Σχεδίαση Εφαρμογών και Υπηρεσιών Διαδικτύου
9η Διάλεξη: Η γλώσσα XML

Δρ. Απόστολος Γκάμας

Λέκτορας (407/80)

gkamas@uop.gr

Γιατί άλλη μία γλώσσα?

- Η XML χρησιμοποιείται για την ανταλλαγή δεδομένων
  - Επιτρέπει σαφή ορισμό των δεδομένων
  - Όλοι οι συμμετέχοντες «μεταφράζουν» με τον ίδιο τρόπο τα δεδομένα
- Αντικαθιστά το EDI (Electronic Data Interchange)
  - Χρησιμοποιεί το διαδίκτυο για την ανταλλαγή δεδομένων
  - Είναι πιο ευέλικτη
- Επιτρέπει τον ορισμό άλλων γλωσσών
  - WSDL – Web Services Description Language

Τι είναι η XML;
EXtensible Markup Language

- Υποσύνολο της SGML (Standard Generalized Markup Language)
  - Μετα-γλώσσα → κατάλληλη για τον ορισμό άλλων γλωσσών
- Σχεδιάστηκε για τον ορισμό δεδομένων
  - Οι δομές δεδομένων ανεξάρτητες από την πλατφόρμα
  - Εύκολη η αυτόματη επεξεργασία των δεδομένων
  - Ο χρήστης μπορεί να ορίσει τα δικά του tags
- Δεν περιγράφει τον τρόπο εμφάνισης δεδομένων!
  - Ένα XSL αρχείο ορίζει την εμφάνιση ενός XML αρχείου
- Ένα DTD (Document Type Definition) ή ένα XML Schema ορίζει τη σύνταξη ενός XML αρχείου

XML & HTML

- Η XML δεν έχει σκοπό να αντικαταστήσει την HTML, αλλά να την συμπληρώσει
  - Η HTML σχεδιάστηκε για να παρουσιάζει δεδομένα δίνοντας έμφαση στο πώς αυτά φαίνονται
  - Η XML σχεδιάστηκε για να περιγράφει δεδομένα δίνοντας έμφαση στο τι είδος δεδομένα είναι

Ένα απλό XML αρχείο .xml

```
<?xml version = "1.0"?>

<!-- Simple introduction to XML markup -->

<myMessage>
    <message>Welcome to XML!</message>
</myMessage>
```

View page

XML δήλωση

- Η πρώτη γραμμή ενός XML αρχείου

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

  - version e.g. “1.0”
  - encoding
    - προαιρετική τιμή - προεπιλεγμένη UTF-8
  - standalone
    - Δηλώνει αν το XML αρχείο δεν εξαρτάται από άλλα XML αρχεία για να είναι έγκυρο ή όχι
    - Εξ’ ορισμού η τιμή είναι yes (δηλ. ανεξάρτητο αρχείο)

Το συντακτικό της XML(1/2)

- Κάθε tag πρέπει να κλείνει
  - <message> Welcome XML! (no)
- Τα tags είναι case sensitive
  - Το <message> είναι διαφορετικό από το <Message>
- Τα tags πρέπει να είναι εμφωλευμένα σωστά
  - <myMessage> <message> Welcome XML <myMessage> <message> (no)
- Άδεια tags
  - <Name/>

Το συντακτικό της XML (2/2)

- Όλα τα XML αρχεία πρέπει να έχουν **ένα και μόνο** root tag!
- Όλες οι τιμές των ιδιοτήτων πρέπει να βρίσκονται ανάμεσα σε εισαγωγικά
  - <message date="12/01/2004"> (yes)
- Όλοι οι κενοί χαρακτήρες διατηρούνται
- Εισαγωγή σχολίων όπως στην HTML
  - <!-- this is a comment -->

XML Elements (1/2)

- Ένα XML αρχείο είναι επεκτάσιμο
  - Νέα στοιχεία μπορούν να προστεθούν

```
<myMessage>
  <message>Welcome to XML!</message>
  <date>6/12/2004</date>
</myMessage>
```

- Σχέσεις μεταξύ των XML elements
  - myMessage – root element and parent element of message & date
  - message & date – siblings & child elements of myMessage

XML Elements (2/2)

- Το περιεχόμενο των **elements** ποικίλλει
  - Άλλα **elements (child elements)**
  - Μεικτό περιεχόμενο – κείμενο & **elements**
  - Μόνο κείμενο
  - Άδειο **element**
- Δεν υπάρχει περιορισμός στο βάθος του εμφωλιασμού στοιχείων
- Τα ονόματα των **elements**
  - Μπορούν να περιέχουν γράμματα, αριθμητικά ψηφία και άλλους χαρακτήρες
  - Δεν μπορούν να αρχίζουν με αριθμητικό ή σημείο στίξης
  - Δεν μπορούν να αρχίζουν με τα γράμματα xml (ή XML ή Xml)
  - Δεν μπορούν να περιέχουν κενά

