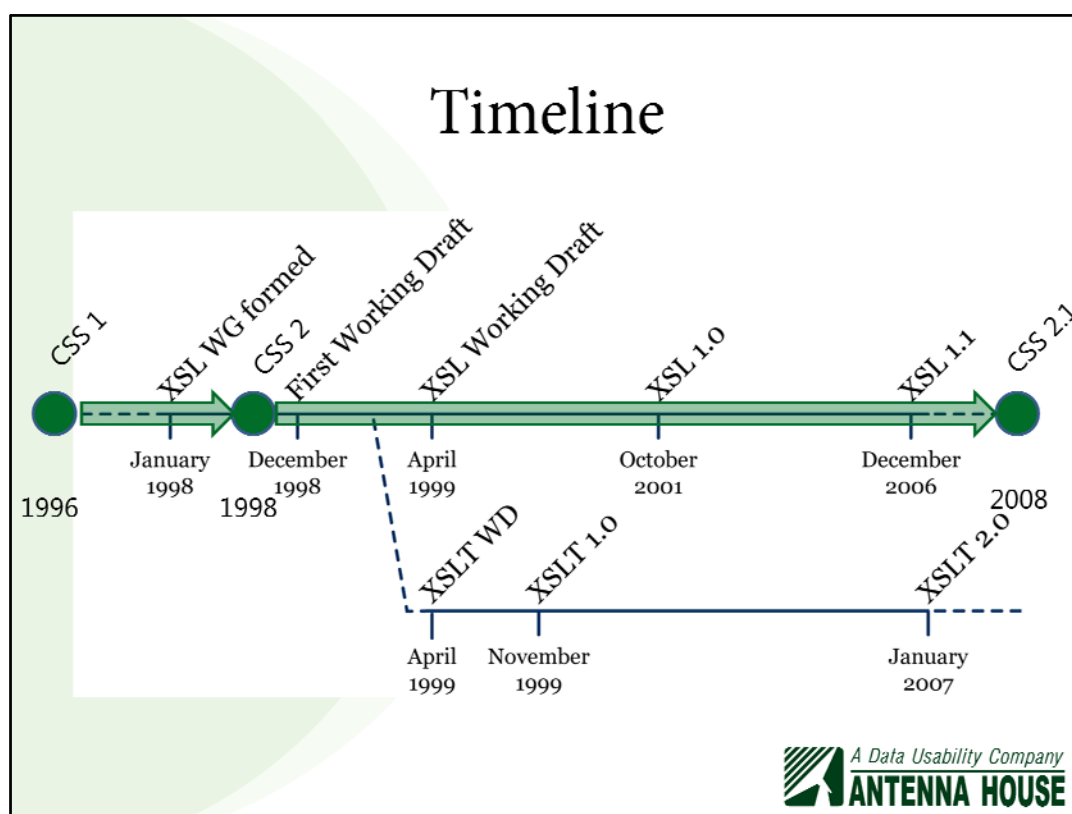
¹ Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean. A small river named Duden flows by their place and supplies it with the necessary regalia. It is a paradisiacal country, in which roasted parts of sentences fly into your mouth. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life.

3 of 3

On the XSL-FO features carried across to the CSS side of AH Formatter, I'm going to talk about headers and footers of tables, about hyphenation properties, about the running headers that are implemented for CSS, and also about footnotes, since part of what was implemented for CSS was carried over to the XSL-FO side and part of what was implemented for XSL-FO was carried over to the CSS side.

The way these are implemented, sometimes it's the same property carried over, and sometimes CSS and XSL-FO have different syntaxes for what is essentially the same thing so they boil down to the same machinery inside AH Formatter to implement this.

Anything I'm not talking about you can assume is already common to XSL-FO and CSS.



