

## ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΕΧΟΜΕΝΑ .....	6-1
ΠΙΝΑΚΕΣ.....	6-1
ΕΙΚΟΝΕΣ.....	6-1
ΚΕΦΑΛΑΙΟ 6 – Εισαγωγή στην XML.....	6-2
6.1 Τι είναι η XML;.....	6-2
6.2 Βασικές διαφορές μεταξύ της XML και της HTML.....	6-2
6.3 Πού χρησιμοποιείται η XML;.....	6-3
6.4 Κανόνες σύνταξης της XML.....	6-4
6.5 Τα στοιχεία και τα χαρακτηριστικά στην XML.....	6-7
6.5.1 Στοιχεία .....	6-7
6.5.2 Χαρακτηριστικά .....	6-8
6.5.3 Επιλογή Χρήσης Στοιχείων ή Χαρακτηριστικών.....	6-8
6.6 Αναφορές οντοτήτων (Entity References) .....	6-10
6.7 Εγκυρότητα εγγράφων XML.....	6-11
6.8 Προσδιορισμός Τύπου Εγγράφου (DTD).....	6-11
6.8.1 Αναφορά και κλήση του DTD .....	6-12
6.8.2 Προσδιορισμός στοιχείων .....	6-13
6.8.3 Προσδιορισμός χαρακτηριστικών στο DTD .....	6-15
6.8.4 Ορισμός οντοτήτων στο DTD.....	6-17
6.8.5 Παραπομπή σε εξωτερικές οντότητες.....	6-18
6.9 Παρουσίαση εγγράφων XML.....	6-18
6.10 XSL.....	6-19
6.10.2 Τροποποίηση αρχείων με XSLT .....	6-20
6.10.3 Στοιχεία της XML .....	6-23
6.11 Ερωτήσεις – Ασκήσεις – Θέματα για ανάπτυξη .....	6-26

## ΠΙΝΑΚΕΣ

Πίνακας 6.6.1 – Ειδικοί χαρακτήρες και οι αντίστοιχες οντότητες .....	6-10
Πίνακας 6.8.1 - Σύμβολα που προστίθενται στον ορισμό ενός στοιχείου .....	6-14
Πίνακας 6.8.2 – Προκαθορισμένες τιμές και λέξεις κλειδιά για τα χαρακτηριστικά.....	6-15
Πίνακας 6.8.3 - Τύποι χαρακτηριστικών .....	6-16

## ΕΙΚΟΝΕΣ

Εικόνα 6.10.1 –Παρουσίαση του εγγράφου XML σε ένα φυλλομετρητή Internet Explorer 6. ....	6-21
Εικόνα 6.10.2 – Εμφάνιση του τροποποιημένου αρχείου XML ως ιστοσελίδα. ....	6-23

## ΚΕΦΑΛΑΙΟ 6 – Εισαγωγή στην XML

### 6.1 Τι είναι η XML;

Η XML (EXtensible Markup Language) είναι ένα πρότυπο επεξεργασίας κειμένων που αναπτύχθηκε από το World Wide Web Consortium (W3C). Η XML αποτελεί μία απλοποιημένη μορφή της SGML (Standard Generalized Markup Language) και σχεδιάστηκε για να ικανοποιήσει την ανάγκη της περιγραφής και της δόμησης των δεδομένων ενός εγγράφου με απλό και ευέλικτο τρόπο.

Η XML είναι μια γλώσσα σήμανσης και μορφοποίησης (markup language), όπως είναι και η HTML. Εστιάζει, όμως, στην περιγραφή των δεδομένων και όχι τόσο στον τρόπο με τον οποίο αυτά θα εμφανίζονται. Οι ετικέτες της XML δεν είναι προκαθορισμένες, όπως στην HTML. Η XML είναι επεκτάσιμη και ευέλικτη, καθώς επιτρέπει την προσθήκη νέων ετικετών και στοιχείων για την περιγραφή των δεδομένων. Η περιγραφή και ο ορισμός των στοιχείων που χρησιμοποιούνται γίνεται με τη χρήση ενός ορισμού τύπου εγγράφου (**DTD: Document Type Definition**), ενώ ο καθορισμός της μορφοποίησης και του τρόπου εμφάνισής τους γίνεται μέσω Cascading Style Sheets ή μέσω της γλώσσας XSL (Extensible Stylesheet Language). Η γλώσσα XSL αποτελεί μία επεκτάσιμη προδιαγραφή μορφοποίησης εγγράφων, που διασφαλίζει ότι τα έγγραφα παρουσιάζονται με τον ίδιο τρόπο ανεξάρτητα από την εφαρμογή ή την πλατφόρμα στην οποία εμφανίζονται.

Η XML είναι κάτι περισσότερο από μία γλώσσα σήμανσης και μορφοποίησης είναι μία μεταγλώσσα μέσω της οποίας ο καθένας μπορεί να ορίσει τις δικές του ετικέτες και τα δικά της στοιχεία μορφοποίησης εγγράφων. Είναι, επίσης, μια γλώσσα που μπορεί να χρησιμοποιηθεί για να καθορίσει νέες γλώσσες σήμανσης.

Η γλώσσα WML (Wireless Markup Language), που χρησιμοποιείται για τη μορφοποίηση και σήμανση εφαρμογών Διαδικτύου για συσκευές χεριού, όπως τα κινητά τηλέφωνα, είναι γραμμένη σε XML. Έτσι, η XML μπορεί να θεωρηθεί η μητέρα των WAP και WML.

### 6.2 Βασικές διαφορές μεταξύ της XML και της HTML

Η XML δεν σχεδιάστηκε για να αντικαταστήσει την HTML, αλλά για να τη συμπληρώσει. Η HTML εστιάζει στον τρόπο με τον οποίο θα εμφανίζεται το περιεχόμενο ενός εγγράφου και όχι τόσο στην περιγραφή του περιεχομένου του. Αντίθετα η XML εστιάζει στη δόμηση και την περιγραφή των δεδομένων του εγγράφου. Για ανάπτυξη εφαρμογών στο Διαδίκτυο είναι πιο σωστό να χρησιμοποιείται η XML για την περιγραφή δεδομένων, ενώ η HTML για τη μορφοποίηση και την εμφάνιση των δεδομένων αυτών.

Η XML μπορεί να διαχωρίσει τα δεδομένα από την HTML και να τα αποθηκεύσει σε ξεχωριστά αρχεία XML. Με αυτόν τον τρόπο η χρήση της HTML επικεντρώνεται στη διάταξη και την εμφάνιση δεδομένων, με αποτέλεσμα αλλαγές στα δεδομένα να μην προκαλούν αλλαγές στον κώδικα της σελίδας HTML. Τα δεδομένα XML μπορούν, επίσης, να αποθηκευτούν εντός των σελίδων HTML.

Στην HTML οι ετικέτες και η δομή ενός εγγράφου HTML, όπως είδαμε στο Κεφάλαιο 3, είναι προκαθορισμένα. Για το λόγο αυτό οι προγραμματιστές εφαρμογών σε HTML είναι υποχρεωμένοι να χρησιμοποιούν τις ετικέτες που έχουν οριστεί από το πρότυπο της HTML. Οι

ετικέτες προσδιορίζουν κυρίως τον τρόπο εμφάνισης των δεδομένων. Για παράδειγμα οι <B> και <I> δηλώνουν εάν το κείμενο θα εμφανιστεί με έντονα ή πλάγια γράμματα αντίστοιχα. Κάποιες από τις ετικέτες προσδιορίζουν, επίσης, τον τύπο των δεδομένων (π.χ. <IMG>).

Αντίθετα στην XML οι ετικέτες δεν έχουν προσδιοριστεί εκ των προτέρων και ορίζονται από τους προγραμματιστές που αναπτύσσουν εφαρμογές σε XML. Συμπεριφέρονται σαν ονόματα πεδίων και χρησιμοποιούνται κυρίως για να περιγράψουν το είδος των δεδομένων. Για παράδειγμα μία ετικέτα <message> ... </message> μπορεί να χρησιμοποιηθεί για να περιγράψει το περιεχόμενο ενός μηνύματος, ενώ μία ετικέτα <from>... </from> μπορεί να χρησιμοποιηθεί για να περιγράψει τον αποστολέα του μηνύματος.

Όπως αναφέρθηκε στην προηγούμενη ενότητα η XML έχει σχεδιαστεί για τη δόμηση και περιγραφή της πληροφορίας. Το παρακάτω παράδειγμα είναι ένα μήνυμα για τον John από την Mary, σε XML:

```
<message>
  <to>John</to>
  <from>Mary</from>
  <date>15-12-03</date>
  <subject>Meeting</subject>
  <content>Don't forget our meeting tomorrow at 11 am!</content>
</message>
```

Το περιεχόμενο του μηνύματος έχει συμπεριληφθεί σε μία ετικέτα <message>...</message>, ενώ η πληροφορία που περιλαμβάνει το μήνυμα έχει περαιτέρω δομηθεί με τη χρήση επιπλέον ετικετών. Το σημείωμα έχει ημερομηνία (date), έχει ένα θέμα (subject) και περιεχόμενο (content). Επίσης, έχει πληροφορία για τον αποστολέα (from) και τον παραλήπτη (to). Στο παράδειγμα αυτό, οι ετικέτες που χρησιμοποιήθηκαν π.χ. οι <to> και <from>, δεν είναι καθορισμένες σε κάποιο πρότυπο XML, αλλά "επινοήθηκαν" από το συγγραφέα του εγγράφου XML.

Αυτό το έγγραφο XML δεν περιέχει κώδικα, που μπορεί να εκτελεστεί, αλλά απλώς παρουσιάζει πληροφορία με ένα δομημένο τρόπο. Για περισσότερη επεξεργασία της πληροφορίας αυτής, όπως εμφάνιση των δεδομένων σε μία σελίδα HTML καθώς και αποστολή, λήψη, επεξεργασία και αποθήκευσή τους σε μία βάση δεδομένων, θα πρέπει να γραφτεί κατάλληλο πρόγραμμα.

### 6.3 Πού χρησιμοποιείται η XML;

Το βασικό πλεονέκτημα της XML είναι ότι μπορεί να χρησιμοποιηθεί σε κάθε πλατφόρμα, επειδή είναι ένα εργαλείο για τη μετάδοση δεδομένων ανεξάρτητο από το εγκατεστημένο λογισμικό και το υλικό. Ένα άλλο πλεονέκτημα της XML είναι ότι είναι επεκτάσιμη και προσφέρεται δωρεάν. Οι προγραμματιστές μπορούν να ορίσουν ό,τι ετικέτα θέλουν, για την περιγραφή και τη δόμηση των δεδομένων που επιθυμούν. Λόγω των προαναφερθέντων πλεονεκτημάτων το πρότυπο της XML αναπτύχθηκε πάρα πολύ γρήγορα και χρησιμοποιήθηκε από πολλές εταιρείες ανάπτυξης λογισμικού.

Με την XML μπορεί να επιτευχθεί η ανταλλαγή δεδομένων μεταξύ διαφορετικών εφαρμογών ή / και συστημάτων. Τα υπολογιστικά συστήματα και οι βάσεις δεδομένων περιέχουν δεδομένα σε ασύμβατες μορφές. Μία από τις πιο χρονοβόρες προκλήσεις για τους σχεδιαστές είναι η ανταλλαγή δεδομένων μεταξύ τέτοιων συστημάτων μέσω του Διαδικτύου. Μετατρέποντας τα δεδομένα σε XML, μειώνεται σημαντικά η πολυπλοκότητα και δημιουργούνται δεδομένα που μπορούν να διαβαστούν από πολλές διαφορετικές εφαρμογές.