XML attributes

- Τα XML elements μπορούν να περιέχουν attributes (ιδιότητες) στο αρχικό tag όπως και στην HTML
  - Τα attributes παρέχουν περισσότερες πληροφορίες για τα elements

    <file type="gif">computer.gif</file>
  - Η τιμή ενός attribute πρέπει να βρίσκεται είτε σε μονά είτε σε διπλά εισαγωγικά
- Οι επιπλέον πληροφορίες για ένα element μπορούν να αποθηκευτούν είτε ως attribute είτε ως child element

Χρήση elements αντί attributes!

- Γιατί να αποφεύγετε τα attributes?
  - Δεν μπορούν να περιέχουν πολλαπλές τιμές
  - Δεν είναι εύκολα επεκτάσιμα
  - Δεν περιγράφουν δομές
  - Διαχειρίζονται δύσκολα από προγράμματα
  - Είναι δύσκολος ο έλεγχος του με βάση ένα DTD
- Πότε να χρησιμοποιείτε attributes?
  - Για πληροφορίες που δεν είναι σχετικές με τα δεδομένα
  - Για την ανάθεση ID στα elements

Προκαθορισμένες οντότητες αναφοράς

- Η εισαγωγή του χαρακτήρα "<" σε ένα XML αρχείο θα παράγει σφάλμα, καθώς ο XML parser θα θεωρήσει ότι αρχίζει ένα καινούριο element
- Η XML υποστηρίζει τις εξής πέντε προκαθορισμένες οντότητες αναφοράς για τους ειδικούς χαρακτήρες:
  - < <
  - > >
  - & &
  - ‘ '
  - " "

XML CDATA

- Όλο το κείμενο ενός XML αρχείου θα επεξεργαστεί από τον XML parser.
- Εάν ο χρήστης επιθυμεί να εισάγει πληροφορία η οποία δεν πρέπει να επεξεργαστεί τότε αυτή πρέπει να εμφανιστεί σε ένα CDATA section
- Ένας CDATA section αρχίζει με το **<![CDATA[** και τελειώνει με το **]]>**
- Σε ένα CDATA δεν επιτρέπεται η ακολουθία χαρακτήρων]]>

```
<script>
  <![CDATA[      function matchwo(a,b) {
                        if (a < b && a < 0)
                        then      { return 1 }
                        else      { return 0 } }
  ]]>
</ script >
```

View page

XML Namespaces (1/3)

- Είναι πιθανό σε δύο διαφορετικά αρχεία να εμφανίζονται **elements** με το ίδιο όνομα, αλλά διαφορετική σημασία
  - πχ. το **element <file>** μπορεί να αναφέρεται είτε σε ένα αρχείο **txt** είτε σε μία εικόνα
- Προστίθεται ένα πρόθεμα μπροστά στο όνομα, καθώς τα ονόματα των **tags** πρέπει να είναι διαφορετικά
  - πχ. **<text:file> & <image:file>**
  - Έτσι προκύπτουν δύο διαφορετικά **elements**
- Το **namespace attribute** έχει την εξής σύνταξη
  - **xmlns:namespace-prefix="namespace"**

XML Namespaces (2/3)

- Προθέματα namespaces, τα οποία είναι Uniform Resource Identifier (URI)

```
<directory xmlns:text="urn:IT:textInfo“>
  <text:file filename="book.xml">
    <text:description>A book list</text:description>
  </text:file>
<directory>
```

View page

XML Namespaces (3/3)

- Είναι δυνατό σε ένα **element** να δηλωθεί ένα **default namespace** με αποτέλεσμα να μην είναι απαραίτητο να τοποθετείται το αντίστοιχο πρόθεμα σε όλα τα παιδιά του στοιχείου αυτού

```
<directory xmlns="urn:IT:textInfo“>
  <file filename="book.xml">
    <description>A book list</description>
  </file>
<directory>
```

View page

Well Formed XML αρχεία

- Ένα XML αρχείο χαρακτηρίζεται ως "Well Formed" όταν η σύνταξή του είναι σωστή
  - Περιέχει ένα μόνο root element
  - Όλα τα elements του είναι παιδιά του root element
  - Όλα τα elements είναι σωστά εμφωλιασμένα
  - Το όνομα των elements είναι το ίδιο στα start-tag και end-tag
  - Ένα attribute εμφανίζεται μόνο μία φορά σε ένα element

