

Προγραμματισμός και Συστήματα στον Παγκόσμιο Ιστό

XML

Δρ. Δημήτριος Κουτσομητρόπουλος

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 είδους 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](#)

5

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)

6

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/>`

7

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 -->`

8

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

Το περιεχόμενο των elements ποικίλλει

- Άλλα elements (child elements)
- Μεικτό περιεχόμενο – κείμενο & elements
- Μόνο κείμενο
- Άδειο element

Δεν υπάρχει περιορισμός στο βάθος της εμφώλευσης 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:

- < <
- > >
- & &
- ‘ '
- " "

Σφάλμα παράγουν μόνο οι χαρακτήρες “<” και “&”

13

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](#)

14

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:*prefix*="namespace"

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

```
<directory xmlns:text="http://www.standard.org/text/">  
  <text:file filename="book.xml">  
    <text:description>A book list</text:description>  
  </text:file>  
</directory>
```

[View page](#)

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="http://www.standard.org/text/" >  
  <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 εμφωλευμένα
- 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

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](#)

21

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>
```

22

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>
```

23

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

24

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|message)> - selection

<!ELEMENT note (#PCDATA|body|message)*>

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>

- 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 pl System "usr/bin/perl">`
- xml: μία προκαθορισμένη xml τιμή
`<!ATTLIST note xml:lang CDATA 'en'>`
`<!ATTLIST note xml:space (default|preserve) 'preserve'>`

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">`

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 dcterms "http://purl.org/dc/terms/">`

Εσωτερική δήλωση entity

- `<!ENTITY entity-name "entity-value">`
`<!ENTITY dcterms "http://purl.org/dc/terms/">`

Εξωτερική δήλωση entity

- `<!ENTITY entity-name SYSTEM "URI/URL">`
`<!ENTITY writer SYSTEM "entities.xml">`
`<!ENTITY writer SYSTEM "entities.dtd">`

XML Schema

31

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>
```

32

XSD vs DTD

Πλεονεκτήματα των XSD αρχείων

- Εύκολα επεκτάσιμα
- *Είναι 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

33

Ένα απλό XML Schema

```
<?xml version="1.0"?>
<xs:schema xmlns:xs=http://www.w3.org/2001/XMLSchema
  targetNamespace=http://www.notes.com
  xmlns=http://www.notes.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>
```

34

Δήλωση XML Schema μέσα σε ένα XML αρχείο

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

[note.xml](#) [note.xsd](#)

35

<Schema> element

<schema> - root element

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

36

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"/>`

37

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"/>`

38

XSD – restrictions & facets

```
<xs:element name="age">
  <xs:simpleType name="new_stype">
    <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 | collapse"/>
<xs:length value="8"/>, minlength, maxlength, pattern
```

39

XSD – facets

```
<simpleType name='celsiusBodyTemp'>
  <restriction base='decimal'>
    <totalDigits value='4'/>
    <fractionDigits value='1'/>
    <minInclusive value='36.4'/>
    <maxInclusive value='40.5'/>
  </restriction>
</simpleType>
```

40

XSD – complex elements

```
<xs:element name="employee">
  <xs:complexType name="new_ctype">
    <xs:attribute name="emplID" type="xs:integer" />
    <xs:sequence>
      <xs:element name="firstname" type="xs:string" minOccurs="1" maxOccurs="2" />
      <xs:element name="lastname" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

<xs:all> : δεν έχει σημασία η σειρά (+ minOccurs="1")

<xs:choice> : ένα από

<xs:any>, **<xs:anyAttribute>** : επέκταση με elements/attributes που δεν ορίζονται στο τρέχον Schema

XML Parsing

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

43

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 **DOM Node Interface** χρησιμοποιείται για να προσπελάζονται μεμονωμένα elements του δένδρου

Parsing XML με JavaScript

- Φόρτωση από αρχείο: Χρήση **XMLHttpRequest**
 - Πραγματοποιεί ένα AJAX call για να φέρει το αρχείο
 - Το αποτέλεσμα επιστρέφεται στο property *responseXML*
- Φόρτωση από string: Χρήση **DOMParser**

