

CURS 12

XSL(Extensible Stylesheet Language)

3. XSL-FO (Extensible Stylesheet Language -Formatting Object)

XSL-FO datează din 2001 și aparține consorțiului W3C. Se folosește versiunea 1.0, iar specificația oficială se găsește la adresa www.w3.org/TR/xsl.

XSL-FO reprezintă un limbaj de tag-uri bazat pe specificațiile XML și folosit pentru a descrie regulile de formatare a datelor din documentul XML, cu scopul de a le afișa sau tipări.

XSLT reprezintă un „vocabular de transformare” a fișierelor XML în alte fișiere.

XSL-FO reprezintă un „vocabular de prezentare” a datelor din fișierele XML.

Spațiul de nume pentru XSL-FO se află la adresa <http://www.w3c.org/1999/XSL/Format>. Acesta este referit prin prefixul .fo sau .fob.

1.Structura fișierului

```
1. <?xml version = "1.0" encoding="ISO-8859-1" ?>
2. <fo:root xmlns:fo="http://www.w3org/1999/XSL/format">
3.     <fo:layout-master-set>
4.         <!--Aici vin toate paginile tipar-->
5.         <fo:simple-page-master master-name="nume">
6.             <!--Aici vine o singura pagina tipar-->
7.             </ fo:simple-page-master>
8.             <!--Aici vin secvențele de pagini tipar -->
9.             <fo:page-sequence-master master-name="nume">
10.                .....
11.                </fo:page-sequence-master>
12.            </ fo:layout-master-set>
13.            <!--Aici vin declaratiile globale pentru foi de stiluri-->
14.            < fo :declarations>
15.                .....
16.            </fo :declarations>
17.            <fo :page-sequence master-reference= " name">
18.                <!--Aici vine continutul paginii-->
19.            </ fo :page-sequence >
20.            .....
21.        </fo:root>
```

1. - Se referă la declarația XML
2. - Elementul fo:root reprezintă elementul rădăcină a unui document XSL-FO.
 - Declararea spațiului de nume folosit la documentele XSL-FO se face prin intermediul atributului xmlns. Acest element nu influențează în nici un fel modul în care se realizează

formatarea. El acceptă un singur atribut opțional numit `media-usage`, folosit pentru a indica modul de prezentare al paginii rezultat.

El are 4 valori:

- `auto`: valoarea implicită; XSL-FO face alegerea potrivită;
- `paginate`: din elementul `fo:page-sequence` se va genera o secvență de pagini rezultat;
- `bounded-in-one-dimension`: se generează o singură pagină rezultat pentru fiecare element `fo:page-sequence`;
- `unbounded`: se generează o singură pagină rezultat pentru fiecare element `fo:page-sequence` nespecificând attributele `page-width` și `page-height`;

3. Elementul `fo:layout-master-set` conține una sau mai multe declarații de pagini tipar și/sau una sau mai multe declarații de secvențe-de-pagini-tipar. Acest element este obligatoriu și poate apărea o singură dată în cadrul fișierului.

4. Elementul `fo:simple-page-master`.

Formatarea fișierului XML folosind limbajul XSL-FO se face pe baza așa numitelor pagini tipar. Aceste pagini se obțin pe baza paginilor master.

XSL-FO definește un singur tip de pagină master definit de o suprafață rectangulară, care poate fi personalizată prin intermediul `fo:simple-page-master`.

Putem percepe pagina master ca pe o matrice cu caracteristici ajustabile și pe baza lor se pot crea mai multe pagini tipar.

Cel care scrie documentul XSL-FO poate defini oricâte pagini tipar, prin folosirea repetată a `fo:simple-page-master`. Fiecare pagină tipar e identificată printr-un nume unic ce reprezintă valoarea atributului `master-name`.

5. Elementul `fo:page-sequence-master`.

Dacă există mai multe pagini tipar acestea se pot grupa în secvențe de pagini tipar, prin intermediul acestui element. Folosirea acestui element e opțională.

6. Elementul `fo:declarations` este folosit la gruparea declarațiilor globale pentru o foaie de stiluri. Acesta este opțional și poate apărea o singură dată.

7. Elementul `fo:page-sequence`. Pentru a defini paginile documentului rezultat, se apelează la unul sau mai multe elemente `fo:page-sequence` (cel puțin unul e obligatoriu).

Paginile documentului rezultat sunt grupate în secvențe de pagini rezultat și fiecare secvență, începe pe o pagină rezultat nou.

Fiecare pagină rezultat din secvență, are asociată o pagină tipar. Specificarea paginii se face prin setarea atributului `master-reference`.

2. Afișarea rezultatului procesării unui fișier XSL-FO

Pentru a afișa rezultatul procesării se apelează la suprafețe rectangulare numite cutii (boxes). Orice tip de conținut: text, imagini, tabele, liste, este formatat și afișat prin intermediu acestor cutii. Cele mai cunoscute cutii sunt:

După procesarea unui document XSL-FO se obține un document numit document rezultat, format dintr-un set de pagini, numite pagini rezultat.

Pagina este considerată cea mai mare cutie, restul fiind încorporată în ea.

³⁵₁₇ Regiuni (regions)

Regiunile sunt suprafețele rectangulare principale. Fiecare pagină rezultat are 5 astfel de regiuni reprezentate de elementele următoare:

- fo:region-body
- fo:region-start
- fo:region-end
- fo:region-before
- fo:region-after

³⁵₁₇ Blocurile de conținut (block areas)

Sunt suprafețe rectangulare care pot conține: text, imagini, linii de conținut , suprafețe-inline, blocuri de conținut. Acestea sunt distribuite secvențial, nu sunt plasate la coordonate specificate. Fiecare bloc de conținut este caracterizat de 3 straturi ce înconjoară acest conținut.

³⁵₁₇ Linie de conținut (line areas)

Sunt suprafețe rectangulare ce se găsesc în blocurile de conținut și pot conține suprafețe-inline și spații. Acestea sunt generate automat, nu exista elemente pentru a fi definite.

³⁵₁₇ Suprafețe-inline (inline areas)

Suprafețele inline sunt suprafețe rectangulare ce se succed în interiorul liniilor de conținut și care pot conține: un singur caracter, o ecuație sau alte suprafețe in-line.

³⁵₁₇ Suprafete-caracter

Acestea sunt cele mai cunoscute si utilizatr suprafete-inline si reprezinta de fapt, caractere singulare.

Observatie!

Apare fenomenul de moștenire a valorilor atributelor. Există attribute pentru:

- tipul de scriere (writing mode)
- tipul de indentare
- tipul de space before
- tipul de space after