Valid XML αρχεία

- Για να χαρακτηριστεί ένα XML αρχείο ως "Valid" πρέπει να ισχύουν τα εξής:
  - Το XML αρχείο να είναι "Well Formed" και
  - Το συντακτικό του να καθορίζεται από ένα DTD (Document Type Definition)
- Δηλ. δεν μπορούν να χρησιμοποιηθούν elements που δεν ορίζονται στο DTD

DTD – Document Type Definition

- Ορισμός των νόμιμων **tags** ενός **XML** αρχείου
- Ορισμός των σχέσεων μεταξύ των **tags**

```
<!ENTITY % Binary “yes | no”>
<!ELEMENT bookstore (book+)>
<!ELEMENT book (title,author,price))>
<!ATTLIST book paperback (%Binary) #REQUIRED
                cdrom (%Binary) #REQUIRED>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT price (#PCDATA)>
```

View xml

Δήλωση DTD μέσα στο XML αρχείο

```
<?xml version="1.0"?>
<!DOCTYPE bookstore [
        <!ENTITY % Binary “yes | no”>
        <!ELEMENT bookstore (book+)>
        <!ELEMENT book (title,author,price))>
        <!ATTLIST book paperback (%Binary) #REQUIRED
                       cdrom (%Binary) #REQUIRED>
        <!ELEMENT title (#PCDATA)>
        <!ELEMENT author (#PCDATA)>
        <!ELEMENT price (#PCDATA)>
]>
<bookstore>
<book paperback="yes" cdrom="no">
<title>A Guide to XML technology</title>
<author>Robert Stewart</author>
<price>30$</price>
</book>
</bookstore>
```

Το DTD σε ξεχωριστό αρχείο

- Δήλωση του κατάλληλου DTD αρχείου στο XML αρχείο

```
<?xml version="1.0"?>
<!DOCTYPE bookstore System “bookstore.dtd”>
<bookstore>
<book paperback="yes" cdrom="no">
<title>A Guide to XML technology</title>
<author>Robert Stewart</author>
<price>30$</price>
</book>
</bookstore>
```

Τα στοιχεία ενός DTD αρχείου

- !ELEMENT
  - Ορίζει ένα element του XML αρχείου
- !ATTLIST
  - Ορίζει τα χαρακτηριστικά ενός element
- !ENTITY
  - Ορίζει μία συντόμευση για ένα όνομα ή μία έκφραση
- PCDATA – parsed character data
  - Το κείμενο μεταξύ των start-tags και end-tags ενός element το οποίο θα επεξεργαστεί από τον XML parser
- CDATA – character data
  - Κείμενο που δε θα επεξεργαστεί από τον XML parser

DTD - ELEMENT

- <!ELEMENT element-name category>
  - <!ELEMENT book EMPTY>
  - <! ELEMENT author ANY>
  - <! ELEMENT title (#PCDATA) >
- <!ELEMENT element-name (element-content)>
  - <!ELEMENT book (title,author,price)>
  - <!ELEMENT bookstore (book+)> - min one
  - <!ELEMENT bookstore (book*)> - zero or more
  - <!ELEMENT bookstore (book?)> - zero or one
  - <!ELEMENT note (body|mesage)> - selection
  - <!ELEMENT note (#PCDATA|body|mesage)*>

DTD – ATTLIST (1/3)

- <!ATTLIST element-name attribute-name attribute-type default-value>
- Attribute type
  - CDATA: string

    <!ATTLIST person name CDATA #REQUIRED>
  - (en1|en2|..): μία επιλογή από μία λίστα απαρίθμησης

    <!ATTLIST payment type (check|cash) "cash">
  - ENTITY: παίρνει τη τιμή από μία οντότητα
  - ENTITIES: λίστα οντοτήτων

    <!ENTITY % Binary “yes | no”>

    <!ATTLIST book paperback (%Binary) #REQUIRED>

DTD – ATTLIST (2/3)

- ID: μοναδικό id
  <!ATTLIST car serial_no ID #REQUIRED>
- IDREF: το id ενός άλλου element & IDREFS: λίστα από άλλα ids
  <!ATTLIST car serial_no ID #REQUIRED>
  <!ATTLIST ford number IDREF #IMPLIED>
- NMTOKEN: ένα valid XML όνομα NMTOKENS: λίστα ονομάτων
  <!ATTLIST car plate_number NMTOKEN #REQUIRED>
- NOTATION: μία σήμανση
  - <!NOTATION Gif System “image/gif”>
- xml: μία προκαθορισμένη xml τιμή