44

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 parser = new DOMParser();
    xmlDocument = parser.parseFromString (text, "text/xml")
    // ..... processing the document goes here
</script>
```

XML αρχείο

DOM παράδειγμα

45

Φόρτωση XML *αρχείου* στον parser

```
<html>
<body>
<script type="text/javascript">

// Load XML
xhttp = new XMLHttpRequest();
xhttp.async=false;
xhttp.open("GET","article.xml");
xhttp.send();
xmlDocument = xhttp.responseXML;
```

```
//... process
</script>
</body>
</html>
```

XML αρχείο



--allow-file-access-from-files

DOM παράδειγμα

46

DOM – node object

Εύρεση του root element

`xmlDocument.documentElement;`

Node object

- attributes
- childNodes **vs** children
- firstChild / lastChild **vs** firstElementChild / lastElementChild
- nextSibling / previousSibling ...
- nodeName / nodeType
- nodeValue
- parentNode

[XML αρχείο](#)

[DOM παράδειγμα](#)

47

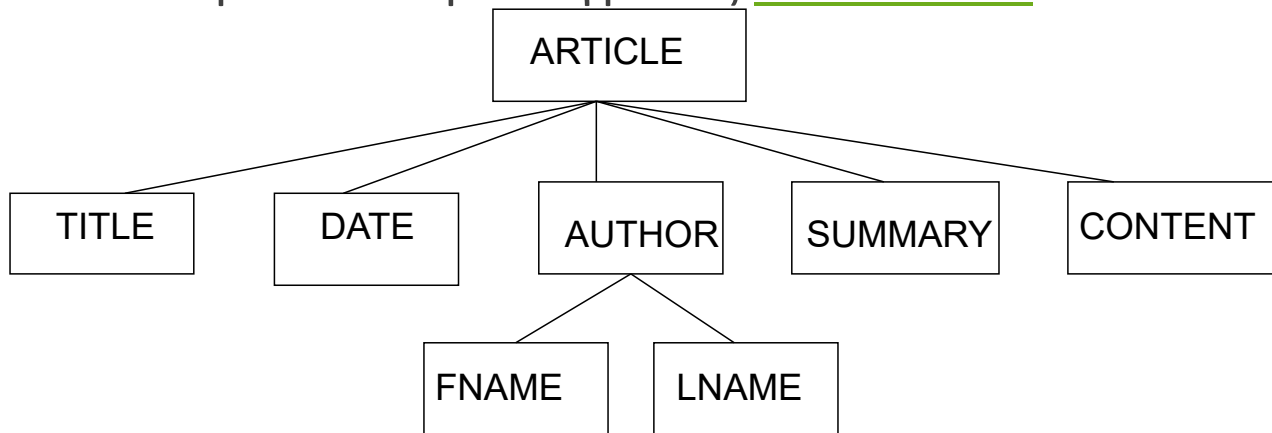
XML αρχείο για DOM

```
<?xml version = "1.0"?>
<!-- Article formatted with XML -->
<article>
  <title>Simple XML</title>
  <date>December 6, 2000</date>
  <author>
    <fname>Tem</fname>
    <lname>Nieto</lname>
  </author>
  <summary>XML is pretty easy.</summary>
  <content>Once you have mastered HTML, XML is easily
    learned. You must remember that XML is not for
    displaying information but for managing information.
  </content>
</article>
```

48

DOM Example

Το δέντρο του παραδείγματος: DOM Parser



49

XML Parsing - PHP

SimpleXML is a tree-based parser

- `simplexml_load_string();`
- `simplexml_load_file();`

SimpleXMLElement Object (
[to] => Tove [from] => Jani
[heading] => Reminder [body]
=> Don't forget me this
weekend!)