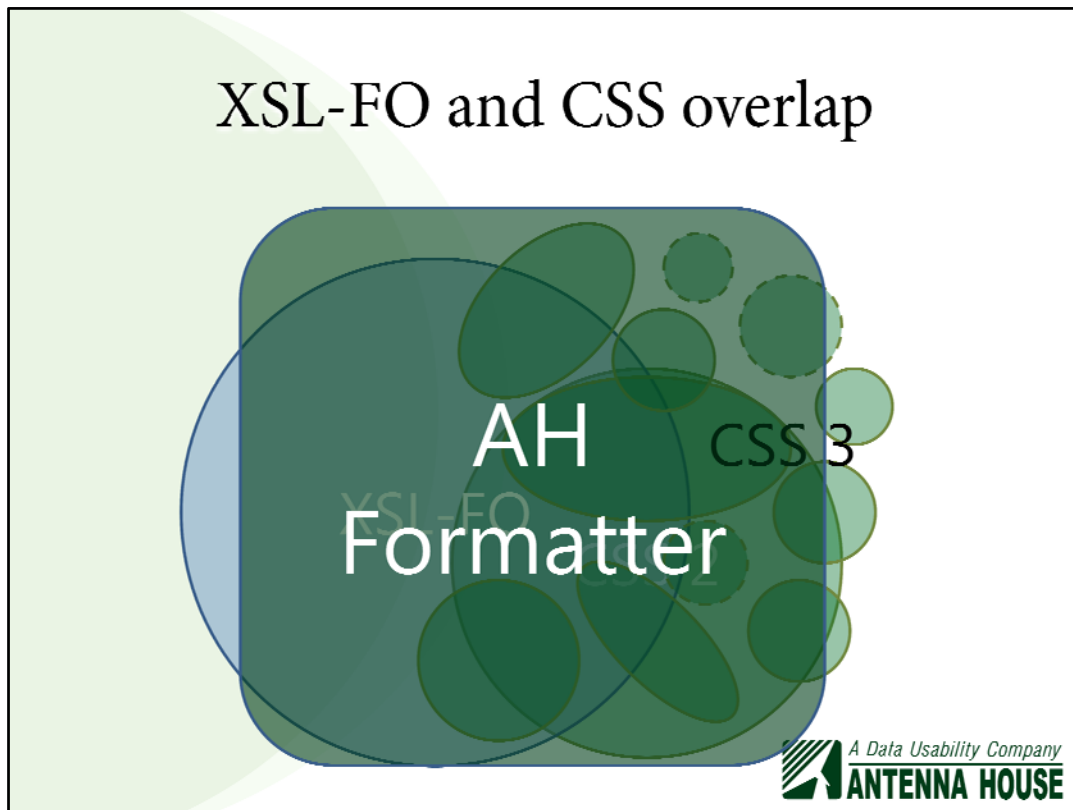
How come they have so much in common to begin with?

This is the timeline for XSL-FO. The Working Group was formed in 1998, and XSL 1.0 came out in 2001, and XSL 1.1 came out in 2006.

But CSS was operating at the same time: CSS 1 was produced in 1996, and during the time that XSL-FO was being worked on, CSS was producing CSS 2.

The XSL-FO people were required to work with the CSS people so that what they had in common, where they expressed the same thing, had to use the same properties. The wider W3C required that the XSL-FO people and the CSS people work together to define the common properties. A comment from a person there at the time was "The XSL-FO work was basically free to do anything it wanted as long as anything that could be done in CSS was done exactly the same way in XSL-FO." They did work together, but it wasn't always harmonious.

While XSL-FO was being developed, CSS had a Recommendation, and they were working on and off on CSS 2.1, which is a rationalization of CSS 2, and if you want to look in the XSL spec, you'll find that there's many references from the XSL spec both to CSS 2 and to errata that had been raised against CSS 2, so there's a lot of commonality.



So there's a lot of overlap between the two technologies: things like properties for fonts, margins, the box model, borders and padding, baselines, the list goes on...

If you take this to the CSS 3 level, there's still a lot of commonality between CSS modules and XSL-FO, plus other parts that either don't relate to what XSL-FO is targeting or have been produced since.

Antenna House started out as an XSL-FO formatter and, along the way, they added the capability to handle CSS. As a result, they can handle almost all of XSL-FO – you can't, for example, do unbounded page sizes – and it handles a large proportion of CSS, plus it has extra extensions of its own.