Τα δεδομένα στην XML αποθηκεύονται σε μορφή απλού κειμένου, με αποτέλεσμα να μπορούν να επεξεργαστούν από πολλών ειδών πλατφόρμες. Το γεγονός ότι η XML παρέχει έναν τρόπο παρουσίασης και δόμησης δεδομένων ανεξάρτητο από λογισμικό και υλικό, διευκολύνει τη διανομή δεδομένων και τη χρήση και επεξεργασία τους από διαφορετικές εφαρμογές.

Χρησιμοποιώντας την XML τα δεδομένα μας είναι διαθέσιμα σε περισσότερους χρήστες, εφόσον πολλές εφαρμογές πελάτη μπορούν να έχουν πρόσβαση στα αρχεία XML σαν πηγές δεδομένων (σαν βάσεις δεδομένων).

Η XML έχει, επίσης, δώσει λύσεις σε εφαρμογές ηλεκτρονικού εμπορίου. Στο ηλεκτρονικό εμπόριο τύπου "πελάτη προς επιχείρηση", η XML παρέχει έναν ευέλικτο τρόπο για τη δόμηση των πληροφοριών που παρουσιάζονται στον πελάτη, όπως οι πληροφορίες που αφορούν ένα προϊόν, ή οι πληροφορίες που αφορούν το καλάθι αγορών του καταναλωτή. Τα δεδομένα μπορούν να διαχωριστούν από την σελίδα HTML και με τη χρήση CSS και XSL να εμφανιστούν σε ένα φυλλομετρητή.

Η XML χρησιμοποιείται, επίσης, στο ηλεκτρονικό εμπόριο τύπου "επιχείρηση προς επιχείρηση", όπου εξελίσσεται ως βασική γλώσσα για την ανταλλαγή οικονομικής πληροφορίας μέσω του Διαδικτύου. Η έρευνα για την ανάπτυξη προτύπων XML ικανών να υποστηρίξουν ανταλλαγή πληροφορίας, όπως ηλεκτρονικές παραγγελίες, ηλεκτρονικά τιμολόγια κτλ., βρίσκεται σε εξέλιξη, ενώ έχουν ήδη αναπτυχθεί ενδιαφέρουσες εφαρμογές B2B σε XML.

## 6.4 Κανόνες σύνταξης της XML

Οι κανόνες σύνταξης της XML είναι πολύ απλοί, αλλά πρέπει να ακολουθούνται με ακρίβεια. Ένα πλήρες έγγραφο XML έχει την ακόλουθη μορφή:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<message>
<to>John</to>
<from>Mary</from>
<date>15-12-03</date>
<subject>Meeting</subject>
<content>Don't forget our meeting tomorrow at 11 am!</content>
</message>
```

Η πρώτη γραμμή του εγγράφου – η δήλωση XML – ορίζει την έκδοση της XML και την κωδικοποίηση χαρακτήρων που χρησιμοποιείται στο έγγραφο. Γενικά, η πρώτη γραμμή ενός εγγράφου XML έχει την ακόλουθη σύνταξη:

```
<?xml version= "number" [encoding= "encoding"] [standalone= "yes|no"]?>
```

Παρόλο που δεν είναι υποχρεωτικό, τα έγγραφα XML συνήθως ξεκινούν με μία δήλωση XML της παραπάνω μορφής. Το πρώτο χαρακτηριστικό δείχνει την έκδοση της XML, που χρησιμοποιείται. Το δεύτερο και τρίτο χαρακτηριστικό δεν είναι υποχρεωτικά και γι' αυτό έχουν τοποθετηθεί μέσα σε αγκύλες. Το χαρακτηριστικό "encoding" δηλώνει την κωδικοποίηση που έχει χρησιμοποιηθεί στο έγγραφο (π.χ. "US-ASCII"), ενώ το χαρακτηριστικό "standalone" παίρνει τιμές "yes" και "no" και προσδιορίζει εάν για την ανάλυση του κώδικα του εγγράφου απαιτείται κάποιο DTD. Πληροφορίες και επεξηγήσεις για τα DTD θα δοθούν σε επόμενη ενότητα.

Στο παράδειγμα αυτό, το έγγραφο διαμορφώνεται σύμφωνα με την προδιαγραφή 1.0 της XML και χρησιμοποιεί το σύνολο χαρακτήρων ISO-8859-1 (Latin-1/West European).

Οι επόμενες γραμμές περιγράφουν τα δεδομένα του εγγράφου. Στο συγκεκριμένο παράδειγμα όλα τα δεδομένα έχουν τοποθετηθεί μέσα σε μία ετικέτα <message>, που αποτελεί τη ρίζα του εγγράφου και δηλώνει πως πρόκειται για τα περιεχόμενα ενός μηνύματος. Οι πληροφορίες του μηνύματος έχουν περαιτέρω δομηθεί με τη βοήθεια επιπλέον ετικετών όπως οι <to>, <from>, <date>, <subject> και <content>. Οι ετικέτες αυτές περιλαμβάνονται μέσα στη ρίζα <message> και αποτελούν τα παιδιά της.

```
<to>John</to>
<from>Mary</from>
<date>15-12-03</date>
<subject>Meeting</subject>
<content>Don't forget our meeting tomorrow at 11 am!</content>
```

Στις παραγράφους που ακολουθούν περιγράφονται οι βασικοί κανόνες σύνταξης ενός εγγράφου XML.

- **Όλα τα στοιχεία της XML πρέπει να έχουν ετικέτες τέλους**

Στην XML, δεν επιτρέπεται η παράλειψη της ετικέτας τέλους. Αντίθετα στην HTML κάποια στοιχεία μπορούν να μην έχουν ετικέτα τέλους. Για παράδειγμα, ενώ στην HTML θα μπορούσε να γραφτεί:

```
<p>This is a paragraph
<p>This is another paragraph
```

Στην XML το ίδιο θα πρέπει να γραφτεί ως εξής:

```
<p>This is a paragraph</p>
<p>This is another paragraph</p>
```

Όπως φαίνεται στο προηγούμενο παράδειγμα, η μόνη ετικέτα που δεν έχει ετικέτα τέλους είναι η δήλωση της XML: <?xml ...>. Η δήλωση δε θεωρείται μέρος του εγγράφου XML, δεν αποτελεί στοιχείο της XML και επομένως δεν πρέπει να έχει ετικέτα τέλους.

- **Όταν μία ετικέτα δεν έχει περιεχόμενο (κενή ετικέτα), περιέχει μία πλάγια γραμμή τέλους πριν το κλείσιμο της ετικέτας**

Η XML επιτρέπει, επίσης, την προσθήκη μιας ετικέτας χωρίς ετικέτα τέλους. Δηλαδή μιας ετικέτας που τελειώνει σε /> αντί σε >. Μια τέτοια ετικέτα δεν έχει κάποιο περιεχόμενο και γι' αυτό ονομάζεται “κενή”. Μία κενή ετικέτα μπορεί για παράδειγμα, να χρησιμοποιηθεί σε ένα μήνυμα για να σηματοδοτήσει την σπουδαιότητά του:

```
<message to="you@yourAddress.com" from="me@myAddress.com"
subject="XML" >
<flag/>
<text>
XML is great!
</text>
</message>
```

Η κενή ετικέτα <flag/> αντικαθιστά τη <flag></flag> και μας απαλλάσσει από την άσκοπη προσθήκη μιας ετικέτας τέλους.

- **Οι ετικέτες XML διαφέρουν αν γραφτούν με πεζά ή κεφαλαία γράμματα**

Για παράδειγμα, η ετικέτα <Lesson> είναι διαφορετική από την <lesson>. Γι' αυτό το λόγο η ετικέτα αρχής θα πρέπει να είναι ίδια με την ετικέτα τέλους:

```
<Lesson>This syntax is incorrect </lesson>
```

Στην HTML δεν ισχύει κάτι τέτοιο καθώς η ετικέτα <Body> είναι για παράδειγμα ίδια με τη <BODY> ή τη <body>.

- **Όλα τα στοιχεία της XML πρέπει να είναι τοποθετημένα σωστά το ένα μέσα στο άλλο**

Στην HTML δεν παρουσιάζεται πρόβλημα εάν γραφεί:

```
<b><i>This text is bold and italic</b></i>
```

Στην XML όμως, η σύνταξη είναι πιο αυστηρή και τα στοιχεία πρέπει να τοποθετούνται με τη σωστή σειρά το ένα μέσα στο άλλο:

```
<b><i>This text is bold and italic</i></b>
```

- **Όλα τα έγγραφα XML πρέπει να έχουν ένα αρχικό στοιχείο/ρίζα**

Όλα τα έγγραφα XML πρέπει να περιέχουν ένα ζεύγος ετικετών που καθορίζουν τη ρίζα του και όλα τα υπόλοιπα στοιχεία πρέπει να βρίσκονται εντός αυτών. Στο παράδειγμα που παρουσιάστηκε το στοιχείο <message> αποτέλεσε τη ρίζα του εγγράφου, ενώ όλα τα στοιχεία <to>, <from> κτλ., τα παιδιά του.

- **Όλα τα στοιχεία μπορούν να έχουν υπο-στοιχεία (παιδιά)**

Όπως φαίνεται στο ακόλουθο δείγμα κώδικα, όλα τα στοιχεία που βρίσκονται σε ένα έγγραφο XML μπορούν να περιλαμβάνουν παιδιά:

```
<root>
  <child1>...
    <subchild1>.....</subchild1>
  </child1>
  <child2>...
    <subchild2>.....</subchild2>
    <subchild2>.....</subchild2>
  </child2>
  <child3>...</child3>
</root>
```

- **Στα στοιχεία μπορούν να συμπεριληφθούν χαρακτηριστικά. Οι τιμές των χαρακτηριστικών βρίσκονται πάντα μέσα σε εισαγωγικά**

Όπως θα περιγραφεί, στις ενότητες που ακολουθούν τα στοιχεία μπορούν να περιλαμβάνουν χαρακτηριστικά που τους δίνουν πρόσθετη πληροφορία. Τα χαρακτηριστικά των στοιχείων στην XML παρουσιάζονται σε ζεύγη όνομα/τιμή, όπως και στην HTML. Οι τιμές των χαρακτηριστικών τοποθετούνται πάντα μέσα σε εισαγωγικά

Η παράλειψη των εισαγωγικών στις τιμές των χαρακτηριστικών δεν επιτρέπεται. Έτσι, το πρώτο από τα δύο παραδείγματα που ακολουθούν είναι λάθος και το δεύτερο σωστό:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<price currency=Euro>
  <value>25</value>
</price>
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<price currency ="Euro">
  <value>25</value>
</price>
```

## ▪ Σχόλια στην XML

Όπως φαίνεται στο δείγμα κώδικα που ακολουθεί, η σύνταξη για τη γραφή σχολίων είναι παρόμοια με αυτή της HTML:

```
<!-- This is a comment -->
```

## 6.5 Τα στοιχεία και τα χαρακτηριστικά στην XML

### 6.5.1 Στοιχεία

Τα στοιχεία XML χρησιμοποιούνται για να περιγράψουν πληροφορία και μπορούν να περιλαμβάνουν πληροφορία, άλλα στοιχεία ή συνδυασμό και των δύο.

```
<paragraph> This is a paragraph that contains <emphasis> emphasized words
</emphasis> </paragraph>
```

Τα στοιχεία μπορούν να περιέχουν υπο-στοιχεία και μπορούν να επεκταθούν ώστε να συμπεριλάβουν περισσότερη πληροφορία. Για παράδειγμα ένα στοιχείο που περιλαμβάνει πληροφορία για ένα βιβλίο (όπως τον τίτλο και τον συγγραφέα) μπορεί να επεκταθεί ώστε να συμπεριλάβει και τους τίτλους των κεφαλαίων του βιβλίου:

```
<book>
<title>Using HTML, XML and Java</title>
<author>Eric Ladd</author>
</book>
```

Η πληροφορία αυτή μπορεί να επεκταθεί ως εξής:

```
<book>
<title>Using HTML, XML and Java</title>
<author>Eric Ladd</author>
<chapter>HTML and Graphics</chapter>
<chapter>XML</chapter>
```

```
</book>
```

Τα στοιχεία συμβολίζονται με ετικέτες των οποίων τα ονόματα δίνονται με βάση τους παρακάτω κανόνες:

- Τα ονόματα μπορούν να περιλαμβάνουν όχι μόνο γράμματα και αριθμούς αλλά και άλλους χαρακτήρες όπως η κάτω παύλα. Παραδείγματα έγκυρων ονομάτων είναι τα `<_Budget>`, `<Italic>`, `<Chapter1>`.
- Τα ονόματα δε θα πρέπει να ξεκινούν με κάποιο αριθμό, με σημεία στίξης ή με τα γράμματα xml (xml ή XML). Όνομα του τύπου `<25target>` είναι λάθος.
- Επίσης, τα ονόματα των ετικετών δε θα πρέπει να περιλαμβάνουν κενά, όπως `<book title>`.
- Τέλος στο εσωτερικό των ονομάτων θα πρέπει να αποφεύγονται οι τελείες και τα μείον.