DTD – ATTLIST (3/3)

- Default value
  - Μία προκαθορισμένη τιμή
    <!ATTLIST square width CDATA "0">
  - #REQUIRED: το χαρακτηριστικό είναι απαραίτητο
    <!ATTLIST person name CDATA #REQUIRED>
  - #IMPLIED: το χαρακτηριστικό δεν είναι απαραίτητο και δεν υπάρχει προκαθορισμένη τιμή
    <!ATTLIST contact fax CDATA #IMPLIED>
  - #FIXED: το χαρακτηριστικό μπορεί να πάρει μόνο αυτή τη συγκεκριμένη τιμή
    <!ATTLIST sender uni CDATA #FIXED “Patras">

DTD – ENTITY (1/2)

- Παραμετρική οντότητα
  - Μία συντόμευση για μεγάλο κείμενο που εμφανίζεται πολλές φορές σε ένα DTD αρχείο
  - Πρέπει να δηλωθεί πριν χρησιμοποιηθεί
  - Συνήθως δηλώνεται στην αρχή του DTD αρχείου

    <!ENTITY % όνομα "κείμενο αντικατάστασης">

    <!ENTITY % Binary “yes | no”>
- Γενική οντότητα

  Συντόμευση για κείμενο που εμφανίζεται συχνά σε ένα XML αρχείο

  <!ENTITY όνομα "κείμενο αντικατάστασης">

  <!ENTITY name “Eleni Christopoulou">

DTD – ENTITY (2/2)

- Εσωτερική δήλωση entity
  - <!ENTITY entity-name "entity-value">

    <!ENTITY writer “Eleni Christopoulou">
- Εξωτερική δήλωση entity
  - <!ENTITY entity-name SYSTEM "URI/URL">

    <!ENTITY writer SYSTEM “http:www.eleni.xml”>

    <!ENTITY writer SYSTEM “http:www.eleni.dtd”>

XML Validator


```
xmlDoc = new ActiveXObject("Microsoft.XMLDOM")
xmlDoc.validateOnParse="true“

xmlDoc.load(“message.xml")
xmlDoc.loadXML(… string …)
```

- Απενεργοποίηση XML validator
  xmlDoc.validateOnParse=“false“

View page

XML Schema Definition (XSD)

- Ένα XML Schema μπορεί να χρησιμοποιηθεί αντί για ένα DTD
- Περιγράφει τη δομή ενός XML αρχείου
- Ένα XML Schema μπορεί να διαχειριστεί όπως ένα απλό XML αρχείο

```
<xs:schema>
  ...
</xs:schema>
```

XSD vs DTD

- Πλεονεκτήματα των XSD αρχείων
  - Εύκολα επεκτάσιμα
  - Γραμμένα σε XML
  - Υποστηρίζουν τύπους δεδομένων
    - Καλύτερος έλεγχος δεδομένων, χρήση δεδομένων από βάσεις δεδομένων, μετατροπή μεταξύ διάφορων τύπων, πιο εύκολοι οι περιορισμοί
  - Υποστηρίζουν namespaces
  - Αξιόπιστη επικοινωνία δεδομένων
    - <date type="date">2004-03-11</date>
    - Χρήση του τύπου δεδομένων date με το format YYYY-MM-DD

Ένα απλό XML Schema

```
<?xml version="1.0"?>
    <xs:schema xmlns:xs=http://www.w3.org/2001/XMLSchema
      targetNamespace=http://www.eleni.com
      xmlns=http://www.eleni.com
      elementFormDefault="qualified">
      <xs:element name="note">
            <xs:complexType>
                  <xs:sequence>
                  <xs:element name="to" type="xs:string"/>
                  <xs:element name="from" type="xs:string"/>
                  <xs:element name="heading" type="xs:string"/>
            <xs:element name="body" type="xs:string"/>
      </xs:sequence>
      </xs:complexType>
      </xs:element>
    </xs:schema>
```

Δήλωση XML Schema μέσα σε ένα XML αρχείο

```
<?xml version="1.0"?>
    <note xmlns="http://www.eleni.com"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.eleni.com note.xsd">
      <to>Dim</to>
      <from>Eleni</from>
      <heading>Reminder</heading>
      <body>Don't forget me this weekend!</body>
    </note>
```