XSL/CSS Properties List		
<p>The following table shows XSL-FO properties and their corresponding CSS properties. A blank cell indicates that no corresponding CSS property is implemented for the XSL-FO property, or vice-versa. See also XSL-FO Conformance or CSS Conformance to learn the current implementation status for each property. CSS Conformance also defines the CSS module abbreviations, such as [CSS3-GCPM] etc., that are used in the table. The fact that a property is implemented in both XSL-FO and CSS does not always mean that their specifications in XSL-FO and CSS are completely aligned. Some properties are functionally equivalent or mostly similar.</p>		
	CSS	Description
axf:deviation-character-count		Specifies the minimum number of characters considered to be an
7.6.1 absolute-position	[CSS2.1] position	
axf:action-type	-ah-action-type	
axf:additive-symbols	[CSS3-CounterStyle] (-ah-)additive-symbols	axf:counter-style
	[CSS2.1]vertical-align	
		[CSS3-Line] Setting the alignment point: the 'alignment-adjust' property
		[CSS3-Line] Aligning the alignment point of an element: the 'alignment-baseline' property
	baseline	
7.15.1 allowed-height-scale	-ah-allowed-height-scale	
7.15.2 allowed-width-scale	-ah-allowed-width-scale	
axf:alt-glyph	-ah-alt-glyph	Specifies the alternative glyph of a character.
axf:alt		Specifies the alternate text of the image.
axf:author		Specifies the author of the annotation. V6.2MR2 no-LT

If you look in the Online Manual for AH Formatter, one of the pages is this properties list.

If you want to work out which of these is originally XSL-FO and which are originally CSS:

- A property that has a section number for XSL-FO and a spec reference for CSS is one of the properties that is common to both of them.
- A property that is prefixed with 'axf:' for XSL-FO and prefixed with '-ah-' for CSS is an extension that Antenna House has done, usually in response to a customer request.
- A property that is prefixed on the XSL-FO side and has a reference to a CSS spec is something that is in CSS that has been reimplemented for XSL-FO.
- A property that has a section number reference for the XSL-FO property and a prefix for the CSS property is something in XSL-FO that is reimplemented for CSS.

XSL-FO to CSS: Omitting table header/footer at break

Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean. A small river named Duden flows by their place and supplies it with the necessary regalia. It is a paradise-matic country, in which roasted parts of sentences fly into your mouth. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life.

thead
1
2
3
4
5
6

2 of 3

6
7
8
9
10
tfoot

One day however a small line of blind text by the name of Lorem Ipsum decided to leave for the far World of Grammar. The Big Oxmox advised her not to do so, because there were thousands of bad Commas, wild Question Marks and devious Semikoli, but the Little Blind Text didn't listen. She packed her seven versalia, put her initial into the belt and made herself on the way.

壹、One
貳、Two
参、Three

- CSS has/had no control over repeating header/footer at breaks
- CSS2.1:
 - Print user agents may repeat header rows on each page spanned by a table.
 - Print user agents may repeat footer rows on each page spanned by a table.
- Browsers inconsistent



The first of these is controlling whether to omit the header or footer when a table breaks across a page or column.

CSS 2.1 says that a print User Agent has the option to omit them or repeat them. This came up on the www-style@w3.org mailing list in January when someone pointed out that it wasn't nailed down and browsers were inconsistent. This person was looking for a CSS property to control whether headers and footers were repeated or not.

XSL-FO to CSS: Omitting table header/footer at break

```
.omit {  
  -ah-table-omit-header-at-break: true;  
  -ah-table-omit-footer-at-break: true;  
}  
...  
<table class="omit">  
  <thead>  
    <tr><td>thead</td></tr>  
  </thead>  
  <tfoot>  
    <tr><td>tfoot</td></tr>  
  </tfoot>  
  <tbody>  
    <tr><td>1</td></tr>  
    ...  
    <tr><td>10</td></tr>  
  </tbody>  
</table>
```



XSL-FO has had two properties controlling this since XSL 1.0: `table-omit-header-at-break` and `table-omit-footer-at-break`.

These have just been transferred across as the XSL-FO property expressed in CSS syntax.