### 6.5.2 Χαρακτηριστικά

Όπως είδαμε, οι ετικέτες στην XML μπορούν να περιέχουν χαρακτηριστικά, πρόσθετη δηλαδή πληροφορία που τοποθετείται εντός της ετικέτας σαν μέρος της ετικέτας. Το παρακάτω παράδειγμα δείχνει τη δομή ενός ηλεκτρονικού μηνύματος, που χρησιμοποιεί χαρακτηριστικά για τα πεδία "to", "from", και "subject":

```
<message to="you@yourAddress.com" from="me@myAddress.com" subject="XML">
  <text>
    XML is great!
  </text>
</message>
```

Όπως και στην HTML, το όνομα του χαρακτηριστικού ακολουθείται από το σύμβολο "=" και την τιμή του χαρακτηριστικού. Τα χαρακτηριστικά χωρίζονται μεταξύ τους με κενά. Αντίθετα με την HTML, στην XML το κόμμα μεταξύ των χαρακτηριστικών δεν αγνοείται. Αν υπάρχει κόμμα, θα προκύψει λάθος.

Οι τιμές των χαρακτηριστικών μπορούν να τοποθετηθούν είτε ανάμεσα σε διπλά εισαγωγικά είτε ανάμεσα σε μονά εισαγωγικά, όπως φαίνεται στα παραδείγματα:

```
<meeting day= "Monday"> ... </meeting>
```

ή

```
<meeting day= 'Monday'> ... </meeting>
```

### 6.5.3 Επιλογή Χρήσης Στοιχείων ή Χαρακτηριστικών

Κάτι που συνήθως πρέπει να επιλυθεί αρχικά κατά τη δημιουργία μιας δομής δεδομένων σε XML είναι εάν κάποια πληροφορία θα παρουσιαστεί σαν υπο-στοιχείο ή σαν χαρακτηριστικό ενός υπάρχοντος στοιχείου. Για παράδειγμα, θα μπορούσαμε να ορίσουμε τον τίτλο μιας εικόνας ως:

```
<slide>
  <title>This is the title</title>
</slide>
```

ή ως:



Η επιλογή για το τι από τα δύο θα χρησιμοποιηθεί βασίζεται πολλές φορές στον τύπο και τα χαρακτηριστικά των δεδομένων που πρόκειται να απεικονιστούν.

- **Τα δεδομένα περιέχουν υποδομές**

Όταν τα δεδομένα περιέχουν υποδομές είναι προτιμότερο να παρουσιαστούν με στοιχεία παρά με χαρακτηριστικά. Τα στοιχεία μπορούν να παρουσιάσουν υποδομές ενώ τα χαρακτηριστικά όχι. Για παράδειγμα, ένας τίτλος που περιλαμβάνει έντονο κείμενο όπως: The **Best** Choice, είναι προτιμότερο να περιγραφεί από κάποιο στοιχείο. Τα χαρακτηριστικά μπορούν να πάρουν μόνο μία τιμή αλφαριθμητικού τύπου .

- **Τα δεδομένα περιέχουν πολλαπλές γραμμές**

Και σε αυτήν την περίπτωση καλό είναι να χρησιμοποιούνται στοιχεία. Τα χαρακτηριστικά βρίσκονται μέσα στις ετικέτες των στοιχείων και πρέπει να έχουν περιορισμένου μήκους τιμές για να είναι εύκολα αναγνώσιμα.

- **Τα δεδομένα αλλάζουν συχνά**

Όταν τα δεδομένα τροποποιούνται συχνά, ειδικά από τον τελικό χρήστη, είναι προτιμότερο να παρουσιάζονται ως στοιχεία διότι η αναζήτηση και η τροποποίηση των χαρακτηριστικών είναι πολύ πιο δύσκολη.

- **Τα δεδομένα είναι μικρά, απλό κείμενο που δεν τροποποιείται σχεδόν ποτέ**

Στην περίπτωση αυτή τα δεδομένα μπορούν να απεικονιστούν και ως χαρακτηριστικά.

- **Οι τιμές των δεδομένων είναι συγκεκριμένες και περιορίζονται σε ένα μικρό αριθμό επιλογών**

Σε αυτή την περίπτωση είναι προτιμότερη η χρήση ενός χαρακτηριστικού καθώς μέσω του DTD, μπορεί να προσδιοριστεί λίστα με τις δυνατές τιμές που μπορεί να πάρει το χαρακτηριστικό. Αυτό όμως συνεπάγεται ότι στο έγγραφο XML δεν μπορούν να χρησιμοποιηθούν τιμές που δεν αναφέρονται στο DTD. Αν χρειαστεί να προστεθεί μια επιπλέον τιμή θα πρέπει να αλλάξει το DTD.

Εκτός από τους προαναφερθέντες παράγοντες που επηρεάζουν σημαντικά την επιλογή απεικόνισης δεδομένων με χρήση είτε στοιχείων είτε χαρακτηριστικών, υπάρχουν και λιγότερο προφανείς παράγοντες, όπως αυτοί που αναλύονται στις παρακάτω παραγράφους:

- **Εμφάνιση δεδομένων στον τελικό χρήστη**

Αν τα δεδομένα πρόκειται να παρουσιαστούν στον τελικό χρήστη, είναι προτιμότερο να απεικονίζονται ως στοιχεία. Ενώ αν η πληροφορία συμβάλει στην επεξεργασία του εγγράφου XML αλλά δεν πρόκειται να εμφανιστεί πουθενά, είναι ίσως καλύτερα να παρουσιάζεται ως χαρακτηριστικό. Για παράδειγμα, εάν το έγγραφο XML αφορά δεδομένα για παραγγελία ενδυμάτων, το μέγεθος θα μπορούσε να απεικονιστεί με ένα στοιχείο, ενώ ο κωδικός του κατασκευαστή με ένα χαρακτηριστικό.

- **Εισαγωγή πληροφορίας**

Ένας άλλος παράγοντας που επηρεάζει τον τρόπο απεικόνισης της πληροφορίας είναι ο τρόπος εισαγωγής της. Εάν για παράδειγμα, το μέγεθος ενός ενδύματος εισαχθεί από κάποιον

υπάλληλο, είναι προτιμότερο να παρουσιαστεί ως στοιχείο. Ενώ εάν δοθεί αυτόματα μέσω κάποιας εφαρμογής μπορεί να παρουσιαστεί και ως χαρακτηριστικό.

#### ▪ Είδος πληροφορίας

Ο τρόπος παρουσίασης της πληροφορίας επηρεάζεται, επίσης από το ίδιο το είδος της πληροφορίας. Αν για παράδειγμα θεωρήσουμε ένα στοιχείο ως δοχείο, τότε μπορούμε να πούμε ότι τα περιεχόμενα του δοχείου (νερό ή γάλα) ανταποκρίνονται σε δεδομένα XML, σχεδιασμένα ως στοιχεία, ενώ, τα χαρακτηριστικά του δοχείου (άσπρο, μαύρο) αντιστοιχούν σε δεδομένα XML που θα αντιμετωπιστούν σαν χαρακτηριστικά.

## 6.6 Αναφορές οντοτήτων (Entity References)

Στην XML κάποιοι χαρακτήρες είναι δεσμευμένοι και δεν μπορούν να τοποθετηθούν στο περιεχόμενο ενός εγγράφου XML. Τέτοιου είδους χαρακτήρες μπορούν να εισαχθούν στα δεδομένα ενός εγγράφου XML με τη χρήση αναφορών στις οντότητες αυτές. Οι αναφορές σε οντότητες (entity references) χρησιμοποιούνται για να υποκαταστήσουν οντότητες που εάν είχαν τοποθετηθεί στο έγγραφο της XML, θα είχαν μεταφραστεί ως χαρακτήρες σήμανσης και μορφοποίησης. Η εισαγωγή μίας αναφοράς σε οντότητα προκαλεί την εισαγωγή της ίδιας της οντότητας, δηλαδή του ειδικού χαρακτήρα, στο έγγραφο XML.

Στην XML υπάρχουν πέντε προκαθορισμένες αναφορές οντοτήτων που αφορούν βασικούς χαρακτήρες μορφοποίησης. Υποστηρίζεται, όμως, και η δυνατότητα προσδιορισμού νέων αναφορών σε οντότητες, μέσω του DTD, που συνοδεύει ένα έγγραφο XML.

Οι αναφορές σε οντότητες ξεκινούν πάντα με το σύμβολο & και έχουν πάντα στο τέλος το σύμβολο του ερωτηματικού, π.χ. &entityName;

Οι προκαθορισμένες οντότητες που υποστηρίζει η XML φαίνεται στον πίνακα που ακολουθεί:

**Πίνακας 6.6.1 – Ειδικοί χαρακτήρες και οι αντίστοιχες οντότητες**

Χαρακτήρας	Αναφορά
&	&amp;
<	&lt;
>	&gt;
"	&quot;
'	&apos;

Σύμφωνα με τα προαναφερόμενα αν σε ένα έγγραφο XML θέλουμε να προσθέσουμε την ακόλουθη γραμμή:

```
Market Size < predicted
```

Θα γράψουμε :

```
Market Size &lt; predicted
```

Το "<", δηλαδή, που είναι ένας χαρακτήρας σήμανσης, θα αντικατασταθεί με την αντίστοιχη αναφορά.

Στην XML υποστηρίζεται, επίσης, αναφορά σε χαρακτήρες τύπου Unicode η οποία γίνεται με αναφορά στον δεκαεξαδικό αριθμό που αντιστοιχεί στο χαρακτήρα. Η αναφορά αυτή αποτελείται από τη συμβολοσειρά &#x που ακολουθείται από τον δεκαεξαδικό αριθμό που αντιστοιχεί στο χαρακτήρα. Για παράδειγμα το &#xA9, αποτελεί μία αναφορά στο χαρακτήρα © που συμβολίζει τα πνευματικά δικαιώματα.

