XSL-FO

Fabio Vitali

Introduzione

Oggi esaminiamo in breve:

- XSLFO, ovvero un vocabolario di elementi che specificano una semantica di formattazione per documenti XML.

CSS (un ricapitolo)

Linguaggio per assegnare proprietaà di formattazione a documenti HTML e XML.

- Non ha meccanismi di manipolazione del contenuto
- Non fissa in maniera definitiva le proprie tà del documento(possono essere cambiate dal lettore)
- Orientato verso visualizzazioni a flusso continuo (paginazione come concetto aggiunto)

Identifica le caratteristiche di presentazione come **ornamenti** dell'albero di partenza (che NON cambia).

- Informazioni stilistiche sono aggiunte ai nodi esistenti
- Posso fare semplici prefissi e suffissi al contenuto dei nodi
- Usa in maniera significativo il concetto di ereditarietà delle proprietà stilistiche.

Supporto per media multipli

- Su display interattivo grafico, display interattivo testuale (terminale VT100 o terminale braille), su supporto cartaceo, su presentatore vocale.

Il W3C spinge per uniformare il modello e il vocabolario di proprietà stilistiche tra i linguaggi a quello di CSS.

XSL (un ricapitolo)

Composto da due linguaggi:

- XSLT esegue trasformazioni attraverso regole. Qualunque linguaggio di destinazione è accettabile, perché dipende dall'applicazione che ne fa uso: un browser HTML vuole HTML, un browser WAP vuole WML, un browser XSLFO vuole XSLFO).
- XSL-FO (anche noto come XSL e basta): linguaggio di formattazione basato sulla paginazione di documenti di flusso. Permette costrutti di formattazione device-independent: è sempre possibile specificare due fogli di stile per due media diversi (es. carta e schermo), ma comunque il risultato nell'usarne uno solo può essere comunque reso accettabile.
- Distingue tra proprietà visive e uditive (aurali)

Evoluzione di XSL

- 27 agosto 1997: prima nota del W3C che stabilisce la filosofia generale del linguaggio.
- 18 agosto 1998: primo Working Draft che separa nettamente la fase di trasformazione dalla fase di visualizzazione. Cambia la sintassi. Sono introdotti i namespace.
- 21 aprile 1999: separazione di XSL in due Draft distinti XSLT e XSL-FO.
- Situazione attuale: Entrambi recommendation. Di XSLT esistono decine di implementazioni, di XSL-FO alcune per la creazione di documenti cartacei, solo una per la visualizzazione a schermo.

Alcune distinzioni

Rendering a layout o a flusso

- Rendering a layout: la formattazione rispetta le caratteristiche fisiche del medium ponendo dei limiti sulla quantità o l'aspetto delle informazioni da presentare
- Rendering a flusso: la formattazione rispetta le caratteristiche fisiche del contenuto generando quanto medium serve (nuove pagine o scroll più lunghi)

Paginazione o scrolling

- Lo scrolling è basato sul concetto di canvas infinito (o forse finito in una direzione sola), all'interno del quale le aree di visualizzazione si presentano in sequenza senza interruzioni. La presentazione uditiva è in un certo senso a scrolling
- La paginazione introduce dimensioni assolute e non modificabili (la pagina) e la possibilità di specificare regioni fisse e ripetibili (logo, intestazione, piè pagina, ecc.). Analogo al caso dei frame, più o meno.

XSL-FO

Scopi del linguaggio:

- definire la fase di formattazione.
- definire un vocabolario di elementi di formattazione indipendenti dal tipo di supporto utilizzato per l'output.

Il legame tra le due definizioni è ovviamente molto stretto in quanto la fase di formattazione interpreta l'albero che risulta dall'eventuale trasformazione in base alla semantica degli oggetti di formattazione che lo costituiscono.