XSL-FO to CSS: Omitting table header/footer at break

- XSL-FO: table-omit-header-at-break, table-omit-footer-at-break
- CSS: -ah-table-omit-header-at-break, -ah-table-omit-footer-at-break

thead	thead	thead	7
1	6	1	8
2	7	2	9
3	8	3	10
4	9	4	tfoot
5	10	5	
tfoot	tfoot	6	

-ah-table-omit-header-at-break: false

-ah-table-omit-footer-at-break: false

-ah-table-omit-header-at-break: true

-ah-table-omit-footer-at-break: true

You can see how these control whether the table headers and footers appear or not.

This isn't a very exciting thing unless you absolutely have to have the headers and footers or you absolutely don't, but it illustrates the simplest level of reimplementing one property for another.

XSL-FO to CSS: Hyphenation

7.10.4 hyphenate	[CSS3-Text] (-ah-)hyphens
axf:hyphenate-hyphenated-word	-ah-hyphenate-hyphenated-word
7.10.5 hyphenation-character	[CSS3-GCPM] (-ah-)hyphenate-character
7.16.1 hyphenation-keep	-ah-hyphenation-keep
7.16.2 hyphenation-ladder-count	[CSS3-GCPM] (-ah-)hyphenate-lines
axf:hyphenation-minimum-character-count	-ah-hyphenation-minimum-character-count
7.10.6 hyphenation-push-character-count	[CSS3-GCPM] (-ah-)hyphenate-after
7.10.7 hyphenation-remain-character-count	[CSS3-GCPM] (-ah-)hyphenate-before
axf:hyphenation-zone	-ah-hyphenation-zone

Another thing that has been implemented for XSL-FO and provided for CSS is the hyphenation properties.

There's a lot of properties for hyphenation in XSL-FO and there's a lot of properties for hyphenation in CSS. I can assure you that they didn't implement two separate hyphenation engines, so what's common is the implementation.

XSL-FO to CSS: Hyphenation

- XSL-FO properties reimplemented as similar-but-different CSS properties
- XSL-FO:
 - hyphenate: true | false
 - hyphenation-character: <character> | inherit
- CSS:
 - hyphens: none | manual | auto
 - hyphenate-character: auto | <string>

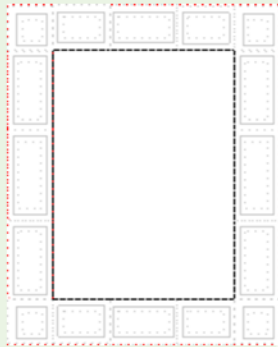


What's common is the implementation, and they had the XSL-FO implementation before they had the CSS one.

This is one where the syntaxes differ and the allowed property values differ but their effects are the same. In XSL-FO, it's 'hyphenate' versus 'hyphens' in CSS, and 'hyphenation-character' versus 'hyphenate-character'.

XSL-FO to CSS: Headers & Footers

CSS



XSL-FO



Another aspect where the syntaxes are completely different but the implementation is in common is in headers and footers.

CSS currently defines 16 areas around the page where you can put strings or, more recently, running elements. XSL-FO defines these four outer regions...

XSL-FO to CSS: Headers & Footers

First	Even	Odd	Even
			
<pre> <fo:static-content flow-name="Even-Header"> <fo:block>UDHR in Unicode</fo:block> </fo:static-content> <fo:static-content flow-name="First-Outside"> <fo:block-container reference-orientation="270"> ... </fo:block-container> </fo:static-content> <fo:static-content flow-name="Odd-Outside"> <fo:block-container reference-orientation="270"> ... </fo:block-container> </fo:static-content> </pre>		<pre> <fo:static-content flow-name="Odd-Header"> <fo:block>...</fo:block> </fo:static-content> <fo:static-content flow-name="Even-Footer"> <fo:block><fo:page-number/></fo:block> </fo:static-content> <fo:static-content flow-name="Odd-Footer"> <fo:block><fo:page-number/></fo:block> </fo:static-content> <fo:flow flow-name="xsl-region-body" id="sco"> ... </fo:flow> </pre>	

...where you can put almost anything by using 'fo:static-content' and using 'fo:marker' and 'fo:retrieve-marker' to get content from your body flow into the headers and the footers.