## 6.7 Εγκυρότητα εγγράφων XML

Ένα έγγραφο XML, που είναι συντακτικά σωστό ονομάζεται *καλά σχηματισμένο* ("well-formed"). Ένα συντακτικά ορθό έγγραφο δεν περιλαμβάνει αγκύλες, που δεν είναι μέρος των ετικετών. Στην περίπτωση που πρέπει να τοποθετηθούν αγκύλες στο περιεχόμενο του εγγράφου χρησιμοποιούνται οι αναφορές &lt; και &gt;. Όλες οι ετικέτες έχουν μια τελική ετικέτα ή είναι κενές ετικέτες που κλείνουν μόνες τους (<slide>..</slide> ή <slide/>). Σε ένα καλά σχηματισμένο έγγραφο, οι ετικέτες είναι όλες σωστά εμφωλιασμένες. Έτσι, δεν υπάρχουν συνδυασμοί ετικετών της μορφής <slide><image>..</slide></image>, αλλά <slide><image>..</image></slide>.

Ένα καλά σχηματισμένο έγγραφο δεν είναι όμως απαραίτητα και *έγκυρο*. Για να είναι ένα έγγραφο έγκυρο θα πρέπει εκτός από συντακτικά σωστό, να βασίζεται και στους κανόνες ενός DTD.

## 6.8 Προσδιορισμός Τύπου Εγγράφου (DTD)

Ένα DTD προσδιορίζει τη δομή ενός εγγράφου XML καθώς και τη λίστα με τα "νόμιμα" στοιχεία που το αποτελούν. Παρέχει τους γραμματικούς κανόνες του εγγράφου και των στοιχείων του. Προσδιορίζει τα χαρακτηριστικά των στοιχείων και καθορίζει τις ετικέτες που θα συμπεριληφθούν σε ένα έγγραφο XML.

Εάν ένα έγγραφο XML είναι καλά σχηματισμένο δεν είναι απαραίτητο να συνοδεύεται από ένα DTD. Άλλωστε σε πολύ καλά δομημένα έγγραφα XML το DTD μπορεί να προκύψει αυτόματα, κοιτάζοντας απλά τη δομή και το περιεχόμενο του εγγράφου XML. Η έλλειψη, όμως ενός DTD οδηγεί σε περιορισμούς στις δυνατότητες του εγγράφου XML.

Σε περίπτωση που το έγγραφο XML δεν συνοδεύεται από κάποιο DTD, για να είναι έγκυρο θα πρέπει να ισχύουν τα ακόλουθα:

- Θα πρέπει να έχουν προσδιοριστεί τιμές για όλα τα χαρακτηριστικά, καθώς δεν είναι δυνατή η ύπαρξη προκαθορισμένων τιμών για τα χαρακτηριστικά.
- Δεν μπορεί να συμπεριληφθούν αναφορές σε οντότητες, εκτός βέβαια από τις πέντε προκαθορισμένες αναφορές (&amp; &lt; &gt; &apos; και &quot;) που υποστηρίζει η XML.
- Δε μπορεί να υπάρχουν χαρακτηριστικά των οποίων οι τιμές χρειάζονται κανονικοποίηση.
- Δε μπορεί να υπάρχουν κενά ανάμεσα στην ετικέτα αρχής ενός στοιχείου και στην ετικέτα αρχής υποστοιχείου της. Για παράδειγμα η δομή <chapter> <title> ... </title></chapter> δεν είναι σωστή και μπερδεύει τον επεξεργαστή XML. Χωρίς ένα κατάλληλο DTD, το οποίο προσδιορίζει εάν το κενό είναι μέρος της τιμής του στοιχείου ή εάν θα πρέπει να αγνοηθεί, ο επεξεργαστής της XML δε ξέρει εάν θα πρέπει να τα αγνοήσει ή όχι.

Οι προαναφερόμενοι περιορισμοί δεν υπάρχουν εάν το έγγραφο XML συνοδεύεται από ένα DTD. Το DTD περιγράφει ένα μοντέλο για τη δομή και το περιεχόμενο ενός εγγράφου XML. Περιγράφει τις σχέσεις ανάμεσα στα στοιχεία του εγγράφου και προσδιορίζει ποια στοιχεία είναι απαραίτητο να υπάρχουν και ποια μπορούν να παραληφθούν. Ενώ η HTML έχει

μόνο ένα DTD, η XML επιτρέπει τη δημιουργία ενός διαφορετικού DTD για κάθε εφαρμογή. Στο DTD, που προσδιορίζεται για κάθε εφαρμογή, βασίζεται ο έλεγχος της εγκυρότητας (validation), δηλαδή ο έλεγχος της δομής και του περιεχομένου ενός εγγράφου XML.

Ένα DTD έχει τη μορφή που φαίνεται στο παράδειγμα που ακολουθεί:

```
<!ELEMENT product (name, category?, color?, description?, product-number, availability?, price?)>
<!ELEMENT name (#PCDATA)*>
<!ELEMENT description (#PCDATA)*>
<!ELEMENT product-number (#PCDATA)*>
<!ELEMENT availability (#PCDATA)*>
<!ELEMENT price (#PCDATA)*>
<!ATTLIST product price valid-from #required>
```

Οι αρχικές ετικέτες του παραδείγματος αφορούν τον προσδιορισμό στοιχείων, ενώ η τελευταία τον προσδιορισμό των χαρακτηριστικών ενός στοιχείου. Οι ετικέτες ενός DTD ξεκινούν με θαυμαστικό <!, το οποίο ακολουθείται από μία δεσμευμένη λέξη (ELEMENT, ENTITY, ATTLIST κ.α.) που προσδιορίζει το είδος της οντότητας που θα οριστεί. Δηλώνεται δηλαδή εάν πρόκειται για ορισμό στοιχείου, αναφορά σε οντότητα ή χαρακτηριστικού. Περισσότερες επεξηγήσεις για το περιεχόμενο των ετικετών του παραδείγματος θα δοθούν στις παραγράφους που ακολουθούν.

### 6.8.1 Αναφορά και κλήση του DTD

Η διασύνδεση ενός εγγράφου XML με ένα DTD γίνεται με την προσθήκη μίας δήλωσης μετά τη δήλωση XML και πριν το αρχικό στοιχείο του αρχείου XML. Η δήλωση του DTD μπορεί να έχει την ακόλουθη μορφή:

```
<!DOCTYPE root-element SYSTEM "URI_of_DTD">
```

ή

```
<!DOCTYPE root-element PUBLIC "name" "URI_of_DTD">
```

Η λέξη DOCTYPE ορίζει τη δήλωση ενός DTD. Το root-element είναι το όνομα του αρχικού στοιχείου/ρίζας του εγγράφου XML.

Η δήλωση SYSTEM προσδιορίζει τη διεύθυνση URI του DTD και χρησιμοποιείται όταν το DTD έχει οριστεί μόνο για το συγκεκριμένο αρχείο. Αντίθετα η δήλωση PUBLIC χρησιμοποιείται όταν το DTD έχει δημοσιευτεί και μπορεί να χρησιμοποιηθεί από πολλά διαφορετικά έγγραφα XML. Στην περίπτωση αυτή το DTD έχει ένα όνομα με το οποίο το αναγνωρίζει ο επεξεργαστής XML, καθώς και τη διεύθυνση URI στην οποία βρίσκεται.

Στο παράδειγμα δήλωσης που ακολουθεί, το DOCTYPE δηλώνει τον προσδιορισμό ενός DTD για ένα έγγραφο XML με στοιχείο ρίζας το “example”.

```
<!DOCTYPE example SYSTEM "example.dtd">
```

Ο προσδιορισμός του στοιχείου ρίζας δηλώνει ότι το έγγραφο αποτελείται από ένα στοιχείο example το οποίο περιλαμβάνει οποιοδήποτε άλλο στοιχείο του εγγράφου.

```
<example>
...
</example>
```

Μετά τη δήλωση SYSTEM ορίζεται η τοποθεσία του αρχείου DTD. Στο συγκεκριμένο παράδειγμα υπάρχει μόνο το όνομα του αρχείου “example.dtd” χωρίς να έχει προσδιοριστεί κάποιο μονοπάτι που να αρχίζει με http:/ ή να προσδιορίζει ονόματα φακέλων. Αυτό σημαίνει ότι η τοποθεσία του DTD είναι ίδια με αυτή του εγγράφου XML.

Η δήλωση DOCTYPE, αντί να αναφέρεται σε κάποιο εξωτερικό αρχείο DTD, θα μπορούσε να περιέχει δηλώσεις DTD εντός του εγγράφου XML. Όπως δείχνει το παράδειγμα σύνταξης που ακολουθεί οι δηλώσεις αυτές περικλείονται σε αγκύλες:

```
<!DOCTYPE example SYSTEM "example.dtd" [
...here we put the local subset definitions...
]>
```

### 6.8.2 Προσδιορισμός στοιχείων

Ο προσδιορισμός στοιχείων σε ένα DTD γίνεται με τη χρήση της δήλωσης ELEMENT. Η σύνταξη της δήλωσης ενός στοιχείου είναι η ακόλουθη:

```
<!ELEMENT element_name rule>
```

Η δήλωση αυτή προσδιορίζει ένα στοιχείο με όνομα element\_name. Περιλαμβάνει επίσης κάποιο κανόνα που αφορά είτε την τιμή του στοιχείου είτε τις σχέσεις του με άλλα στοιχεία του εγγράφου XML.

Όταν η τιμή ενός στοιχείου είναι οποιοσδήποτε τύπος συνόλου χαρακτήρων μπορεί να οριστεί με τη χρήση της δήλωσης PCDATA, που συμβολίζει χαρακτήρες που έχουν αναλυθεί (parsed character data), δηλαδή αλφαριθμητικά δεδομένα. Ο προσδιορισμός ενός τέτοιου στοιχείου γίνεται με μία δήλωση της μορφής:

```
<!ELEMENT element_name (#PCDATA)>
```

Το σύμβολο "#" που προηγείται του PCDATA δηλώνει ότι η λέξη που ακολουθεί είναι μια ειδική λέξη και όχι κάποιο όνομα στοιχείου. Η δήλωση αυτή προσδιορίζει πως το στοιχείο με όνομα element\_name μπορεί να πάρει οποιαδήποτε τιμή αρκεί να μην πρόκειται για κάποιο άλλο στοιχείο. Με βάση τη δήλωση αυτή το στοιχείο “element\_name” μπορεί να παρουσιαστεί σε ένα έγγραφο XML με τη μορφή <element\_name></element\_name>ή <element\_name>a sentence </element\_name>. Δε μπορεί, όμως, να εμφανιστεί με τη μορφή <element\_name><another\_element> a value </another\_element></element\_name>.