View page

<Schema> element

- <schema> - root element
- xmlns:xs="http://www.w3.org/2001/XMLSchema"
  - Χρησιμοποιεί το "http://www.w3.org/2001/XMLSchema" namespace
  - Ό,τι προέρχεται από αυτό το namespace έχει μπροστά το **xs:**
- targetNamespace=http://www.eleni.com
  - Τα elements που ορίζονται σε αυτό το σχήμα προέρχονται από αυτό το namespace
- xmlns=http://www.eleni.com
  - Δηλώνεται το default namespace
- elementFormDefault="qualified"
  - Όλα τα elements πρέπει να έχουν το namespace μπροστά

XSD - elements

- Element: <xs:element name="xxx“ type="yyy"/>
- <xs:element name=“message" type="xs:string"/>
- Types: xs:string | xs:decimal | xs:integer | xs:boolean | xs:date | xs:time
- “Default” και “Fixed” τιμές
  - <xs:element name="color" type="xs:string" default="red"/>
  - <xs:element name="color" type="xs:string" fixed="red"/>

XSD - attributes

- Attribute: <xs:attribute name="xxx" type="yyy"/>
- <xs:attribute name="lang" type="xs:string"/>
- “Default” και “Fixed” τιμές
- use= optional | required
  - <xs:attribute name="lang" type="xs:string" use="optional"/>

XSD - restrictions

```
<xs:element name="age">
<xs:simpleType>
        <xs:restriction base="xs:integer">
                <xs:minInclusive value="0"/>
        <xs:maxInclusive value="100"/>  </xs:restriction>
</xs:simpleType>
</xs:element>
```

- `<xs:enumeration value="Audi"/>`
- `<xs:whiteSpace value="preserve|replace"/>`
- `<xs:length value="8"/>`

XSD – complex elements

```
<xs:element name="employee">
<xs:complexType>
        <xs:sequence>
        <xs:element name="firstname" type="xs:string" />
        <xs:element name="lastname" type="xs:string" />
        </xs:sequence>
</xs:complexType>
</xs:element>
```

- `<xs:enumeration value="Audi" />`
- `<xs:whiteSpace value="preserve|replace" />`
- `<xs:length value="8" />`

Παράδειγμα XML αρχείων

- Το XML Schema
- Ένα XML Αρχείο και Validation
- Ένα δεύτερο XML Αρχείο και Validation (δεν είναι VALID)

Παρουσίαση XML αρχείων

- CSS – Cascading Style Sheets
  - Όπως χρησιμοποιούνται και στην HTML
- XSL - eXtensible Stylesheet Language
  - Μετατρέπουν XML αρχεία σε HTML
  - Είτε client side είτε server side
  - Προτιμότερο server side

XSL - EXtensible Stylesheet Language

- Δεν είναι μία απλή style sheet γλώσσα
- Ένα σύνολο γλωσσών
  - XSLT- μετατροπή XML αρχείων
  - XPath – ορισμός τμημάτων ενός XML αρχείου
  - XSL-FO – μορφοποίηση XML αρχείων & δεδομένων

XSLT

- Μετατρέπει XML αρχεία σε
  - άλλα XML αρχεία
  - αρχεία αναγνωρίσιμα από τους browsers, πχ HTML, XHTML
  - Αντιστοιχώντας σε ένα XML element ένα (X)HTML element
- Προσθέτει ή αφαιρεί καινούρια elements στο παραγόμενο αρχείο
- Ταξινομεί elements
- Ορίζει τον τρόπο αναπαράστασης των δεδομένων
- Το XSLT μετατρέπει ένα XML source tree σε ένα XML result tree

XSLT - συμβατότητα

- Δεν υποστηρίζουν όλοι οι browsers 100% το XSLT
- Internet Explorer 6
  - Διαθέτει τον MSXML Parser 3.0
- & Netscape 7

Δήλωση ενός XSL αρχείου

- Root element

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

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

Ένα απλό XSL αρχείο

```
<?xml version = "1.0"?>
<!-- Simple XSLT document for intro.xml -->
<xsl:stylesheet version = "1.0"
      xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match = "myMessage">
    <html>
      <body><xsl:value-of select = "message"/></body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

Μετατροπή XML αρχείου σε HTML

- Δήλωση μέσα στο XML αρχείο για τη σύνδεση με το XSL αρχείο