XSL-FO to CSS: Headers & Footers

- XSL-FO `fo:static-content` allows almost any content
- 16 predefined areas in CSS
- CSS `string()` and running elements implemented similarly to XSL-FO
`fo:marker` and `fo:retrieve-marker`



Since they have `fo:static-content` with `fo:retrieve-marker`, they've just reused the same basic machinery underneath. The CSS syntax and what they're trying to do with it is different from XSL-FO, but it's still the same engine underneath.

XSL-FO to CSS: Footnotes

- Machinery for XSL-FO footnotes reused with CSS

¹ Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts. Separated they live in Bookmarksgrove right at the coast of the Semantics, a large language ocean. A small river named Duden flows by

their place and supplies it with the necessary regalia. It is a paradisiatic country, in which roasted parts of sentences fly into your mouth. Even the all-powerful Pointing has no control about the blind texts it is an almost unorthographic life.



Another thing which would have been implemented for XSL-FO and reimplemented for CSS is footnotes.

Syntaxes differ

Italic Mountains>... ,

```
the Italic Mountains<fo:footnote font-size="0.8em">
  <fo:inline baseline-shift="super" font-size="0.8em">1</fo:inline>
  <fo:footnote-body>
    <fo:list-block space-before="12pt" space-after="12pt" start-indent="-6pt">
      <fo:list-item>
        <fo:list-item-label end-indent="label-end()">...</fo:list-item-label>
        <fo:list-item-body start-indent="0">...</fo:list-item-body>
      </fo:list-item>
    </fo:list-block>
  </fo:footnote-body>
</fo:footnote>, ...
```



The syntaxes for declaring footnotes are obviously different. You have more control about what you put in in the XSL-FO.

Processing differs

- CSS has explicit @footnote rule in @page

```
@page {  
  size: 120mm 140mm;  
  @footnote {  
    float: bottom page;  
    width: 100%;  
  }  
}
```

- XSL-FO footnote-reference-area implicit in every fo:region-body



There's also differences in how they think about this. In CSS, as I read the spec, you have to define a '@footnote' rule for pages where you want to allow footnotes, whereas in XSL-FO, the 'footnote-reference-area' is implicit in every 'fo:region-body'. If you have a footnote in the region-body area, it will appear in the footnote-reference-area that will just spring to life when there's a footnote to put in there.

CSS to XSL-FO: Footnote counter

- CSS generates numbers on-the-fly

```
::footnote-call {  
  content: counter(footnote);  
  font-size: 8pt;  
  vertical-align: super;  
}  
::footnote-marker {  
  content: counter(footnote);  
  font-size: 8pt;  
  vertical-align: super;  
}
```



But this is also where I swap to the other side, where parts of CSS are reimplemented for XSL-FO.

Something CSS can do is generate footnote numbers on the fly, because they aren't in your HTML, as a rule. So you have counters for keeping track of how many footnotes you have, and you put the counter value in as the footnote marker in the rendered footnote or the footnote call.

CSS to XSL-FO: Footnote counter

- XSL-FO footnote numbers expected to be generated by XSLT
- Can't restart footnote numbers per page
- `<axf:footnote-number>` generates a footnote number
- `<axf:footnote-number-citation>` refers to a footnote
- `axf:footnote-number-reset` restarts at page, odd-page, even-page, column



Whereas, in XSL-FO, the expectation has always been that you generate it with XSLT, and that there's machinery in XSLT, with 'xsl:number', for generating these numbers.

There's never been, for example, a standard facility to restart the footnote numbers on every page, so Antenna House has implemented 'axf:footnote-number' and 'axf:footnote-number-citation' to have the same effect as CSS rules for putting in a number and referring to that number, and Antenna House has implemented an extension so that you can, for example, have the classic sequence of '*', '+', '#', ... restarting per-page.

CSS to XSL-FO: Counter styles

- Sequence of numbers, letters and/or symbols representing a numbering sequence
- CSS 3 counter styles are named algorithms for generating counters for integers
- Used in:
 - `list-style-type` property
 - `counter()` and `counters()` functions

壹	One
貳	Two
參	Three
四	Four



The one that I think is generally useful for internationalization and that you should use in your CSS to begin with is the new CSS counter styles.