```
<?php
$xmlData =
"<?xml version='1.0'
encoding='UTF-8'?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this
weekend!</body>
</note>";
```

```
$xml=simplexml_load_string($xmlData) or die("Error: Cannot
create object");
print_r($xml);
?>
```

50

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++

51

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

52

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

53

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

54

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

XSL

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

57

XSL - EXtensible Stylesheet Language

Δεν είναι μία απλή style sheet γλώσσα

Ένα σύνολο γλωσσών

- **XSLT**- μετατροπή XML αρχείων
- **XPath** – ορισμός τμημάτων ενός XML αρχείου
 - Πλοήγηση (και αναφορά) στο XML έγγραφο
- **XSL-FO** – μορφοποίηση XML αρχείων & δεδομένων
 - Στην οθόνη, σε χαρτί ή αλλού
 - Σελιδοποίηση και τυπογραφία
 - Π.χ. Adobe InDesign

58

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

59

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">
```

60

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>
```

61

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

62

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

63

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 > 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

64

<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

65

<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

66

<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](#)

67

<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](#)

68

<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 > 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 > 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 τρέχον element ή στο παιδί του τρέχοντος 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](#)

Παραδείγματα XSL αρχείων

Απλό παράδειγμα
Αρχικό XML αρχείο

XSL αρχείο
Τελικό XML αρχείο

XSL iteration & sorting
Αρχικό XML αρχείο

XSL αρχείο
Τελικό XML αρχείο

XSL conditions
Αρχικό XML αρχείο

XSL αρχείο
Τελικό XML αρχείο

73

XSLT – on the Client

Ένα XSLT αρχείο μπορεί να χρησιμοποιηθεί για να μετατρέψει ένα XML αρχείο σε HTML στον browser

- Σύνδεση του XML αρχείου με ένα XSLT
- Χρησιμοποιώντας JavaScript, VBScript ...

Με τη JavaScript μπορούμε

- να ελέγξουμε ποιον browser χρησιμοποιεί ο χρήστης
- να χρησιμοποιήσουμε διαφορετικά style sheets με βάση τον browser και τις απαιτήσεις του χρήστη

74

XSLT – on the Client

```
<script type="text/javascript">
  // Load XML
  xhttp = new XMLHttpRequest();
  xhttp.async=false;
  xhttp.open("GET","cdcatalog.xml", false);
  xhttp.send();
  xml = xhttp.responseXML;

  // Load the XSL
  xhttp = new XMLHttpRequest();
  xhttp.async=false;
  xhttp.open("GET","cdcatalog.xsl", false);
  xhttp.send();
  xsl = xhttp.responseXML;