```
<?xml:stylesheet type = "text/xsl" href = "intro.xsl"?>
```

XSL αρχείο

XML αρχείο

Τα στοιχεία ενός XSL αρχείου (1/2)

- <xsl:template match="tag-name">…</xsl:template>
  - Ορίζει τον HTML κώδικα για ένα συγκεκριμένο XML tag
- <xsl:value-of select="tag-name">… </xsl:value-of>
  - Επιστρέφει την τιμή ενός attribute ενός tag ή το κείμενο που σχετίζεται με έναν κόμβο του XML κειμένου
- <xsl:for-each select="tag-name"> … <xsl:for-each>
  - Επαναλαμβάνει ότι ακολουθεί για κάθε tag που ταιριάζει με το συγκεκριμένο tag name.
- <xsl:sort select="tag-name" />
  - Ορίζει με βάση ποιο tag να γίνει η ταξινόμηση

Τα στοιχεία ενός XSL αρχείου (2/2)

- <xsl:if test="price > 10">...</xsl:if>
  - Εάν η συνθήκη παίρνει την τιμή "true" τότε εμφανίζεται το αποτέλεσμα στον browser
- <xsl:choose>

```
<xsl:when test="price &gt; 10">...</xsl:when>
<xsl:otherwise>...</xsl:otherwise>
</xsl:choose>
```

- <xsl:apply-templates match="name" />
  - Εφαρμόζει όλα τα δυνατά templates στα tags που ταιριάζουν στην περιγραφή
- <xsl:element name = "Name">. . .</xsl:element>
  - Ορίζει ένα element δυναμικά
- <xsl:attribute name = "Name">Value</xsl:attribute>
  - Ορίζει ένα attribute για ένα element

<xsl:template>

- Ένα XSL αποτελείται από ένα σύνολο κανόνων, τα templates

  <xsl:template match="tag-name">…</xsl:template>

- Κάθε στοιχείο <xsl:template> περιέχει κανόνες που θα εφαρμοστούν όταν βρεθεί ένα συγκεκριμένο XML element
  - Το πεδίο "match" συνδέει το template με ένα XML element
  - Ορισμός ενός template για ένα ολόκληρο XML αρχείο match="/"

XSL αρχείο

XML αρχείο

<xsl:value-of>

<xsl:value-of select=”tag-name”>… </xsl:value-of>

- Επιστρέφει την τιμή ενός attribute ενός element ή ενός κόμβου του XML αρχείου
  - Το πεδίο “select” είναι απαραίτητο και περιέχει μία XPath expression
    - Χρησιμοποιεί το forward slash (/) για να επιλέγει υπο-καταλόγους

```
<title>ISBN <xsl:value-of select = "@isbn"/> -
  <xsl:value-of select = "title"/></title>
```

XSL αρχείο

XML αρχείο

<xsl:for-each> (1/2)

<xsl:for-each select=”tag-name”> … <xsl:for-each>

- Επιτρέπει τον ορισμό looping στο XSLT
- Ότι περιέχεται μεταξύ του start και end tag επαναλαμβάνεται για όσα XML elements ταιριάζουν με το tag name που έχει επιλεχθεί
  - Το πεδίο “select” είναι απαραίτητο και περιέχει μία XPath expression

```
<xsl:for-each select="catalog/cd">
    <tr><td><xsl:value-of select="title" /></td>
     <td><xsl:value-of select="artist" /></td></tr>
</xsl:for-each>
```

XSL αρχείο, XML αρχείο

<xsl:for-each> (2/2)

- Στο πεδίο “select” μπορούμε να προσθέσουμε ένα κριτήριο για να φιλτράρουμε τα δεδομένα από ένα XML αρχείο

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

- Τελεστές φιλτραρίσματος
  - = (ίσο)
  - != (διάφορο)
  - < (< μικρότερο)
  - > (> μεγαλύτερο)

XSL αρχείο

XML αρχείο

<xsl:sort>

- Ταξινόμηση αποτελέσματος

```
<xsl:sort select="tag-name" />
```

  - Το element "sort" πρέπει να μπει μέσα στο "for-each" element στο XSL αρχείο

```
<xsl:for-each select="catalog/cd">
  <xsl:sort select="artist" />
    <tr><td><xsl:value-of select="title" /></td>
      <td><xsl:value-of select="artist" /></td></tr>
</xsl:for-each>
```

- Το πεδίο "select" δηλώνει με βάση ποιο XML element θα γίνει η ταξινόμηση

XSL αρχείο XML αρχείο

<xsl:if>

<xsl:if test="price > 10">...</xsl:if>

- Το περιεχόμενο του “if” element εμφανίζεται εάν η συνθήκη είναι αληθής (true)
- Το “if” element μπορεί να τοποθετηθεί οπουδήποτε μέσα σε ένα XSL αρχείο

```
<xsl:if test="price &gt; 10">
  some output ...
</xsl:if>
```

- Το πεδίο “test” είναι απαραίτητο και περιέχει τη συνθήκη που θα ελεγχθεί

XSL αρχείο

XML αρχείο

<xsl:choose>

- Για πολλαπλούς ελέγχους συνθήκης