Broadly speaking, a counter style is a sequence of letters, numbers, or symbols representing a numbering sequence, which is only ever counting integers.

So the CSS counter styles have names for defined algorithms of sequences of integers, and you would use these in 'list-style-type' for making the markers for your list items, and also in the 'counter()' and 'counters()' functions.

CSS to XSL-FO: Counter styles

```
ol li {  
  list-style-type: japanese-formal;  
}  
...  
<ol>  
  <li title="1">One</li>  
  <li title="2">Two</li>  
  <li title="3">Three</li>  
  <li title="4">Four</li>  
</ol>
```

壱	One
弐	Two
参	Three
四	Four



An example from CSS would be, in this case, using the predefined 'japanese-formal' counter style. Because we're doing that, instead of the literal '1, 2, 3, 4' western digits, we're getting Japanese 'formal' characters (or what I would have described as 'old-style' Japanese) for the numbers.

CSS to XSL-FO: Counter styles

```
<fo:list-block space-before="12pt" space-after="12pt">
  <fo:list-item>
    <fo:list-item-label end-indent="label-end()">
      <fo:block
        axf:number-transform="japanese-formal">1</fo:block>
    </fo:list-item-label>
    <fo:list-item-body start-indent="body-start()">
      <fo:block>One</fo:block>
    </fo:list-item-body>
  </fo:list-item>
  ...
</fo:list-block>
```



This is reimplemented for XSL-FO using a preexisting extension called 'axf:number-transform', but now Antenna House has built in all the pre-defined CSS numbering schemes.

Verbosity of XSL-FO markup

- Transform *your* markup into FO vocabulary

```
<move><board>  
oXX  
oo-  
o-X</board><comment>Yes, we have a  
winner!</comment>  
</move>
```

- XSL-FO was not meant to be serialised
- Highly granular markup allows great control
- Including using a table as list-item label

o	x	x
o	o	
o		x

Yes, we have a winner!

A quick aside on the verbosity of the XSL-FO markup compared to the CSS.

In the paper, I said you could put anything in an 'fo:list-item-label', including a table, so I had to think why on earth you would do that.

The only case that I could think of was to put a tic-tac-toe game inside the 'fo:list-item-label'. The verbosity is not meant to be seen because the design of XSL with XSLT and XSL-FO is that you're supposed to convert your vocabulary into the Formatting Objects vocabulary, and you're never actually supposed to work with XSL-FO as XML. In practice, we tend to generate XSL-FO to feed into the formatter

What's wrong with @format?

- `xs1:number/@format` since XSLT 1.0
- `format` also in XSL since XSL 1.0
- Nine formats defined in XSLT 2.0
- Everything else is *implementation-defined*
- Some implementations provide many
- No help for implementation-*undefined* formats



So if CSS counter styles are the best thing since sliced bread, what's wrong with 'format', which has been in XSLT and XSL-FO since the beginning?

They've always been limited because, even in XSLT 2.0, there's nine defined formats and everything else in implementation-defined. So implementations can provide many more – I work for a company that provides many more in its formatter – but if you go beyond what the implementation defines, then you're stuck.

Define your own counter styles

```
@counter-style my-cjk-decimal {  
  system: numeric;  
  range: 0 infinite;  
  symbols: \3007 \4E00 \4E8C \4E09 \56DB \4E94 \516D \4E03 \516B \4E5D;  
  /* 〇 一 二 三 四 五 六 七 八 九 */  
  suffix: "\3001";  
  /* "、" */  
}
```

```
<fo:declarations>  
  <axf:counter-style name="my-cjk-decimal"  
    system="numeric"  
    range="0 infinite"  
    symbols="〇'一'二'三'四'五'六'七'八'九" suffix="、" />  
</fo:declarations>
```



CSS allows you to define your own counter styles and refer to those in your alternate stylesheet so you can provide numbers for your pages, list items, etc.

A couple of years ago, I was looking at Khmer formatting and I was put in touch with some people in Cambodia, and their gripe at the time was that they couldn't do Khmer numbering in their HTML in their web browser because it just wasn't supported. CSS counter styles does have a counter style for Khmer, but even if it hadn't, they could now do their own counter styles.