Για να οριστεί ένα στοιχείο που περιλαμβάνει κάποιο υπο-στοιχείο, χρησιμοποιείται πάλι η δήλωση ELEMENT ακολουθούμενη από το όνομα του στοιχείου, μόνο που αντί για τον τύπο περιεχομένου #PCDATA, τοποθετείται το όνομα του υπο-στοιχείου μέσα σε παρένθεση. Για παράδειγμα, η δήλωση που ακολουθεί ορίζει ένα στοιχείο με όνομα product το οποίο περιλαμβάνει ένα υπο-στοιχείο με όνομα price. Στη δεύτερη δήλωση ορίζεται το στοιχείο price το οποίο περιέχει τιμή τύπου PCDATA.

```
<!ELEMENT product (price)>
<!ELEMENT price (#PCDATA)>
```

Το ίδιο στοιχείο μπορεί να περιλαμβάνει πολλά υπο-στοιχεία τα οποία, όπως φαίνεται στο παράδειγμα που ακολουθεί, ορίζονται στη δήλωση του στοιχείου χωρισμένα με ερωτηματικά.

```
<!ELEMENT product (name, price)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT price (#PCDATA)>
```

Στο DTD μπορεί να οριστεί και ένα στοιχείο που περιλαμβάνει ένα εκ των δύο υπο-στοιχείων που εμφανίζονται στον ορισμό του. Αυτό επιτυγχάνεται με τη χρήση της οριζόντιας γραμμής (|). Έτσι στο στοιχείο book που παρουσιάζεται παρακάτω, μπορεί να περιλαμβάνεται είτε το στοιχείο “title” είτε το στοιχείο “author”, αλλά όχι και τα δύο μαζί.

```
<!ELEMENT book (title | author)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
```

### **Ομαδοποίηση και επαναλήψεις στοιχείων**

Για την απόκτηση μεγαλύτερης ευελιξίας στον ορισμό των στοιχείων υπάρχει η δυνατότητα ομαδοποίησης των τιμών ή των υπο-στοιχείων τους. Η ομαδοποίηση επιτυγχάνεται με την προσθήκη παρενθέσεων που περικλείουν τα υπο-στοιχεία. Το παράδειγμα που ακολουθεί δηλώνει πως το στοιχείο product μπορεί να περιλαμβάνει είτε όνομα και κωδικό προϊόντος είτε την τιμή του προϊόντος.

```
<!ELEMENT product ((name, product_code) | price)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT product_code (#PCDATA)>
<!ELEMENT price (#PCDATA)>
```

Στη δήλωση των στοιχείων μπορούν, επίσης, να χρησιμοποιηθούν σύμβολα που επιτρέπουν τον προσδιορισμό προαιρετικών τιμών ή στοιχείων, στοιχείων που υπάρχουν τουλάχιστον μία φορά και στοιχείων που επαναλαμβάνονται. Τα σύμβολα αυτά παρουσιάζονται στον Πίνακα 6.8.1:

**Πίνακας 6.8.1 - Σύμβολα που προστίθενται στον ορισμό ενός στοιχείου**

Σύμβολο	Όνομα	Σημασία
?	Ερωτηματικό	Προαιρετικό (κανένα ή ένα)
*	Επί	Κανένα ή περισσότερα
+	Συν	Ένα ή περισσότερα

Τα σύμβολα αυτά επηρεάζουν τον ορισμό ενός στοιχείου όπως φαίνεται στο παράδειγμα ενός DTD που ακολουθεί. Στο παράδειγμα αυτό ορίζουμε τα στοιχεία slide, title, item, και list:

```
<!ELEMENT slideshow (slide+)>
<!ELEMENT slide (title, item*)>
```

```
<!ELEMENT title (#PCDATA)>
<!ELEMENT item (#PCDATA | item)* >
```

Η πρώτη γραμμή ορίζει το στοιχείο της διαφάνειας προβολής (slideshow), που αποτελείται από ένα ή περισσότερα στοιχεία διαφάνειας (slide). Η επόμενη γραμμή δηλώνει ότι το στοιχείο του slide αποτελείται από ένα τίτλο (title) και από κανένα ή περισσότερα στοιχεία τύπου item. Η τρίτη γραμμή δείχνει πως ο τίτλος περιλαμβάνει μόνο δεδομένα χαρακτήρων PCDATA (*parsed character data*), δηλαδή κείμενο, ενώ η τελευταία γραμμή περιέχει μια κάθετη γραμμή (|), η οποία δηλώνει μια συνθήκη *or*. Ο αστερίσκος δηλώνει ότι τα στοιχεία μπορούν να εμφανιστούν καμία ή περισσότερες φορές. Επομένως, σύμφωνα με τη δήλωση αυτή το στοιχείο item μπορεί να περιλαμβάνει είτε καμία ή περισσότερες τιμές τύπου PCDATA είτε κανένα ή περισσότερα στοιχεία τύπου item.

### **Ειδικές τιμές στοιχείων στο DTD**

Αντί για κάποιο στοιχείο να καθοριστεί σε παρένθεση συγκεκριμένη τιμή ή λίστα με στοιχεία μπορεί να δοθεί μια από τις δυο ειδικές τιμές: ANY ή EMPTY. Το ANY δηλώνει ότι το στοιχείο μπορεί να περιέχει PCDATA ή κάθε άλλο ορισμένο στοιχείο. Μία δήλωση αυτής της μορφής εφαρμόζεται συνήθως για το αρχικό στοιχείο ενός εγγράφου XML, καθώς και για οποιοδήποτε στοιχείο που περιλαμβάνει υπο-στοιχεία τα οποία μπορούν να εμφανιστούν με οποιαδήποτε σειρά στο έγγραφο XML.

Το EMPTY δηλώνει ότι το στοιχείο δεν έχει περιεχόμενα. Έτσι, το DTD για το XML που παρουσιάστηκε στην παράγραφο 6.5.1 θα περιέχει τη γραμμή:

```
<!ELEMENT flag EMPTY>
```

### **6.8.3 Προσδιορισμός χαρακτηριστικών στο DTD**

Για τη δήλωση χαρακτηριστικών ενός στοιχείου χρησιμοποιείται η ετικέτα ATTLIST. Η δήλωση ATTLIST ακολουθείται από το όνομα του στοιχείου στο οποίο αναφέρεται και από μία ακολουθία δηλώσεων των χαρακτηριστικών, όπως φαίνεται στο παρακάτω παράδειγμα:

```
<!ELEMENT book (chapters+)>
<!ATTLIST book
           title      CDATA      #REQUIRED
           date       CDATA      #IMPLIED
           author     CDATA      "unknown" >
<!ELEMENT chapters (title)>
<!ELEMENT title (#PCDATA)>
```

Κάθε χαρακτηριστικό ορίζεται από μια σειρά τριών δηλώσεων που χωρίζονται με κενά. Η πρώτη δήλωση προσδιορίζει το όνομα του χαρακτηριστικού, η δεύτερη τον τύπο της τιμής του και η τρίτη μία προκαθορισμένη τιμή. Σε περίπτωση που δεν υπάρχει κάποια προκαθορισμένη τιμή μπορούν να χρησιμοποιηθούν οι λέξεις κλειδιά , που φαίνονται στον Πίνακα 6.8.2.

**Πίνακας 6.8.2 – Προκαθορισμένες τιμές και λέξεις κλειδιά για τα χαρακτηριστικά.**

Δήλωση	Περιγραφή
--------	-----------



"defaultValue"	Η προκαθορισμένη τιμή που θα χρησιμοποιήσει, αν στο έγγραφο δεν έχει οριστεί κάποια τιμή για το χαρακτηριστικό
#REQUIRED	Δηλώνει ότι η τιμή του χαρακτηριστικού πρέπει να καθοριστεί στο έγγραφο.
#IMPLIED	Δηλώνει ότι η τιμή δεν χρειάζεται να οριστεί στο έγγραφο. Αν δεν οριστεί, τότε η εφαρμογή θα έχει κάποια προκαθορισμένη τιμή που θα χρησιμοποιήσει.
#FIXED "fixedValue"	Η τιμή που θα δοθεί στο χαρακτηριστικό. Αν το έγγραφο ορίζει κάποια τιμή, τότε θα πρέπει να συμπίπτει με αυτή.

Οι τύποι των χαρακτηριστικών, που μπορούν να δοθούν σε ένα στοιχείο, περιγράφονται στον Πίνακα 6.8.3.

**Πίνακας 6.8.3 - Τύποι χαρακτηριστικών**

Τύπος χαρακτηριστικού	Καθορίζει
(value1   value2   ...)	Μια λίστα τιμών που χωρίζονται από κάθετες γραμμές. Το χαρακτηριστικό παίρνει μία από αυτές τις τιμές.
CDATA	"Unparsed character data" (ή απλά κείμενο αλφαριθμητικών χαρακτήρων)
ID	Τιμή που προσδιορίζει μοναδικά το στοιχείο.
IDREF	Μια αναφορά σε ένα ID ενός άλλου στοιχείου.
IDREFS	Μια λίστα από μια ή περισσότερες αναφορές σε ID. Οι αναφορές χωρίζονται μεταξύ τους με κενά.
ENTITY	Το όνομα μιας οντότητας που ορίζεται στο DTD
ENTITIES	Μια λίστα οντοτήτων που χωρίζονται με κενά
NMTOKEN	Ένα έγκυρο XML όνομα
NMTOKENS	Μια λίστα έγκυρων XML ονομάτων που χωρίζονται μεταξύ τους με κενά
NOTATION	Ένα όνομα που συμβολίζει μία συγκεκριμένη ενέργεια. Ο συσχετισμός του ονόματος και της ενέργειας γίνεται στο DTD μέσω μιας δήλωσης <!NOTATION ...>.

Όταν ο τύπος του χαρακτηριστικού αποτελείται από μία λίστα με επιλογές χωρισμένες με κάθετες γραμμές μέσα σε μια παρένθεση το χαρακτηριστικό παίρνει μια από αυτές τις τιμές. Για παράδειγμα:

```
<!ELEMENT book (title, item*)>
<!ATTLIST book
  type    (tech | exec | all) #IMPLIED
>
<!ELEMENT title (#PCDATA)>
<!ELEMENT item (#PCDATA | item)* >
```



Στο παράδειγμα αυτό ορίζεται ένα χαρακτηριστικό type του στοιχείου book το οποίο μπορεί να πάρει μία από τις τιμές "tech", "exec", ή "all". Η τελευταία καταχώρηση (#IMPLIED), στη δήλωση του χαρακτηριστικού, δηλώνει πως η τιμή του χαρακτηριστικού δεν είναι απαραίτητο να έχει προσδιοριστεί μέσα στο έγγραφο XML.

Η λέξη-κλειδί CDATA δηλώνει ότι οποιοσδήποτε χαρακτήρας δεδομένων μπορεί να εμφανιστεί ως μέρος της τιμής ενός χαρακτηριστικού. Στο παράδειγμα της δήλωσης που ακολουθεί το χαρακτηριστικό name του στοιχείου person μπορεί να πάρει οποιαδήποτε τιμή που περιλαμβάνει αλφαριθμητικούς χαρακτήρες. Η λέξη κλειδί #REQUIRED δηλώνει πως η τιμή του name θα πρέπει απαραίτητως να οριστεί μέσα στο έγγραφο XML.

```
< !ATTLIST person name CDATA #REQUIRED>
```

#### 6.8.4 Ορισμός οντοτήτων στο DTD

Σε ένα DTD μπορεί να προσδιοριστούν οντότητες που δεν έχουν προκαθοριστεί από την XML. Οι οντότητες επιτρέπουν τον προσδιορισμό αναφορών οντοτήτων που αντικαθιστούν ένα σύνολο από χαρακτήρες στην XML με άλλους χαρακτήρες.

Όπως αναφέρθηκε στην παράγραφο 6.6, η XML υποστηρίζει προκαθορισμένες αναφορές οντοτήτων (&lt;, &gt;, &amp;, &apos; και &quot;) για πέντε συγκεκριμένους χαρακτήρες (<, >, &, ' και "). Μέσω το υ DTD δίνεται η δυνατότητα προσδιορισμού νέων αναφορών οντοτήτων. Η σύνταξη για τον προσδιορισμό μιας οντότητας φαίνεται στο παράδειγμα σύνταξης που ακολουθεί:

```
<!ENTITY name "replacement characters">
```

Η λέξη ENTITY δηλώνει ότι πρόκειται για ορισμό μιας οντότητας. Στη συνέχεια δηλώνεται το όνομα της οντότητας, και τέλος το κείμενο που θα την αντικαταστήσει, όπου αυτή αναφέρεται στο έγγραφο XML. Το κείμενο αντικατάστασης περιέχεται σε διπλά εισαγωγικά.