  // Transform
  xslt = new XSLTProcessor();
  xslt.importStylesheet(xsl);
  result = xslt.transformToFragment(xml, document);
  document.querySelector("body").appendChild(result);
</script>
```

[html αρχείο](#)

75

Παράδειγμα εφαρμογής - KML

[HTTPS://DEVELOPERS.GOOGLE.COM/KML/DOCUMENTATION/KML_TUT](https://developers.google.com/kml/documentation/kml_tut)

76

KML

KML: Keyhole Markup Language

- Keyhole: είναι το όνομα μιας εταιρείας που απέκτησε η Google το 2004 και η οποία ανέπτυξε την πρώτη έκδοση του Google Earth που ονομαζόταν Keyhole Earth Viewer

Η KML είναι μια μορφή αρχείου που χρησιμοποιείται για την προβολή γεωγραφικών δεδομένων σε ένα πρόγραμμα περιήγησης της Γης, όπως το Google Earth.

KML είναι μία γλώσσα τύπου XML

- Χρησιμοποιεί δομή των tags με εμφωλευμένα elements και attributes
- Όλα τα tags είναι case sensitive και πρέπει να εμφανίζονται ακριβώς όπως επιτρέπει η δομή της KML. Η δομή αυτή υπαγορεύει ποια tags είναι προαιρετικά. Εντός ενός δεδομένου element, τα tags πρέπει να εμφανίζονται με βάση αυτή τη δομή.

77

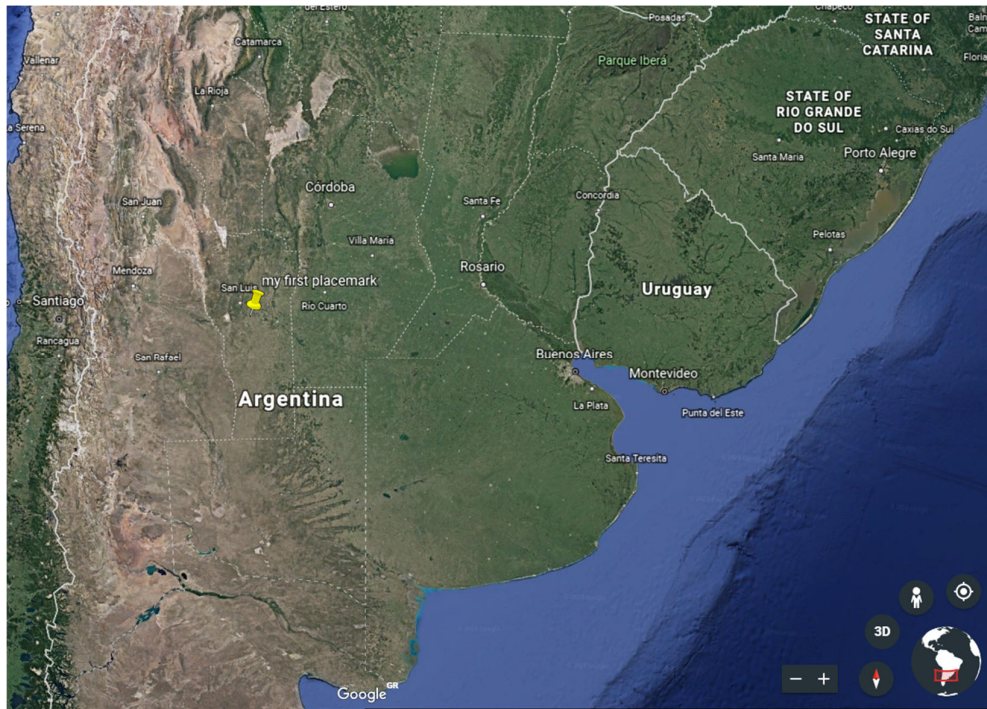
Απλό παράδειγμα KML (1/2)

```
<kml>
  <Document>
    <name>my first point</name>
    <Placemark>
      <name>my first placemark</name>
      <Point>
        <coordinates>-66.09993326298255,-
                      33.56312777383666,0
        </coordinates>
      </Point>
    </Placemark>
  </Document>
</kml>
```

Σώστε αυτό το αρχείο με κατάληξη .kml και ανοίξτε το μέσω του Google Earth. Θα δείτε το σημείο με αυτές τις συντεταγμένες

78

Απλό παράδειγμα KML (2/2)



79

Βασικά στοιχεία ενός KML εγγράφου

Το απλούστερο είδος εγγράφων KML είναι αυτά που μπορούν να δημιουργηθούν απευθείας στο Google Earth, δηλαδή δεν χρειάζεται να επεξεργάζεστε ή να δημιουργείτε KML σε ένα πρόγραμμα επεξεργασίας κειμένου π.χ. notepad.

Τα σημεία αναφοράς (Placemarks), οι επικαλύψεις εδάφους (Ground overlays), τα μονοπάτια (Paths) και τα πολύγωνα (Polygons) μπορούν να δημιουργηθούν απευθείας στο Google Earth.

80

Απλό παράδειγμα KML (1/2)

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Placemark>
    <name>Simple placemark</name>
    <description>Attached to the ground. Intelligently places itself
      at the height of the underlying terrain.</description>
    <Point>
      <coordinates>-122.0822035425683,37.42228990140251,0</coordinates>
    </Point>
  </Placemark>
</kml>
```

81

Απλό παράδειγμα KML (2/2)

<?xml version="1.0" encoding="UTF-8"?>

- Επικεφαλίδα XML. Αυτή βρίσκεται στην πρώτη γραμμή κάθε KML αρχείου. Δεν μπορούν να εμφανιστούν κενά ή άλλοι χαρακτήρες πριν από αυτή τη γραμμή

<kml xmlns="http://www.opengis.net/kml/2.2">

- Ορισμός namespace. Αυτή είναι η 2^η γραμμή σε κάθε KML 2.2 αρχείο.