Introduzione alla formattazione

Nell'output su supporti visuali per astrarre dal tipo di scrittura usato (ad esempio occidentale o orientale) si introduce il concetto di writing-mode che definisce:

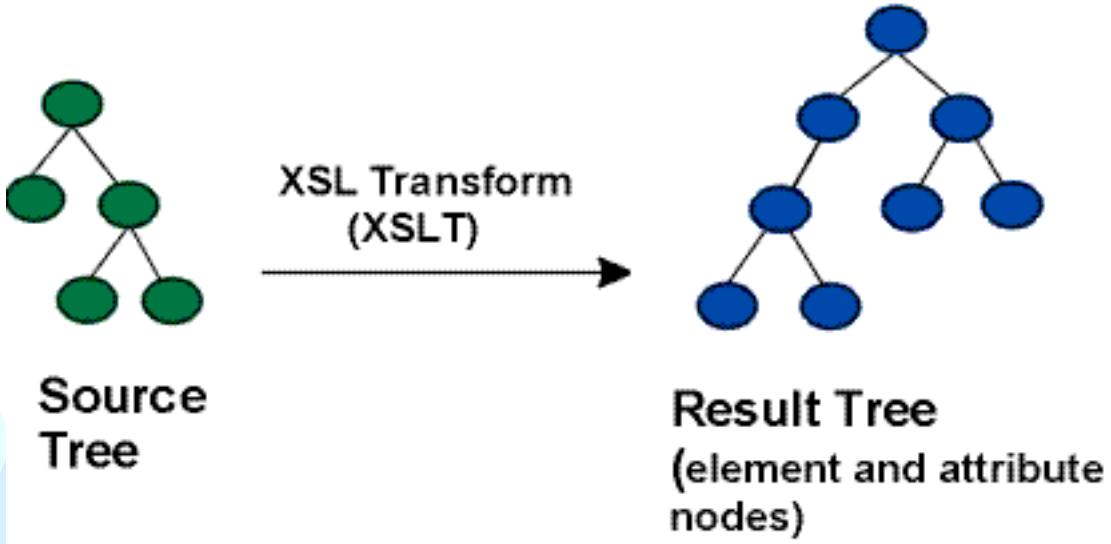
- Direzioni relative (block- e inline- progression-direction)
- Riferimenti relativi (before, after, start o end)

Fasi della formattazione

La fase di formattazione è schematizzabile in cinque passi:

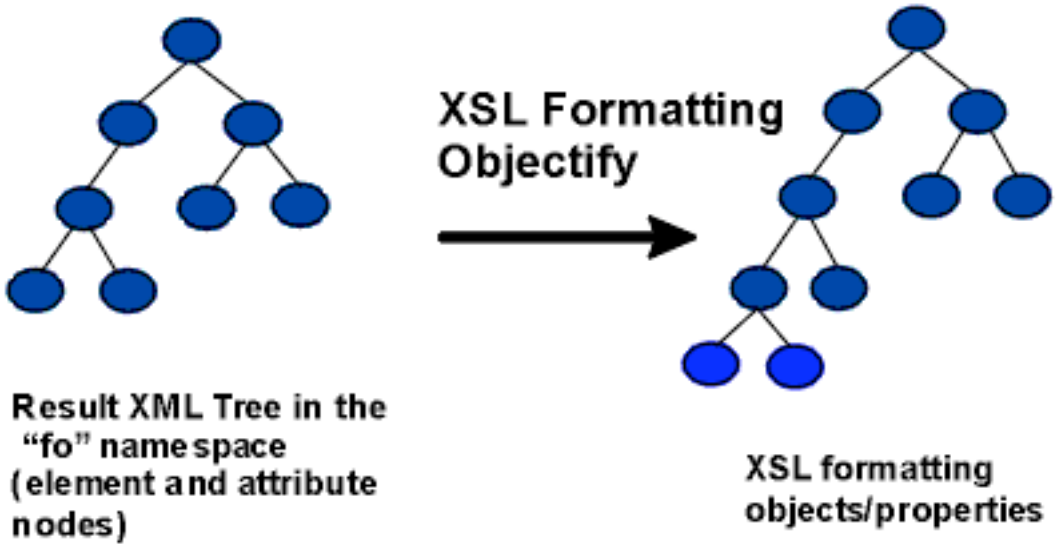
1. Trasformazione del documento XML attraverso un documento XSLT in un documento XSLFO
2. Trasformazione dell'albero in uno costituito, non da elementi e attributi, ma da oggetti di formattazione e loro proprietà.
3. Raffinamento dell'albero degli oggetti di formattazione ovvero mapping dalle proprietà nelle caratteristiche. Scioglimento dei valori relativi, ereditati, calcolati, raggruppati
4. Costruzione dell'albero delle aree. Identificazione degli elementi ripetuti, fissi, ecc.
5. Rendering finale

1: Generazione del documento XSLFO



- La trasformazione cambia completamente il modello dell'albero, generando nuovi oggetti e anche nuovo contenuto (ad esempio elementi ripetuti o fissi).
- La forma dell'albero è completamente diversa dal documento di partenza.

