

# Εκτύπωση από την XML: Εισαγωγή στο XSL-FO

---

Dave Pawson

*Μετάφραση από τον Βασίλειο Τσάγκαλο  
με την άδεια του συγγραφέα.*

*Το πρωτότυπο κείμενο βρίσκεται στον παρακάτω δικτυακό τόπο*

<http://www.xml.com/pub/a/2002/10/09/xslfo.html>

## 1. Εισαγωγή

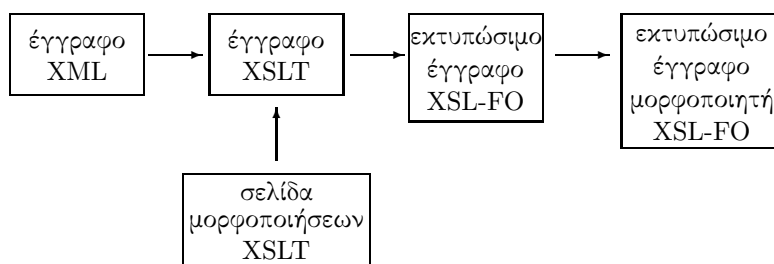
Ένα από τα προβλήματα που αντιμετωπίζουν πολλοί χρήστες όταν ξεκινούν την παραγωγή εκτύπωσης από την XML είναι το πρόβλημα της διάταξης σελίδας. Αν δεν ρυθμιστεί σωστά η διάταξη σελίδας, δεν έχετε πιθανότητες να προχωρήσετε. Προτείνοντας τη λύση W3C XSL Formatting Objects, θέλω να παρουσιάσω μία απλοποιημένη προσέγγιση που θα δώσει στο νέο χρήστη τη δυνατότητα να πατήσει σε στέρεο έδαφος στο θέμα της διάταξης σελίδας.

Ο στόχος αυτού του άρθρου είναι να δημιουργήσουμε την πρώτη εκείνη σελίδα εξόδου - ας την ονομάσουμε πρόγραμμα "Hello World" - με αρκετές πληροφορίες που επιτρέπουν σε κάποιο χρήστη να προχωρήσει σε πιο χρήσιμα πράγματα. Θα παρουσιάσω την πιο απλή από τις διατάξεις σελίδας για XSL-FO, χρησιμοποιώντας όσο το δυνατόν λιγότερα στοιχεία μπορώ από τα απαιτούμενα, ώστε να εξασφαλίσω μία εύλογη έξοδο.

Ένα από τα προβλήματα είναι ότι, σε αντίθεση με την παραγωγή ενός εγγράφου HTML από πηγή XML με χρήση της XSLT, η επεξεργασία των απογόνων των ριζικών στοιχείων δεν είναι μία απλή `xsl:apply-templates` από εντός του ριζικού στοιχείου. Απαιτείται πολύ περισσότερη αρχική έξοδος προκειμένου να μπορέσει ο μορφοποιητής να παράξει τις σελίδες.

Ας δούμε την απαραίτητη επεξεργασία για να μεταβείτε από το έγγραφο XML σε εκτυλώσιμο έγγραφο PDF. Κατ' αρχήν, η XML πρέπει να τροφοδοτείται σε επεξεργαστή XSLT με μία κατάλληλη σελίδα μορφοποιήσεων (που αναπτύσσεται παρακάτω) προκειμένου να παραχθεί ένα άλλο έγγραφο XML που χρησιμοποιεί το διάστημα ονόματος (name space) XSL-FO και προορίζεται για το μορφοποιητή XSL-FO. Το δεύτερο στάδιο είναι η τροφοδότηση της εξόδου του πρώτου σταδίου

στο μορφοποιητή XSL-FO, που μπορεί στη συνέχεια να παράξει το τελικό προϊόν: ένα εκτυπώσιμο έγγραφο, διαμορφωμένο για οπτική παρουσίαση.



Η προσέγγιση αυτή έχει το πλεονέκτημα ότι το έγγραφο πηγής XML είναι ακόμη μορφοουδέτερο και μπορεί να χρησιμοποιηθεί με άλλες σελίδες μορφοποιήσεων XSLT για την παραγωγή άλλων μέσων.

## 2. Το έγγραφο XSL-FO

Πρέπει να γνωρίζουμε για τον αρχικό στόχο της μετατροπής του XSLT, το έγγραφο XSL-FO. Το έγγραφο που παράγετε, το οποίο τροφοδοτείται στο μορφοποιητή XSL-FO, περιέχει ένα μικρό αριθμό στοιχείων:

```

<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <fo:layout-master-set> [1]
    <fo:simple-page-master
      master-name="simple" > [2]
      <fo:region-body/>
    </fo:simple-page-master>
  </fo:layout-master-set>
  <fo:page-sequence
    master-reference="simple"> [3]
    <fo:flow
      flow-name="xsl-region-body"> [4]
      content [5]
    </fo:flow>
  </fo:page-sequence>
</fo:root>
  
```

Ας δούμε το καθένα από τα αριθμημένα στοιχεία με τη σειρά.