```
<xsl:choose>
  <xsl:when test="price &gt; 10">...</xsl:when>
        <xsl:otherwise>...</xsl:otherwise>
</xsl:choose>
```

- Το "choose" element μπορεί να τοποθετηθεί οπουδήποτε μέσα σε ένα XSL αρχείο

XSL αρχείο XML αρχείο

- Το element "when" μπορεί να εμφανίζεται πάνω από μία φορές μέσα σε ένα "choose"

XSL αρχείο XML αρχείο

<xsl:apply-templates>

<xsl:apply-templates>

- Εφαρμόζει ένα template στο τρέχων element ή στο παιδί του τρέχων element
- Είναι δυνατό να επιλεγεί σε ποιο element θα εφαρμοστούν τα templates

<xsl:apply-templates select="title"/>

- Το πεδίο "select" μπορεί να χρησιμοποιηθεί για να καθοριστεί η σειρά επεξεργασίας των παιδιών ενός element

```
<xsl:template match="cd"><p>
                <xsl:apply-templates select="title"/>
                <xsl:apply-templates select="artist"/></p>
</xsl:template>
```

XSL αρχείο

XML αρχείο

XSLT – on the Client (1/3)

- Ένα XSLT αρχείο μπορεί να χρησιμοποιηθεί για να μετατρέψει ένα XML αρχείο σε XHTML στον browser σας
  - Σύνδεση του XML αρχείου με ένα XSLT
  - Χρησιμοποιώντας JavaScript, VBScript …
- Με τη JavaScript μπορούμε
  - να ελέγξουμε ποιο browser χρησιμοποιεί ο χρήστης
  - να χρησιμοποιήσουμε διαφορετικά style sheets με βάση το browser και τις απαιτήσεις του χρήστη

XSLT – on the Client (2/3)

```
<html>
    <body>
    <script type="text/javascript">
      // Load XML
      var xml = new ActiveXObject("Microsoft.XMLDOM")
      xml.async = false
      xml.load("cdcatalog.xml")
      // Load XSL
      var xsl = new ActiveXObject("Microsoft.XMLDOM")
      xsl.async = false
      xsl.load("cdcatalog.xsl")
      // Transform
      document.write(xml.transformNode(xsl))</script>
    </body>
</html>
```

html αρχείο

XSLT – on the Client 3/3

- Load XML
  - Δημιουργία αντίγραφου του Microsoft XML parser
    var xml = new ActiveXObject("Microsoft.XMLDOM")
  - Απενεργοποιεί τη ασύγχρονη φόρτωση ώστε ο parser να μην αρχίσει την εκτέλεσή του πριν φορτωθεί ολόκληρο το αρχείο
    xml.async = false
  - Φόρτωση του XML αρχείου
    xml.load("cdcatalog.xml")

XML αρχείο

html αρχείο

XSLT – on the Server

- Υπάρχουν browsers που δεν υποστηρίζουν το XSLT
  - Η μετατροπή γίνεται στο server και ο browser παίρνει καθαρό XHTML
- Σε μία asp σελίδα (ή php)

```
<%
'Load XML
set xml = Server.CreateObject("Microsoft.XMLDOM")
xml.async = false
xml.load(Server.MapPath("cdcatalog.xml"))
'Load XSL
set xsl = Server.CreateObject("Microsoft.XMLDOM")
xsl.async = false
xsl.load(Server.MapPath("cdcatalog.xsl"))
'Transform file
Response.Write(xml.transformNode(xsl))
%>
```

XML – data islands (1/2)

- Είναι δυνατή η ενσωμάτωση XML δεδομένων σε HTML σελίδες με χρήση των *Data Islands*
  - Απ' ευθείας ενσωμάτωση στο HTML αρχείο

```
<xml id="note">
  <note>
        <to>Tove</to>
        <from>Jani</from>
        <heading>Reminder</heading>
  </note>
</xml>
```

  - Ενσωμάτωση XML αρχείου

```
<xml id="note" src="note.xml"></xml>
```

- Data island: τα έγγραφα της XML που είναι ενσωματωμένα μέσα σ' ένα αρχείο HTML

XML – data islands (2/2)

- Τα Data Islands μπορούν να συνδεθούν με HTML elements
  - Η σύνδεση γίνεται με χρήση του απαραίτητου πεδίου "id"
  - Προσοχή: το tag <xml> είναι της HTML !