2: Costruzione dell'albero dei FO



- Vengono generati gli elementi di formattazione.
- Ogni elemento genera un oggetto, ogni attributo genera un proprietà dell'albero.
- Identificazione degli elementi del namespace FO e mantenimento degli altri.

3 - Raffinamento dell'albero

Valori specificati, calcolati e applicabili.

- Valore in %
- Valore assoluto
- Valore espresso in unità di misura utilizzate dall'output (es. pixel)

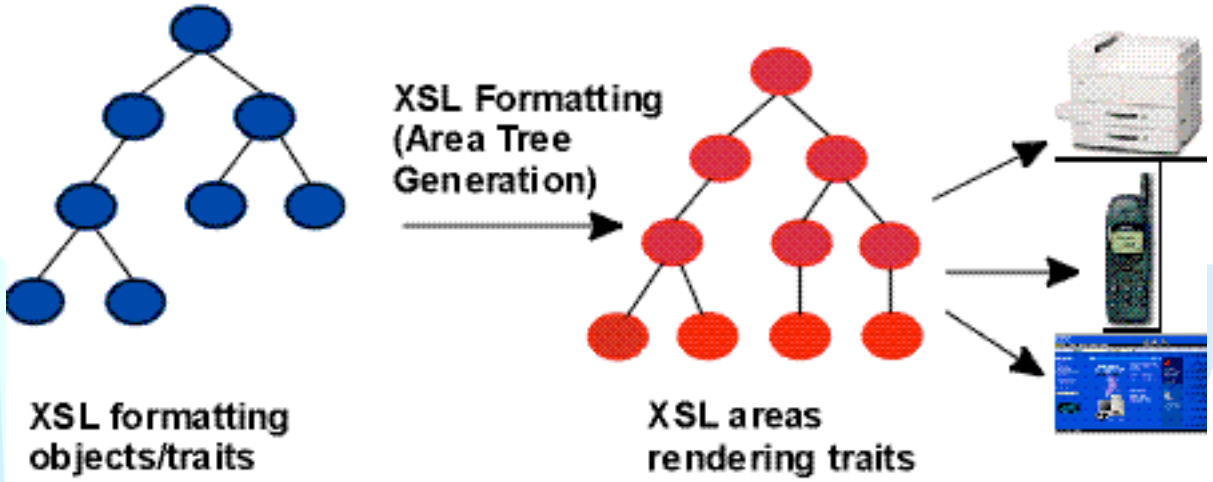
Proprietà espresse in forma breve

- Border: "`10px solid blue`"

Mapping di proprietà relative

- Border-before = border-top se writing-mode è lr-tb

4-5 Albero delle aree e rendering



- Vengono generati gli oggetti di visualizzazione con le dimensioni definitive
- Avviene la paginazione con la moltiplicazione degli oggetti ripetuti, ecc.
- L'oggetto viene visualizzato sul device di output specificato.

Un esempio

```
<?xml version="1.0"?>
<root xmlns="http://www.w3.org/1999/XSL/Format"
      font-size="16pt">
  <layout-master-set>
    <simple-page-master master-name="page"
                page-height="297mm" page-width="210mm"
                margin-top="15mm" margin-bottom="15mm"
                margin-left=15mm" margin-right="15mm">
      <region-body region-name="body"
                margin-top="5mm" margin-bottom="5mm"/>
    </simple-page-master>
  </layout-master-set>
  <page-sequence master-reference="page">
    <title>Ciao mondo</title>
    <flow flow-name="body">
      <block>Ciao mondo!</block>
    </flow>
  </page-sequence>
</root>
```