- [1] Για να διαταχθεί το περιεχόμενο σε μια σελίδα, ο μορφοποιητής πρέπει να γνωρίζει με ποια μεγέθη έχει να κάνει. Το `layout-master-set` περιέχει το [2] `simple-page-master` που περιέχει αυτές τις πληροφορίες, δηλ. αν χρησιμοποιείτε μέγεθος σελίδας A4 ή μέγεθος US-letter. Περιέχει επίσης το στοιχείο `region-body`, το οποίο μπορεί να θεωρηθεί ως το κυρίως σώμα της διάταξης σελίδας.
- [3] Για να υποστηριχτεί η περίπλοκη σελιδοποίηση, χρησιμοποιείται το στοιχείο `page-sequence`. Για διάταξη μιας απλής σελίδας, απαιτείται πολύ μικρό περιεχόμενο, διαφορετική από την αναφορά σε ορισμό συγκεκριμένης σελίδας (το `simple-page-master`).
- [4] Επίσης, εντός του στοιχείου `page-sequence` υπάρχει ένα στοιχείο ροής (flow element). Η ιδέα της ροής μπορεί και να μην σας είναι οικεία. Τη συνάντησα όταν χρησιμοποίησα λογισμικά επιτραπέζιας τυπογραφίας, όταν έπρεπε να ρίξω κείμενο μέσα σε περιοχές σελίδας για τη δημιουργία στηλών σε κάποιο κολεγιακό περιοδικό, οπότε το περιεχόμενο έρεε μέσα σε περιοχές της σελίδας.
- [5] Ο εντοπισμός της περιοχής της σελίδας στην οποία πρέπει να ρίξουμε το κείμενο είναι και το σκεπτικό για το `xsl-region-body`. Αυτό διαφοροποιεί το σώμα της σελίδας από τις εξωτερικές περιοχές (περιθώρια, κεφαλίδα, υποσέλιδο, κλπ.) της σελίδας. Τέλος, κάποιο περιεχόμενο, το οποίο είναι απόγονος της κύριας ροής. Δεν μπορεί να εισαχθεί απλό κείμενο εδώ, επειδή ο μορφοποιητής θα έπρεπε να μαντέψει τι θα θέλατε να το κάνει, επομένως το πραγματικό περιεχόμενο για τη ροή θα έπαιρνε τη μορφή

```
<fo:block>περιεχόμενο</fo:block>
```

που ορίζει ένα μπλοκ κειμένου (ορθογώνιου σχήματος, όσο μεγάλο θέλετε, λαμβάνοντας πλήρη λίστα των προκαθορισμένων τιμών για το καθετί) που θα τοποθετηθεί ως πρώτο στοιχείο επάνω στη σελίδα.

Για να τα καταλάβετε καλύτερα όλα αυτά, ας συμπληρώσουμε, στο ελάχιστο δυνατό, ό,τι θα έμπαινε στη σελίδα μορφοποιήσεων με σκοπό να λάβει ένα απλό έγγραφο XML και να παραγάγει ένα άλλο έγγραφο XML, το οποίο στη συνέχεια τροφοδοτείται σε μορφοποιητή XSL-FO.

Δείτε παρακάτω μία βασική σελίδα μορφοποιήσεων XSLT για παραγωγή XSL-FO:

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform" [1]
```

```

xmlns:fo="http://www.w3.org/1999/XSL/Format"> [2]
version="1.0">
<xsl:output method="xml"/> [3]

<xsl:template match="/">
    . . . . [4]
</xsl:template>

Other prototypes [5]
</xsl:stylesheet>

```

Στο [1] και [2] βλέπουμε τα διαστήματα ονομάτων, αντίστοιχα, του XSLT και του περιεχομένου FO σε αυτό το έγγραφο, που διαφοροποιεί τις αιτήσεις μετασχηματισμού από το περιεχόμενο εξόδου.

Αν η μηχανή XSLT δει περιεχόμενο στο διάστημα ονόματος FO, απλά το γράφει στην έξοδο, που είναι ακριβώς ό,τι θέλουμε. Το [3] λέει ότι θέλουμε το έγγραφο εξόδου να είναι ένα έγκυρο XML, που είναι ακριβώς ό,τι είναι ένα έγγραφο XSL-FO, ένα έγγραφο XML. Το [4] είναι το ριζικό πρότυπο, που ενεργοποιείται πρώτο, οπότε αυτό είναι το σημείο στο οποίο προσθέτουμε το ουσιαστικό περιεχόμενο περιγράμματος που αναφέραμε παραπάνω.

Τέλος, στο [5], μπορούμε να ξεκινήσουμε να προσθέτουμε χρήσιμες επεξεργασίες. Μπορούμε τώρα να συνδυάσουμε τα δύο παραπάνω κομματάκια για να κάνουν κάτι χρήσιμο. Παρακάτω έχουμε μία πλήρη σελίδα μορφοποιήσεων XSLT, η οποία χρησιμοποιείται από τη μηχανή XSLT για την παραγωγή ενός έγκυρου εγγράφου XSL-FO.