Στο παράδειγμα που ακολουθεί δημιουργούμε μία αναφορά οντότητας για τον χαρακτήρα © που συμβολίζει πνευματικά δικαιώματα.

```
<!ENTITY copyright "&#xA9;">
```

Έχουμε αντιστοιχίσει την οντότητα &copyright; με τη δεκαεξαδική τιμή 0xA9, που είναι ο χαρακτήρας ©. Με τον προσδιορισμό της οντότητας αυτής στο έγγραφο XML μπορεί να προστεθεί το ακόλουθο περιεχόμενο σε μία ετικέτα :

```
<copyright> &copyright; 2003 University of Piraeus </copyright>
```

Εκτός από αναφορές οντοτήτων για αντικατάσταση μεμονωμένων χαρακτήρων σε ένα DTD μπορούν, επίσης, να οριστούν αναφορές και σε άλλες οντότητες, όπως φαίνεται στο παράδειγμα που ακολουθεί:

```
<!DOCTYPE bookstore SYSTEM "bookstore1.dtd" [  
  <!ENTITY book "Programming">  
>
```

Στη δήλωση αυτή ορίστηκε μία οντότητα με όνομα "book" που θα αντικατασταθεί από κάποιο τίτλο. Ο τίτλος που αντικαθιστά την παράμετρο book στο παραπάνω παράδειγμα δίνεται από τη συμβολοσειρά "Programming". Το πλεονέκτημα μιας δήλωσης αυτής της μορφής είναι ότι όταν αλλάξει ο τίτλος του βιβλίου, θα χρειαστεί να αλλάχτεί μόνο μία τιμή η οποία αυτόματα θα τοποθετηθεί στη θέση της οντότητας book.

### 6.8.5 Παραπομπή σε εξωτερικές οντότητες

Η αναφορά σε εξωτερικές οντότητες επιτυγχάνεται με την ακόλουθη σύνταξη:

```
<!ENTITY entity-name SYSTEM "URI/URL">
```

Η λέξη SYSTEM χρησιμοποιείται για να ορίσει το URI ενός εξωτερικού αρχείου το οποίο θα αντικαταστήσει την οντότητα που προσδιορίστηκε με το όνομα entity-name.

Για παράδειγμα το τμήμα του DTD που ακολουθεί περιλαμβάνει μία παραπομπή/αναφορά σε εξωτερική οντότητα:

```
<!DOCTYPE bookstore SYSTEM "bookstore.dtd" [  
  <!ENTITY thematicarea "Computer Science">  
  <!ENTITY booklist SYSTEM "http://.../books/booklist.xml">  
>
```

Η αναφορά στην εξωτερική οντότητα δηλώνει πως όταν μέσα στο έγγραφο XML εμφανιστεί η οντότητα BookList αυτή θα αντικατασταθεί από τα περιεχόμενα του αρχείου booklist.xml που βρίσκονται στη διεύθυνση "http://.../books/".

Εάν έχουμε ένα έγγραφο XML της μορφής:

```
<bookstore>  
</bookstore>  
<bookarea> &thematicarea;</bookarea>  
&booklist;  
</bookstore>
```

η οντότητα της μορφής &thematicarea; θα αντικατασταθεί από τη συμβολοσειρά "Computer Science", ενώ η &booklist; θα αντικατασταθεί από τον κώδικα XML που περιλαμβάνεται στο αρχείο booklist.xml.

## 6.9 Παρουσίαση εγγράφων XML

Ο κώδικας ενός εγγράφου XML μπορεί να φανεί σε φυλλομετρητές όπως ο Netscape και ο Internet Explorer, με τονισμένη τη δομή του εγγράφου και τα στοιχεία εμφανίζονται με διαφορετικό χρώμα. Για να μπορέσει όμως το περιεχόμενο του εγγράφου να εμφανιστεί σε μία ιστοσελίδα, θα πρέπει το αρχείο της XML να τροποποιηθεί και να προστεθούν σε αυτό στοιχεία μορφοποίησης.

Όπως αναφέρθηκε στην αρχή του κεφαλαίου, η XML, αντίθετα από την HTML, δίνει έμφαση στη δόμηση του περιεχομένου και όχι στον τρόπο που αυτό θα εμφανίζεται. Μία από τις βασικές αρχές της XML είναι ότι το περιεχόμενο θα πρέπει να είναι ξεχωριστό από την παρουσίασή του, με αποτέλεσμα ο σχεδιασμός ενός εγγράφου XML να μην καθορίζεται μέσα στο ίδιο το έγγραφο ή στο DTD.

Το περιεχόμενο μπορεί να εμφανιστεί σε μία ιστοσελίδα μόνο εάν ενσωματωθεί σε μία σελίδα HTML με τη μορφή “data islands” ή εάν συνδυαστεί με άλλες τεχνολογίες, όπως οι τεχνολογίες CSS και XSL. Στο κεφάλαιο αυτό θα γίνει μία σύντομη αναφορά στη δυνατότητα εμφάνισης του περιεχομένου ενός εγγράφου XML με τη βοήθεια αρχείων και γλωσσών προσδιορισμού στυλ (style sheets).

### **CSS - To Style Sheet της HTML**

Το πρότυπο CSS αποτελεί τον κύριο τρόπο μορφοποίησης σελίδων HTML. Μπορεί, επίσης να χρησιμοποιηθεί για τον προσδιορισμό στυλ των περιεχομένων ενός εγγράφου XML. Η χρήση όμως των απλών CSS για τη μορφοποίηση αρχείων XML δε φαίνεται να έχει ιδιαίτερη απήχηση.

Η HTML χρησιμοποιεί προκαθορισμένες ετικέτες, των οποίων το νόημα είναι γνωστό: Για παράδειγμα το στοιχείο <p> ορίζει μια παράγραφο ενώ το στοιχείο <h1> μια επικεφαλίδα και ο φυλλομετρητής γνωρίζει πώς θα τα εμφανίσει. Επομένως, η εμφάνιση κάθε στοιχείου με συγκεκριμένη γραμματοσειρά ή χρώμα καθώς και η προσθήκη στυλ στα στοιχεία της HTML με το CSS, μπορεί να υλοποιηθεί με απλό τρόπο.

Αντίθετα η XML δεν χρησιμοποιεί προκαθορισμένες ετικέτες και το νόημά τους δεν είναι άμεσα κατανοητό. Αυτό έχει ως αποτέλεσμα η περιγραφή της εμφάνισης ενός XML αρχείου μέσω απλών CSS να μην είναι επαρκής. Γι’ αυτό το λόγο αναπτύχθηκε η XSL, ως μία καταλληλότερη γλώσσα προσδιορισμού στυλ (stylesheet language)

### **XSL – XML stylesheets**

Η XSL (EXtensible Stylesheet Language) είναι, όπως δηλώνει και το όνομά της, μία επεκτάσιμη γλώσσα προσδιορισμού στυλ. Αναπτύχθηκε από τον οργανισμό W3C ως μια γλώσσα κατάλληλη για τη μορφοποίηση και τον προσδιορισμό στυλ ενός αρχείου XML. Επομένως, το πώς θα παρουσιαστεί ένα έγγραφο XML μπορεί να καθοριστεί σε ένα φύλλο στυλ που είναι κατασκευασμένο με XSL. Ένα έγγραφο μπορεί να έχει περισσότερα του ενός XSL φύλλα στυλ, που θα οδηγούν σε διαφορετικούς τρόπους παρουσίασης του εγγράφου.

## **6.10 XSL**

Όπως προαναφέρθηκε η XSL είναι μία γλώσσα μορφοποίησης και προσδιορισμού στυλ εγγράφων XML, που κάνει δυνατή τη μετάφραση των εγγράφων XML σε HTML και ειδικότερα σε XHTML. Η XSL αποτελείται από τρία μέρη: το XSLT, το XPath, και η XSL-FO.

- Η XSLT είναι μία γλώσσα για τη μεταμόρφωση/ τροποποίηση των εγγράφων XML
- Η XPath είναι μια γλώσσα που ορίζει τμημάτων ενός εγγράφου XML και
- Η XSL-FO μία γλώσσα για τη μορφοποίηση των εγγράφων XML

Η XSL επομένως μπορεί να θεωρηθεί ως ένα σύνολο από γλώσσες που

- τροποποιεί την XML σε XHTML
- φιλτράρει και να ταξινομήσει δεδομένα XML
- ορίζει τμήματα ενός εγγράφου XML
- μορφοποιεί δεδομένα XML βασισμένα στην τιμή των δεδομένων, (π.χ. εμφανίζει τους αριθμούς που είναι μικρότεροι από 100 με χρώμα μπλε) και
- εξάγει δεδομένα XML σε διάφορες συσκευές (οθόνη, χαρτί ή φωνή).

## Η γλώσσα XSLT

Η XSLT είναι μία γλώσσα για την τροποποίηση των εγγράφων XML σε άλλα έγγραφα XML, ή σε άλλο τύπο εγγράφου, όπως (X)HTML, που μπορεί να αναγνωριστεί από ένα φυλλομετρητή. Η XSLT τροποποιεί τη δομή δένδρου ενός εγγράφου XML σε μία νέα δομή δένδρου στο τελικό έγγραφο.

Η XSLT έχει περισσότερες δυνατότητες από τα CSS, που απλά αλλάζουν τον τρόπο εμφάνισης των στοιχείων σε μία ιστοσελίδα. Η XSLT επιτρέπει την αλλαγή της σειράς των στοιχείων ενός εγγράφου καθώς και την προσθήκη ή αφαίρεση κάποιων στοιχείων από το έγγραφο. Επίσης, δίνει τη δυνατότητα επιλογής των στοιχείων που πρόκειται να εμφανιστούν.

Στη διαδικασία τροποποίησης, η XSLT χρησιμοποιεί το XPath για να ορίσει τμήματα του αρχικού εγγράφου που ταιριάζουν με μία ή περισσότερες προκαθορισμένες φόρμες / πρότυπα (templates). Η XSLT θα μετασχηματίσει τα τμήματα του αρχικού εγγράφου που ταιριάζουν με πρότυπα και θα τα τοποθετήσει στο τελικό έγγραφο. Τα τμήματα του αρχικού εγγράφου που δεν ταιριάζουν με κάποιο πρότυπο θα καταλήξουν μη τροποποιημένα στο τελικό έγγραφο.