```
<html>
  <body>
        <xml id="cdcat" src="cd_catalog.xml"></xml>
        <table border="1" datasrc="#cdcat">
        <tr>
              <td><span datafld="ARTIST"></span></td>
        <td><span datafld="TITLE"></span></td>
        </tr>
        </table>
  </body>
</html>
```

html αρχείο

DOM – Document Object Model

- Ένα standard **programming interface** για να προσπελάσουμε και να επεξεργαστούμε τη δομή και τα δεδομένα που περιέχονται σε ένα αρχείο XML
  - Βασίζεται στην δενδρική αναπαράσταση ενός XML αρχείου
- Δημιουργία ενός XML αρχείου
- Προσπέλαση της δενδρικής δομής ενός XML αρχείου
- Εισαγωγή / επεξεργασία / διαγραφή XML elements

DOM – Node Interface

- Το DOM αναπαριστά τη δενδρική μορφή ενός XML αρχείου
  - Η ρίζα του δένδρου είναι το **documentElement**
    - Το element αυτό αποτελείται από ένα ή περισσότερα **childNodes**
  - Το **Node Interface Model** χρησιμοποιείται για να προσπελαύνονται μεμονωμένα elements του δένδρου
- Ο Microsoft XML parser υποστηρίζει όλες τις απαραίτητες συναρτήσεις

Φόρτωση XML αρχείου στον parser

```
<html>
    <body>
    <script type="text/javascript">
      // Load XML
      var xml = new ActiveXObject("Microsoft.XMLDOM")
      xml.async = false
      xml.load("cdcatalog.xml")
    //          .... process
    </script>
    </body>
</html>
```

html αρχείο

Φόρτωση XML κειμένου στον parser

```
<script type="text/javascript">
  var text="<note>"
  text=text+"<to>Tove</to><from>Jani</from>"
  text=text+"<heading>Reminder</heading>"
  text=text+"<body>Don't forget me this weekend!</body>"
  text=text+"</note>"

  var xmlDoc = new ActiveXObject("Microsoft.XMLDOM")
  xmlDoc.async="false"
  xmlDoc.loadXML(text)
  // ..... processing the document goes here
</script>
```

DOM – node object

- Εύρεση του root element

  xmlDocument.documentElement;

- Node object
  - attributes
  - childNodes
  - firstChild / lastChild
  - nextSibling / previousSibling
  - nodeName / nodeType
  - nodeValue
  - parentNode

XML αρχείο

DOM παράδειγμα

SAX – Simple API for XML

- Επιτρέπει στους προγραμματιστές να επωφεληθούν από το event-driven XML parsing
- Αντίθετα απ' ότι συμβαίνει με το DOM που απαιτεί να φορτωθεί στη μνήμη ολόκληρο το XML αρχείο, ο XML parser κάθε φορά που διαβάζει ένα στοιχείο XML καλεί τον αντίστοιχο handler που έχει ορίσει ο προγραμματιστής
- Συνήθως χρειάζεται Java/C++

SAX Events (1/3)

- setDocumentLocator
  - Προκαλείται κατά την έναρξη του parsing
- startDocument
  - Προκαλείται όταν ο parser βρίσκεται στην αρχή του XML αρχείου
- endDocument
  - Προκαλείται όταν ο parser βρίσκεται στο τέλος του XML αρχείου

SAX Events (2/3)

- startElement
  - Προκαλείται όταν ο parser βρίσκει το αρχικό tag ενός element
- endElement
  - Προκαλείται όταν ο parser βρίσκει το tag όπου κλείνει ένα element
- characters
  - Προκαλείται όταν ο parser βρίσκει χαρακτήρες κειμένου

SAX Events (3/3)

- ignorableWhitespace
  - Προκαλείται όταν ο parser βρίσκει κενούς χαρακτήρες οι οποίοι μπορούν να αγνοηθούν ασφαλώς
- processingInstruction
  - Προκαλείται όταν ο parser βρίσκει μία εντολή προς εκτέλεση

SAX vs DOM

- DOM
  - Παρέχει ένα επεξεργασμένο δέντρο στη μνήμη
  - Όλο το αρχείο πρέπει να γίνει **parsed** πριν χρησιμοποιηθεί
  - Απαιτείται μνήμη ώστε να αποθηκευτεί όλο το δέντρο
  - Το δέντρο μπορεί να επεξεργαστεί προς οποιαδήποτε κατεύθυνση
- SAX
  - Το αρχείο δεν είναι αποθηκευμένο στη μνήμη
  - Το αρχείο γίνεται **parsed 'on the fly'**
  - Δεν χρειάζεται μνήμη για να γίνει **parsed** το αρχείο
  - Το αρχείο μπορεί να επεξεργαστεί μόνο προς την «μπροστά» κατεύθυνση