```

<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fo="http://www.w3.org/1999/XSL/Format">
  version="1.0">
  <xsl:output method="xml"/>

  <xsl:template match="/">
  <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
    <fo:layout-master-set>
      <fo:simple-page-master
        master-name="simple"
          page-height = "29.7cm" [1]
          page-width  = "21cm"
          margin-left  = "2.5cm"

```

```
margin-right = "2.5cm">
  <fo:region-body margin-top="3cm"/> [2]
</fo:simple-page-master>
</fo:layout-master-set>
<fo:page-sequence
  master-reference="simple"> [3]
  <fo:flow
    flow-name="xsl-region-body"> [4]
    <xsl:apply-templates/> [5]
  </fo:flow>
</fo:page-sequence>
</fo:root>
</xsl:template>

<xsl:template match="document"> [6]
  <fo:block>
  <xsl:apply-templates/>
  </fo:block>
</xsl:template>

<xsl:template match="head"> [7]
  <fo:block>
  <xsl:apply-templates/>
  </fo:block>
</xsl:template>

<xsl:template match="para"> [8]
  <fo:block>
  <xsl:apply-templates/>
  </fo:block>
</xsl:template>

<xsl:template match="em"> [9]
  <fo:inline font-style="italic">
  <xsl:apply-templates/>
  </fo:inline>
</xsl:template>

<xsl:template match="*"> [10]
  <fo:block background-color="red">
```

```

<xsl:apply-templates/>
</fo:block>
</xsl:template>

```

```
</xsl:stylesheet>
```

Πριν εξηγήσουμε τη δομή, θα πρέπει να αναφέρουμε το έγγραφο πηγής για το οποίο σχεδιάζουμε αυτή τη σελίδα μορφοποιήσεων. Ξεκινώ με τροφοδοσία από μία κατηγορία εγγράφου που έχει 4 στοιχεία, με τη δομή που φαίνεται παρακάτω. Τη διατηρώ απλή διότι αναπαριστά τη μεγάλη πλειοψηφία περιεχομένου XML που προορίζεται για ένα έγγραφο XSL-FO. Περιέχει μόνο δύο στοιχεία μπλοκ (`head` και `para`) και ένα εσωτερικό στοιχείο (`em`).

Το έγγραφό μας περιέχεται σε ένα εξωτερικό στοιχείο εγγράφου, καθώς και σε ένα μείγμα στοιχείων `head` και `para` που περιλαμβάνουν κάποια έμφαση:

```

<document>
  <head>My very first xsl-fo document</head>
  <para>has an <em>important</em> paragraph inside it</para>
</document>

```

Το μέγεθος σελίδας καθορίζεται στο [1], με χρήση ευρωπαϊκών διαστάσεων. Αλλάξτε τις βάζοντας το δικό σας μέγεθος χαρτιού αν είναι διαφορετικό. Έχω προσθέσει περιθώρια επειδή το περιεχόμενο που εκτείνεται προς τις άκρες της σελίδας δεν είναι ευπαρουσίαστο.

Στο [2] έχω προσθέσει ένα άνω περιθώριο στην κύρια περιοχή της σελίδας. Το [3] και [4] είναι όπως πριν. Στο [5] έχουμε μία καθοριστική διαφορά: στο σημείο αυτό, όπου πριν απλώς είπα «περιεχόμενο», χρησιμοποιώ τώρα τις δυνατότητες της XSLT για να καθοδηγήσω τη μηχανή XSLT να επεξεργαστεί το έγγραφο εισόδου. Στο [6] η μηχανή XSLT επεξεργάζεται το στοιχείο εγγράφου του αρχείου εισόδου XML με παραγωγή εξόδου ενός στοιχείου `fo:block`, εντός του οποίου τοποθετείται όλο το υπόλοιπο περιεχόμενο. Επειδή τα μπλοκ μπορούν να εμπερικλείονται αρκετά εύκολα στο XSL-FO, αυτό δεν είναι πρόβλημα. Αυτό που πραγματικά κάνει, είναι να εξασφαλίζει τη διατήρηση του περιεχομένου που ξεχειλίζει -δηλαδή, δεν το χειρίζεται σαφώς η σελίδα μορφοποιήσεων - εντός κάποιου μπλοκ.

Στα [7], [8] και [9] επιστρέφω στον κανονικό κόσμο της XML και XSLT. Η αντιστοίχιση ενός στοιχείου του εγγράφου πηγής και η παραγωγή εξόδου ενός κατάλληλου στοιχείου από το λεξιλόγιο του XSL-FO. Τα πρώτα δύο είναι ταυτόσημα και απλώς χρειάζονται διακόσμηση, το τελευταίο είναι ελαφρώς διαφορετικό κατά το ότι είναι ένα αντικείμενο εσωτερικής μορφοποίησης και παράγει έξοδο πλάγιας γραφής.

Το [10] είναι ο σάκος με τα πάντα που δείχνει (στην έξοδο) ποια στοιχεία, αν υπάρχουν, δεν έχουν υπαχθεί σε μορφοποίηση. Από τη στιγμή που θα εφαρμοστεί μορφοποίηση σε όλα τα στοιχεία, τίποτε δεν θα υποστεί επεξεργασία από αυτό το πρότυπο. Είναι καλό ως δυνατότητα εκσφαλμάτωσης κατά τη διάρκεια της ανάπτυξης.

Αυτή η σελίδα μορφοποιήσεων εισάγει δύο νέα στοιχεία. Το πρώτο είναι το στοιχείο `fo:block`, που χρησιμοποιείται για πολλά στοιχεία στη σελίδα μορφοποιήσεων. Αυτό είναι το βασικό στοιχείο διάταξης που χρησιμοποιείται για αναδίπλωση περιεχομένου -σχεφτείτε το ως στοιχείο `<p>` σε HTML.

Το στοιχείο `fo:inline` είναι ένα σύνολο εγκλεισμού για εσωτερικά στοιχεία σε XSL-FO. Το καθένα από τα δύο αυτά στοιχεία έχει ένα ολόκληρο φάσμα ιδιοτήτων, εκφρασμένων συντακτικά ως ιδιοχαρακτηριστικών, που χρησιμοποιούνται για τη διακόσμηση του περιεχομένου που αναδιπλώνουν.

### 3. Έναρξη νέων σελίδων

Ας επεκτείνουμε τη δομή του εγγράφου πηγής για να συμπεριλάβει μία ενότητα που θα πρέπει να έχει σημείο έναρξης νέας σελίδας. Επομένως, το έγγραφο τώρα μπορεί να έχει την εξής εμφάνιση:

```
<document>
  <section>
    <head>My very first xsl-fo document</head>
    <para>has an <em>important</em> paragraph inside it</para>
  </section>
  <section>
    <head>The second section, starting on a new page </head>
    <para>Some content in the second section</para>
  </section>
</document>
```

Τώρα χρειάζεται να δώσουμε στυλ σε αυτήν την προσθήκη, χρησιμοποιώντας μία από τις διαθέσιμες ιδιότητες ενός μπλοκ.

```
<xsl:template match="section">
  <fo:block break-before="page">
    <xsl:apply-templates/>
  </fo:block>
</xsl:template>
```

Αυτό λέει στη μηχανή μορφοποίησης XSL-FO να δημιουργήσει μία νέα σελίδα όταν βρεθεί σε ενότητα. Όλα τα περιεχόμενα αυτής της ενότητας υφίστανται επεξεργασία εντός αυτού του τμήματος. Για να κάνουμε το στοιχείο κεφαλής να ξεχωρίζει, βελτιώνουμε την εμφάνισή του επιλέγοντας μεγαλύτερο, έντονο μέγεθος γραμματοσειράς και προσθέτοντας μικρό διάστημα μετά το περιεχόμενο.

```
<xsl:template match="head">
  <fo:block   font-size="14pt"
font-weight="bold"
              space-after="1cm"
              space-after.conditionality = 'retain'
  >
    <xsl:apply-templates/>
  </fo:block>
</xsl:template>
```

Αυτό ήταν! Ανακεφαλαιώνοντας: η επεξεργασία είναι διαδικασία δύο σταδίων με απλά λόγια. Δώστε το έγγραφο πηγής σας και την παραπάνω σελίδα μορφοποιήσεων XSLT σε έναν επεξεργαστή XSLT, και η έξοδος θα πρέπει να είναι ένα έγκυρο έγγραφο XSL-FO. Στη συνέχεια αυτό μπορεί να τροφοδοτηθεί σε μηχανή XSL-FO – RenderX ή Antenna House (εμπορικά προϊόντα, με δοκιμαστικές εκδόσεις) ή στην PassiveTeX ή FOP (μη εμπορικά δωρεάν προϊόντα).