Modello delle aree

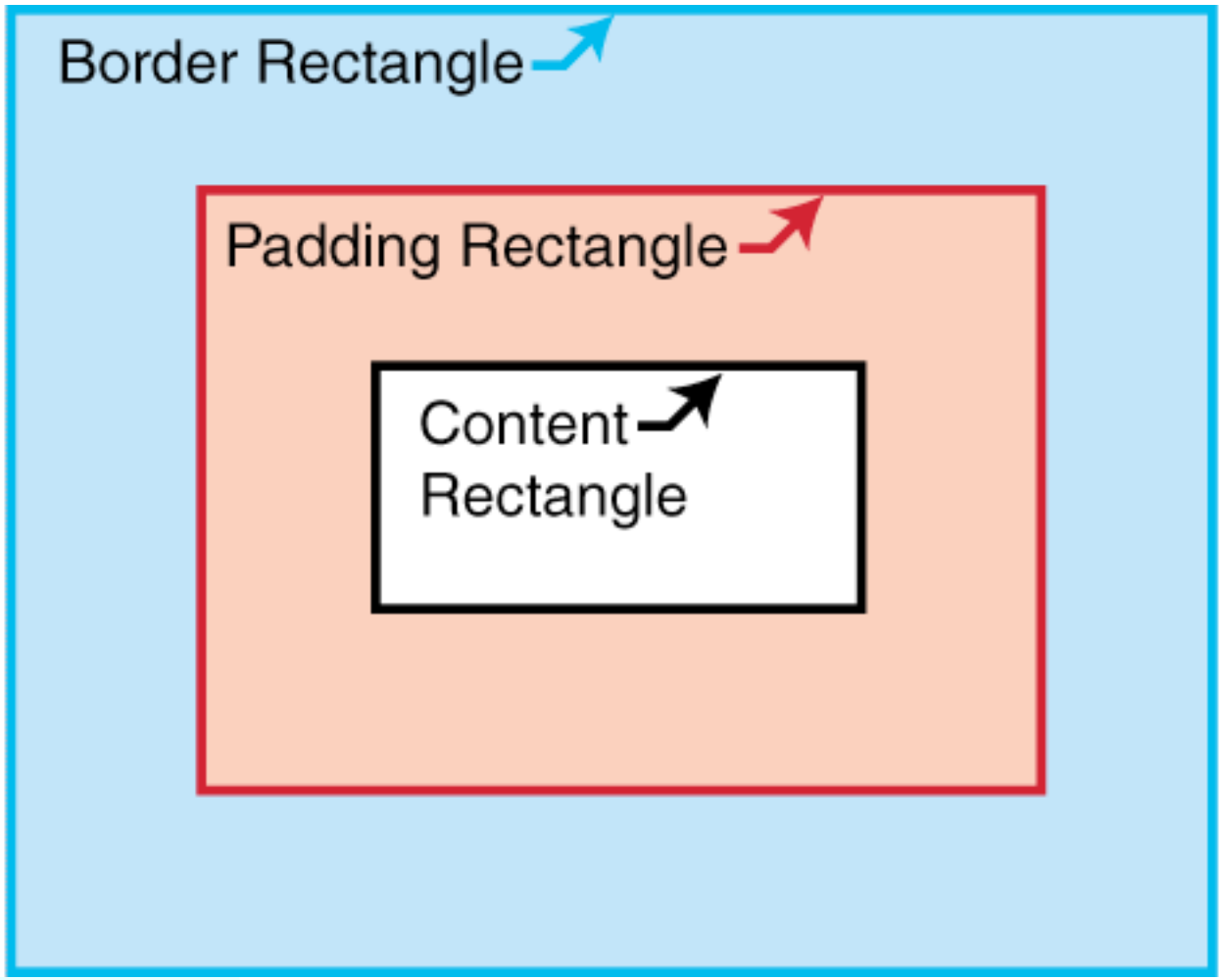
Il modello delle aree definisce aree rettangolari e spazi tra aree. Le aree rettangolari, generate dagli oggetti di formattazione, riservano spazio e racchiudono contenuto. Gli spazi riservano spazio prima e dopo le aree rettangolari e non hanno contenuto.

Aree Rettangolari

- Block-area
- Inline-area.

La loro differenza principale sta nel modo in cui normalmente il formatter le posiziona. Esse vengono tipicamente posizionate seguendo rispettivamente la block-progression-direction e la inline-progression-direction, in alcuni casi però è possibile che siano posizionate esplicitamente (ad esempio in base alle absolute-position-properties).