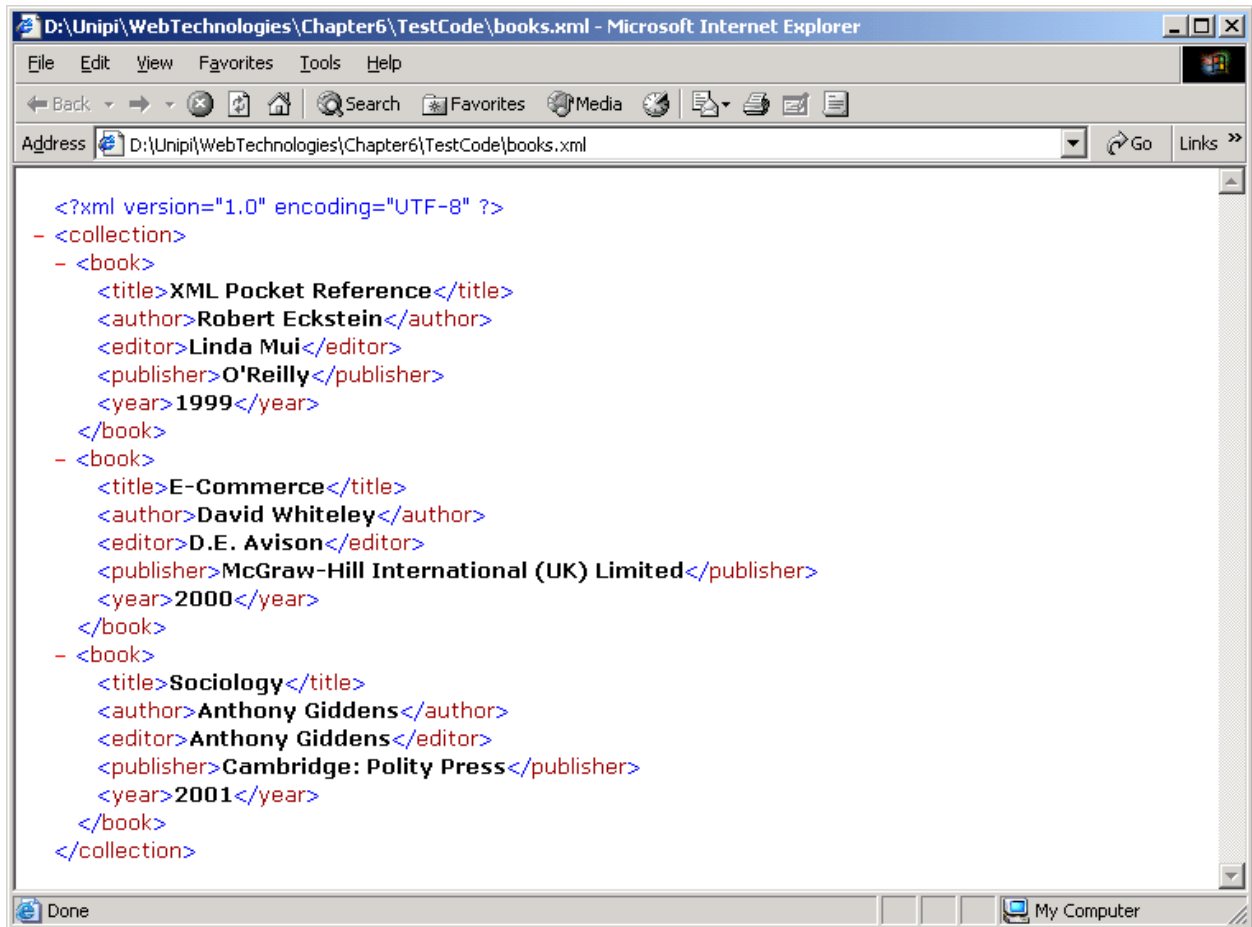
### **6.10.2 Τροποποίηση αρχείων με XSLT**

Η αναλυτική περιγραφή της γλώσσας XSLT δεν αποτελεί στόχο του παρόντος βιβλίου. Σκοπός αυτής της ενότητας είναι να δώσει μία σύντομη περιγραφή του τρόπου με τον οποίο ένα έγγραφο XML μπορεί να τροποποιηθεί σε XHTML ώστε τα περιεχόμενά του να παρουσιαστούν σε ένα φυλλομετρητή.

Έστω ότι θέλουμε να μετασχηματίσουμε το παρακάτω έγγραφο XML ("books.xml") σε XHTML:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="books.xsl"?>
<collection>
  <book>
    <title>XML Pocket Reference</title>
    <author>Robert Eckstein</author>
    <editor>Linda Mui</editor>
    <publisher>O'Reilly</publisher>
    <year>1999</year>
  </book>
  <book>
    <title>E-Commerce</title>
    <author>David Whiteley</author>
    <editor>D.E. Avison</editor>
    <publisher>McGraw-Hill International (UK) Limited</publisher>
    <year>2000</year>
  </book>
  <book>
    <title>Sociology </title>
    <author>Anthony Giddens</author>
    <editor>Anthony Giddens</editor>
    <publisher>Cambridge: Polity Press</publisher>
    <year>2001</year>
  </book>
</collection>
```

Όταν ανοιχτεί το αρχείο αυτό σε ένα φυλλομετρητή, θα φανεί ο παραπάνω κώδικας. Ο κώδικας παρουσιάζεται με τέτοιο τρόπο ώστε η δομή των δεδομένων να είναι εμφανής. Οι ετικέτες χρωματίζονται με διαφορετικό χρώμα από τα περιεχόμενα και οι δηλώσεις εμφανίζονται με άλλη γραμματοσειρά.



**Εικόνα 6.10.1 – Παρουσίαση του εγγράφου XML σε ένα φυλλομετρητή Internet Explorer 6.**

Οι φυλλομετρητές μπορούν επίσης να εμφανίσουν λάθη του κώδικα.

Για να παρουσιαστούν τα περιεχόμενα του εγγράφου σε μία ιστοσελίδα θα πρέπει να δημιουργηθεί ένα αρχείο XSL το οποίο θα καθορίσει τον τρόπο με τον οποίο τα στοιχεία του εγγράφου XML θα εμφανιστούν στην ιστοσελίδα. Ένα παράδειγμα αρχείου XSL που θα μπορούσε να χρησιμοποιηθεί για την τροποποίηση του εγγράφου XML, παρουσιάζεται στο πλαίσιο που ακολουθεί:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
  <body>
    <h2>A Collection of Books</h2>
    <table border="1">
      <tr bgcolor="#9a0532">
        <th align="left">Title</th>
        <th align="left">Author</th>
```

```

        <th align="left">Editor</th>
        <th align="left">Publisher</th>
        <th align="left">Year</th>
    </tr>
    <xsl:for-each select="collection/book">
        <tr>
            <td><xsl:value-of select="title"/></td>
            <td><xsl:value-of select="author"/></td>
            <td><xsl:value-of select="editor"/></td>
            <td><xsl:value-of select="publisher"/></td>
            <td><xsl:value-of select="year"/></td>
        </tr>
    </xsl:for-each>
</xsl:template>
</xsl:stylesheet>

```

Το XSL αρχείο που δημιουργήθηκε διασυνδέεται με το έγγραφο XML με την προσθήκη της ακόλουθης γραμμής κώδικα:

```
<?xml-stylesheet type="text/xsl" href="books.xsl"?>
```

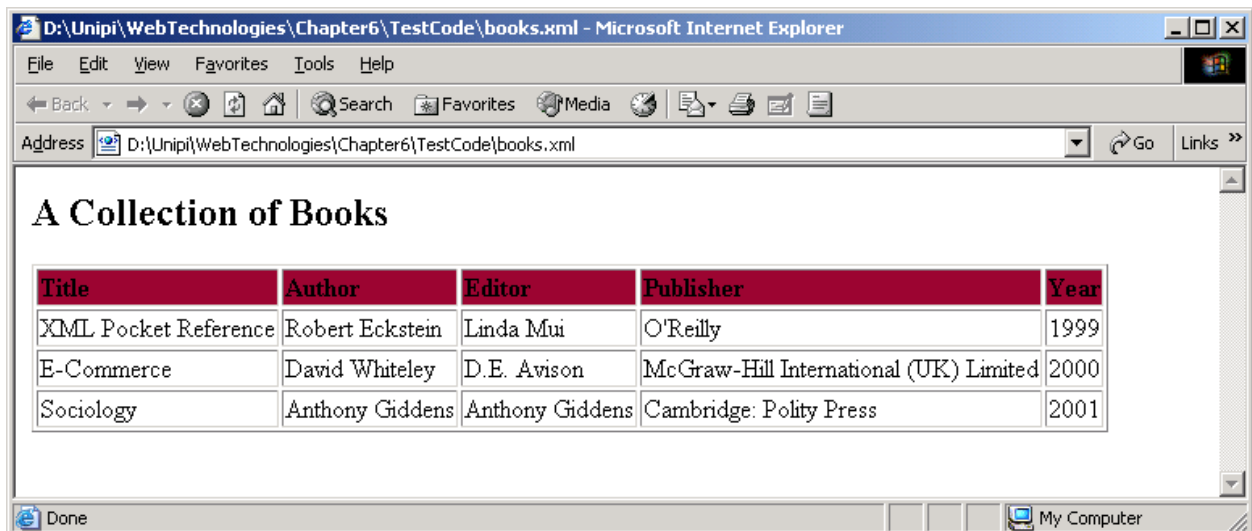
Η δήλωση αυτή προστίθεται αμέσως μετά τη δήλωση της xml και ο κώδικας του εγγράφου XML γίνεται:

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="books.xsl"?>
<collection>
    <book>
        <title>XML Pocket Reference</title>
        <author>Robert Eckstein</author>
        <editor>Linda Mui</editor>
        <publisher>O'Reilly</publisher>
        <year>1999</year>
    </book>
    <book>
        <title>E-Commerce</title>
        <author>David Whiteley</author>
        <editor>D.E. Avison</editor>
        <publisher>McGraw-Hill International (UK) Limited</publisher>
        <year>2000</year>
    </book>
    <book>
        <title>Sociology </title>
        <author>Anthony Giddens</author>
        <editor>Anthony Giddens</editor>
        <publisher>Cambridge: Polity Press</publisher>
        <year>2001</year>
    </book>
</collection>

```

Όταν αυτό το έγγραφο XML φορτωθεί σε κάποιο φυλλομετρητή συμβατό με XML, θα εμφανιστεί ως μία ιστοσελίδα. Θα έχει δηλαδή μετατραπεί σε XHTML.



Εικόνα 6.10.2 – Εμφάνιση του τροποποιημένου αρχείου XML ως ιστοσελίδα.

### 6.10.3 Στοιχεία της XML

#### Στοιχείο ρίζας

Το έγγραφο XSL ξεκινάει με μία δήλωση xml. Στη συνέχεια υπάρχει ένα αρχικό στοιχείο / ρίζα το οποίο δηλώνει ότι πρόκειται για ένα αρχείο στυλ XSL. Το στοιχείο αυτό μπορεί να είναι είτε `<xsl:stylesheet>` είτε `<xsl:transform>`. Τα στοιχεία αυτά είναι συνώνυμα και ότι κι αν χρησιμοποιηθεί από τα δύο, το αποτέλεσμα είναι το ίδιο.

Ο σωστός τρόπος για να ορίσουμε ένα XSL style sheet σύμφωνα με τον οργανισμό W3C είναι:

```
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

ή:

```
<xsl:transform version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

Το `xmlns:xsl="http://www.w3.org/1999/XSL/Transform"` προσδιορίζει την επίσημη περιοχή (namespace) του ονόματος xsl, που τοποθετείται μπροστά από όλα τα στοιχεία της XSL (π.χ. `xsl:template`, `xsl:select`, `xsl:choose`). Η προσθήκη ενός ονόματος στην αρχή κάθε στοιχείου και ο προσδιορισμός της περιοχής του ονόματος έχουν ως στόχο την εξασφάλιση της μοναδικότητας των στοιχείων. Αν χρησιμοποιηθεί αυτή η περιοχή ονόματος, θα πρέπει στην ετικέτα `xsl:transform`, να συμπεριληφθεί το χαρακτηριστικό `version="1.0"`.

Αν έχουμε την έκδοση 6 των φυλλομετρητών Internet Explorer ή Netscape, τότε χρησιμοποιούμε τις παραπάνω γραμμές για την δήλωση ενός XSL style sheet. Υπάρχει, όμως, και η δήλωση:

```
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/TR/WD-xsl">
```



η οποία, όμως, είναι παλιά και την χρησιμοποιούμε μόνο όταν έχουμε τον φυλλομετρητή IE 5.

Ανάμεσα στις ετικέτες `<xsl:stylesheet>` και `</xsl:stylesheet>` τοποθετούνται οι κανόνες που θα εφαρμοστούν στο έγγραφο XML. Οι κανόνες αυτοί χωρίζονται σε δύο κατηγορίες σε αυτούς που περιγράφουν το πρότυπο σχήμα εγγράφου (template pattern) και σε αυτούς που προσδιορίζουν την εφαρμογή του προτύπου (template action).

### **Το στοιχείο <xsl:template>**

Η ετικέτα `<xsl:template>` ορίζει την αρχή ενός προτύπου σχήματος εγγράφου. Περιέχει κανόνες που προσδιορίζουν το πρότυπο σχήμα του εγγράφου και θα εφαρμοστούν στα στοιχεία του αρχείου XML. Το χαρακτηριστικό `match` χρησιμοποιείται για να προσδιορίσει το στοιχείο ή το σύνολο των στοιχείων XML στο οποίο θα εφαρμοστεί το πρότυπο. Στο παράδειγμα της προηγούμενης ενότητας χρησιμοποιήθηκε η παρακάτω ετικέτα `template` με τιμή χαρακτηριστικού `match` ίση με `"/`.