So, between counter styles and hyphenation rules, you have a good start on doing halfway-decent formatting in almost any language.

Incompatible differences

- Page selection mechanisms differ between XSL-FO and CSS
- XSL-FO tree contains allowed sequence of page masters
- CSS does some choosing of @page rules
 - Other switching between @page rules at element boundaries



Of course, there's also incompatible differences between the two technologies. The one that I'm going to talk about now is the page selection mechanism.

In XSL-FO, you define your page masters and you define the sequence in which they can occur. In CSS, there's some choosing of the pages, whether it's left or right, etc.

XSL-FO fo:layout-master-set

```
<fo:layout-master-set>
  <fo:simple-page-master master-name="left">
    ...
  </fo:simple-page-master>
  <fo:simple-page-master master-name="right">
    ...
  </fo:simple-page-master>
  <fo:page-sequence-master master-name="psm">
    <fo:repeatable-page-master-alternatives>
      <fo:conditional-page-master-reference
        master-reference="left"
        odd-or-even="even"/>
      <fo:conditional-page-master-reference
        master-reference="right"
        odd-or-even="odd"/>
    </fo:repeatable-page-master-alternatives>
  </fo:page-sequence-master>
</fo:layout-master-set>
```



Your XSL-FO contains 'fo:page-sequence-master', where you can pick alternative page master based on whether it's a left page or right page, etc.

CSS @page rule

```
@page {  
  size: 120mm 140mm;  
  @footnote {  
    float: bottom page;  
    width: 100%;  
  }  
}  
@page :right {  
  @bottom-right {  
    content: counter(page) " of " counter(pages);  
  }  
}  
@page :left {  
  @bottom-left {  
    content: counter(page) " of " counter(pages);  
  }  
}
```



You can do similar in CSS by saying this is the page to use for the left page, the right page, or a blank page, etc.

XSL-FO has more ways to choose

- odd-or-even
- blank-or-not-blank
- page-position: first | last | only | rest | any
- Maximum number of repeats
- Change conditions after set number of pages



XSL-FO has more ways to choose.

You can do things like:

- Set the maximum number of repeats.
- For the first five pages, use this one, then you have a choice for the next ten pages, and a different set of alternatives for the next ten after that.

CSS has different way to choose

```
@page narrow { size: 9cm 18cm }  
@page rotated { size: landscape }  
div { page: narrow }  
table { page: rotated }  
...  
<div>  
  <table>...</table>  
  <table>...</table>  
  <p>This text is rendered on a 'narrow' page</p>  
</div>
```



CSS formats-as-you-go, and it has a rather different mechanism where you can make named pages – similarly to XSL-FO page masters – but the selection of which named page to use depends on the current element that you are formatting.

In this example from the CSS spec, tables are set to use the 'rotated' page, which is in landscape mode, and the paragraph is set to use the 'narrow' page by default. When this is formatted, the tables will appear on one or maybe more landscape pages, then when it got to the paragraph, the formatter would switch to a different named page and use a narrow page for that. Again, this is on-the-fly and less pre-defined than in XSL-FO.

You could probably do the same with Antenna House's nested page sequences.

Summary

- XSL-FO and CSS designed with common core properties
- Minor differences in some properties
- Other properties can be reimplemented in second language
- Some differences can't or needn't be reimplemented in the other language



To conclude, XSL-FO and CSS have a lot of properties in common, by design.

Some properties can be swapped between the two with minor differences, e.g., rename the property and let it have the same values.

Some properties are essentially the same but have different keywords.

Other things may have very different syntaxes, such as the way footnotes are put together, but the underlying implementation can be the same.

With some things, such as page master selection, there are fundamental differences that aren't worth trying to reconcile.

In all of this, the thing that I have left out is "Why are the AH Formatter developers doing this?" If this *was* a marketing talk, I would say that it's "to maximize the holistic synergies of the fundamental technologies", but the real, technical reason is it's because they could.

Thank you.



A Data Usability Company

ANTENNA HOUSE

Questions?

Tony Graham

XML Division

Antenna House, Inc.

tgraham@antenna.co.jp

tony@antennahouse.com

@tgraham_antenna