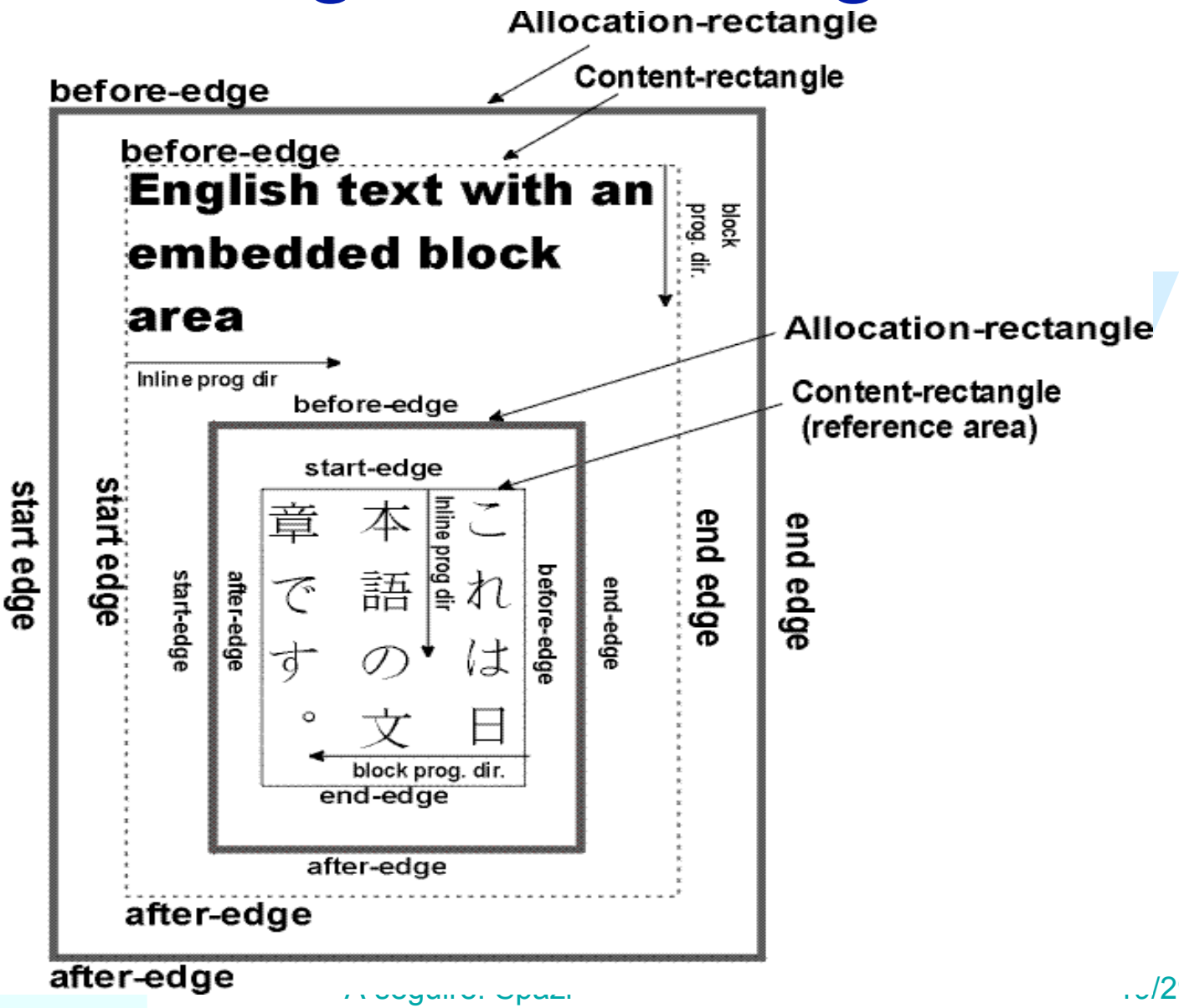
Aree Rettangolari, caratteristiche comuni



Caratteristiche comuni (2)

- Content rectangle.
- Padding Rectangle.
- Border Rectangle.
- Allocation Rectangle: definisce la dimensione usata per allocare spazio quando l'area viene posizionata nell'area padre.