```
<xsl:template match="/">
```

Η τιμή αυτή του χαρακτηριστικού `match` αναφέρεται στη ρίζα του στοιχείου XML και αφορά ολόκληρο το έγγραφο XML.

### **Το στοιχείο <xsl:value-of>**

Το στοιχείο `<xsl:value-of>` αποσπά την τιμή ενός επιλεγμένου στοιχείου XML. Χρησιμοποιείται για να ορίσει την τιμή ενός στοιχείου XML, που πρόκειται να τροποποιηθεί και να προστεθεί στο τελικό έγγραφο:

```
<xsl:value-of select="collection/book"/>
```

Η τιμή του χαρακτηριστικού `select` περιέχει μια έκφραση Xpath, η οποία προσδιορίζει τη θέση του στοιχείου σε μία δομή XML.

### **Το στοιχείο <xsl:for-each>**

Το στοιχείο `<xsl:for-each>` επιτρέπει τη δημιουργία βρόχων και την επανάληψη μιας εντολής τροποποίησης στην XSL. Χρησιμοποιείται για την επιλογή μιας σειράς από στοιχεία του ίδιου τύπου, στα οποία θα εφαρμόσει τους ίδιους κανόνες.

```
<xsl:for-each select="collection/book">
  <tr>
    <td><xsl:value-of select="title"/></td>
    <td><xsl:value-of select="author"/></td>
    <td><xsl:value-of select="editor"/></td>
    <td><xsl:value-of select="publisher"/></td>
    <td><xsl:value-of select="year"/></td>
  </tr>
</xsl:for-each>
```

Και σε αυτήν την περίπτωση η τιμή του χαρακτηριστικού 'select' περιέχει μια έκφραση XPath.

### **Το στοιχείο <xsl:sort>**



Το στοιχείο `<xsl:sort>` χρησιμοποιείται για την ταξινόμηση των δεδομένων. Με την εφαρμογή του στοιχείου αυτού τα δεδομένα του εγγράφου XML θα τροποποιηθούν και θα τοποθετηθούν ταξινομημένα στο τελικό έγγραφο XHTML. Η σύνταξη της δήλωσης αυτής είναι η ακόλουθη:

```
<xsl:sort select="keyword"/>
```

Το χαρακτηριστικό `select` δηλώνει το στοιχείο XML με βάση το οποίο γίνεται η ταξινόμηση.

### **Το στοιχείο `<xsl:if>`**

Το στοιχείο `<xsl:if>` περιέχει ένα πρότυπο που θα εφαρμοστεί μόνο αν μία καθορισμένη συνθήκη είναι αληθής.

```
<xsl:if test="size>'10'">
  some output ...
</xsl:if>
```

Η τιμή του χαρακτηριστικού `test` περιέχει τη συνθήκη που θα ελεγχθεί.

Ένα παράδειγμα ενός XSL style sheet είναι το ακόλουθο:

```
<xsl:for-each select="stock/shirt">
  <xsl:if test="size>10">
    <tr>
      <td><xsl:value-of select="designer"/></td>
    </tr>
  </xsl:if>
</xsl:for-each>
```

Ο παραπάνω κώδικας διαλέγει μόνο το σχεδιαστή αν το μέγεθος του shirt στην αποθήκη είναι μεγαλύτερο από 10.

### **Το στοιχείο `<xsl:choose>`**

Το στοιχείο `<xsl:choose>` χρησιμοποιείται σε συνδυασμό με τα `<xsl:when>` και `<xsl:otherwise>` για την έκφραση υπό-συνθήκη εντολών. Για να εισάγουμε μία επιλογή με συνθήκη έναντι του περιεχομένου του αρχείου XML, θα προσθέσουμε τα `<xsl:choose>`, `<xsl:when>`, και `<xsl:otherwise>` στο έγγραφο XSL ως εξής:

```
<xsl:choose>
  <xsl:when test="size>'10'">
    ... some code ...
  </xsl:when>
  <xsl:otherwise>
    ... some code ....
  </xsl:otherwise>
</xsl:choose>
```

## Το στοιχείο <xsl:apply-templates>

Το στοιχείο <xsl:apply-templates> εφαρμόζει έναν πρότυπο κανόνα στο τρέχον στοιχείο ή στα παιδιά– υποστοιχεία του τρέχοντος στοιχείου.

Αν προσθέσουμε ένα χαρακτηριστικό select με τιμή το όνομα ενός υποστοιχείου στοιχείου στο στοιχείο <xsl:apply-templates> θα γίνει επεξεργασία μόνο του παιδιού – κόμβου που ταιριάζει στην τιμή του χαρακτηριστικού. Μπορούμε επίσης να χρησιμοποιήσουμε το χαρακτηριστικό select για τον καθορισμό της σειράς με την οποία θα επεξεργαστούν τα παιδιά – κόμβοι.

## Πως φιλτράρουμε τα δεδομένα εξόδου

Μπορούμε να φιλτράρουμε τα δεδομένα εξόδου από ένα αρχείο XML, προσθέτοντας ένα κριτήριο στο χαρακτηριστικό select στο στοιχείο <xsl:for-each>:

```
<xsl:for-each select="catalog/cd[artist='Madonna']">
```

Τελεστές που μπορούμε να χρησιμοποιήσουμε είναι:

- = (equal)
- != (not equal)
- &lt; (less than)
- &gt; (greater than)

## 6.11 Ερωτήσεις – Ασκήσεις – Θέματα για ανάπτυξη

Ερωτήσεις στην XML	
Τι σημαίνουν τα αρχικά XML;	Example Markup Language eXtra Modern Link eXtensible Markup Language x-Markup Language
Υπάρχει ένας τρόπος για να περιγραφούν δεδομένα XML, ποιός;	Η XML χρησιμοποιεί ένα DTD για να περιγράψει τα δεδομένα. Η XML χρησιμοποιεί ένα XSL για να περιγράψει τα δεδομένα. Η XML χρησιμοποιεί ένα κόμβο περιγραφής για να περιγράψει τα δεδομένα.
Σκοπός της XML είναι να αντικαταστήσει την HTML.	Σωστό Λάθος
Ποιά είναι η σωστή σύνταξη της δήλωσης που ορίζει μία έκδοση XML;	<?xml version="1.0"?> <xml version="1.0"/> <?xml version="1.0"/>
Τι σημαίνουν τα αρχικά DTD;	Dynamic Type Definition

	Data Type Definition Direct Type Definition Document Type Definition
Είναι σωστό το κείμενο XML: <?xml version="1.0"?> <note> <to>Tania</to> <from>John</from> <heading>Reminder</heading> <body>Do not forget the meetintg!</body> </note>	Ναι Όχι
Ποια πρόταση είναι σωστή;	Όλες οι προτάσεις είναι σωστές Όλα τα στοιχεία XML πρέπει να γράφονται με πεζούς χαρακτήρες. Όλα τα κείμενα XML πρέπει να έχουν ένα DTD Όλα τα στοιχεία XML πρέπει να έχουν μία ετικέτα τέλους
Ποια πρόταση είναι σωστή;	Τα στοιχεία XML πρέπει να είναι σωστά εμφωλιασμένα. Τα κείμενα XML πρέπει να έχουν μία ετικέτα ρίζας. Παίζει ρόλο εάν οι XML ετικέτες γράφονται με πεζούς ή κεφαλαίους χαρακτήρες. Όλες οι προτάσεις είναι σωστές
Είναι σωστό το κείμενο XML: <?xml version="1.0"?> <note> <to age="29">Tania</to> <from>John</from> </note>	Ναι Όχι
Είναι σωστό το κείμενο XML: <?xml version="1.0"?> <note> <to age=29>Tania</to> <from>John</from> </note>	Ναι Όχι
Τα στοιχεία XML δεν μπορεί να είναι κενά.	Σωστό Λάθος
Ποιο δεν είναι σωστό όνομα για ένα στοιχείο XML;	Και τα 3 ονόματα είναι λανθασμένα <Note> <1dollar> <h1>
Ποιο δεν είναι σωστό όνομα για ένα στοιχείο XML;	Και τα 3 ονόματα είναι λανθασμένα <age> <NAME> <first name>

Ποιο δεν είναι σωστό όνομα για ένα στοιχείο XML;	<phone number> Και τα 3 ονόματα είναι λανθασμένα <7eleven> <xmldocument>
Οι τιμές των χαρακτηριστικών XML πρέπει πάντα να περικλείονται σε εισαγωγικά.	Σωστό Λάθος
Τι σημαίνουν τα αρχικά XSL;	eXtensible Style Listing eXtensible Stylesheet Language eXtra Style Language eXpandable Style Language
Ποιος είναι ο σωστός τρόπος αναφοράς σε ένα αρχείο προσδιορισμού στυλ;	<?xml-stylesheet type="text/xsl" href="mystyle.xml"?> <stylesheet type="text/xsl" href="mystyle.xml"/> <link type="text/xsl" href="mystyle.xml"/>
Ποια είναι η σωστή σύνταξη ώστε ο αναλυτής της XML να αγνοεί ένα συγκεκριμένο τμήμα του κειμένου XML;	<PCDATA> Text to be ignored </PCDATA> <CDATA> Text to be ignored </CDATA> <![CDATA[ Text to be ignored]]> <xml:CDATA[ Text to be ignored]>

### Θέματα για ανάπτυξη

1. Τι είναι και πού χρησιμοποιείται η XML;
2. Ποιες οι διαφορές της XML και της HTML;
3. Πότε χρησιμοποιούμε στοιχεία και πότε χαρακτηριστικά για την απεικόνιση δεδομένων στην XML;
4. Πότε ένα έγγραφο XML θεωρείται έγκυρο;
5. Τι είναι το DTD και πού μας χρησιμεύει
6. Τι είναι η γλώσσα XSL;

### Ασκήσεις

1. Δημιουργήστε ένα έγγραφο XML που περιλαμβάνει μία συλλογή από CD. Δημιουργήστε ένα κατάλληλο DTD για την περιγραφή του εγγράφου σας. Χρησιμοποιήστε την XSL για να εμφανίσετε τα στοιχεία αυτά σε μία ιστοσελίδα.
2. Δημιουργήστε μία σελίδα που παρουσιάζει μία λίστα με κινηματογραφικά έργα. Χρησιμοποιήστε τις xsl:select και xsl:choose για να παρουσιάσετε με διαφορετικό χρώμα τα έργα ανάλογα των οποίων τα διαθέσιμα εισιτήρια είναι λιγότερα από 10.