Aree Rettangolari e writing-mode



Spazi

Gli spazi riservano spazio prima e dopo le aree rettangolari e non hanno contenuto. Essi vengono definiti attraverso un tipo di dato composto dalle seguenti informazioni:

- Space.minimum
- Space.maximum
- Space.optimum
- Space.conditionality
- Space.precedence

In generale gli stessi formatting-object che generano aree rettangolari generano gli spazi ad esse associati.

Viewport

Ci sono situazioni in cui il contenuto di un'area non viene completamente visualizzato. Questo può accadere se tale contenuto genera un overflow e l'area che lo contiene ha la proprietà overflow uguale a hidden. Un altro caso può verificarsi se il designer riduce la zona visibile di un'area attraverso la proprietà clip. Date queste situazioni si definisce viewport la zona visibile di un area, ad esempio page-viewport-area o block-viewport-area.

I formatting-object

Ci sono tre tipi di formatting-object: quelli che generano aree, quelli che restituiscono aree modificandone eventualmente le caratteristiche e quelli usati da altri formatting-object per generare aree. I primi due tipi sono comunemente chiamati flow-object , il terzo tipo può essere detto layout-object o auxiliary-object.

FO di impaginazione e layout

Permettono di definire sia la struttura di layout di una pagina o frame (dimensioni e posizione del body e sua eventuale suddivisione in colonne, header, footer, side-bar) sia le regole attraverso cui il contenuto di partenza è sistemato in questi contenitori attraverso:

- fo:simple-page-master
- fo:page-sequence

Block-level FO

- Sono generalmente usati per la formattazione di titoli, paragrafi, didascalie di immagini, tabelle o liste.
- Generano block-area. Ad esempio:

```
<fo:block text-align="center"
      space-after="8pt"
      space-before="16pt"
      space-after.precedence="3">
    Contenuto dell'area
 </fo:block>
```

Inline-level FO

- Il loro utilizzo più comune è per la gestione di immagini, la formattazione di elementi all'interno di linee come ad esempio assegnare bordi colorati a caratteri o parole e la formattazione di elementi di link.

```
<fo:inline font-style="italic">
     Il contenuto dell'elemento
</fo:inline>
```

Altri FO

- fo:wrapper

Permette di attribuire un set di proprietà a tutti i suoi elementi figli.

- Utilizzo classico:

```
<fo:wrapper font-style="italic">
    Testo in stile corsivo
</fo:wrapper>
```

Proprietà dei FO (1)

Le proprietà di XSLFO sono in buona parte uguali a quelle di CSS (per volontà del W3C, che vuole uniformare concetti e vocabolari fin dove possibile). Quindi ogni proprietà XSLFO (un attributo) può essere uno di:

- Una proprietà CSS2 non modificata
- Una proprietà CSS2 con valori in un set esteso
- Una proprietà CSS2 modificata o estesa in ambito
- Una proprietà solo XSLFO

Le proprietà possono essere caratterizzate nel seguente modo:

- Posizionamento assoluto (top, bottom, left, right)
- Proprietà uditive (pause, pitch, stress, voice-family, ecc)
- Proprietà di border, padding e background
- Proprietà di font (font-family, font-size, font-style, font-weight, ecc.)

A seguire: Proprietà dei FO (2)

Proprietà dei FO (2)

- Proprietà di sillabazione (dipendenti dal linguaggio)
- Proprietà di margine per blocchi e inline
- Posizionamento relativo
- Proprietà di area
- Proprietà di blocchi e inline
- Proprietà di carattere
- Colori
- Elementi Float, keep e break (inclusi orfani e vedove)
- Proprietà per effetti dinamici
- Proprietà di paginazione e layout
- Proprietà di tabelle
- Altre proprietà miste

Riferimenti

- S. Adler et alii., *Extensible Stylesheet Language (XSL) Version 1.0*, W3C Recommendation 15 October 2001, http://www.w3.org/TR/xsl/
- Crane Softwrights, *Practical Formatting Using XSLFO*, Second Edition - ISBN 1-894049-09-8, 2002-04-05, http://www.CraneSoftwrights.com/training/pfux/pfux-20020405-prev-a4-dbl-pdf.zip