<Placemark>

- Σηματοδοτεί μια θέση στην επιφάνεια της Γης, χρησιμοποιώντας μια κίτρινη πινελιά ως εικόνα.
- Περιλαμβάνει τα εξής elements:
- Name: χρησιμοποιείται ως η ετικέτα του placemark
- Description: εμφανίζεται ως ένα έξτρα πεδίο με πληροφορίες
- Point: δίνει τις ακριβείς συντεταγμένες (longitude, latitude, και προαιρετικά altitude)

82

Εισάγοντας HTML (1/2)

Αν θέλετε να γράψετε HTML μέσα σε ένα `<description>` tag, μπορείτε να βάλετε HTML σε ένα tag.

Μπορείτε να βάλετε HTML μέσα σε CDATA ή χωρίς αλλά με προσοχή στους χαρακτήρες "<" να γράφονται ως `<`; κλπ

83

Εισάγοντας HTML (2/2)

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <Placemark>
      <name>CDATA example</name>
      <description>
        <![CDATA[
          <h1>CDATA Tags are useful!</h1>
          <p><font color="red">Text is <i>more readable</i> and
            <b>easier to write</b> when you can avoid using entity
            references.</font></p>
        ]]>
      </description>
      <Point>
        <coordinates>102.595626,14.996729</coordinates>
      </Point>
    </Placemark>
  </Document>
</kml>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Document>
    <Placemark>
      <name>Entity references example</name>
      <description>
        &lt;h1&gt;Entity references are hard to type!&lt;/h1&g
        &lt;p&gt;&lt;font color="green"&gt;Text is
          &lt;i&gt;more readable&lt;/i&gt;
          and &lt;b&gt;easier to write&lt;/b&gt;
          when you can avoid using entity references.&lt;/font&gt;&lt;
        </description>
        <Point>
          <coordinates>102.594411,14.998518</coordinates>
        </Point>
      </Placemark>
    </Document>
  </kml>
```

84

Placemarks (σημεία αναφοράς)

Τα Placemarks είναι ένα από τα πιο συχνά χρησιμοποιούμενα χαρακτηριστικά του Google Earth.

Σηματοδοτεί μια θέση στην επιφάνεια της Γης, χρησιμοποιώντας μια κίτρινη πινελιά ως εικόνα.

Ο απλούστερος δείκτης τοποθεσίας περιλαμβάνει μόνο ένα στοιχείο <Point>, το οποίο καθορίζει τη θέση του σημείου αναφοράς.

Μπορείτε να ορίσετε ένα όνομα και ένα προσαρμοσμένο εικονίδιο για το Placemark και μπορείτε επίσης να προσθέσετε άλλα στοιχεία γεωμετρίας σε αυτό.

85

```
SAMPLES SUPPORT
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Placemark>
    <name>The Pentagon</name>
    <Polygon>
      <extrude>1</extrude>
      <altitudeMode>relativeToGround</altitudeMode>
      <outerBoundaryIs>
        <LinearRing>
          <coordinates>
            -77.05788457660967,38.87253259892824,100
            -77.05465973756702,38.87291016281703,100
            -77.05315536854791,38.87053267794386,100
            -77.05552622493516,38.868757801256,100
            -77.05844056290393,38.86996206506943,100
            -77.05788457660967,38.87253259892824,100
          </coordinates>
        </LinearRing>
      </outerBoundaryIs>
      <innerBoundaryIs>
        <LinearRing>
          <coordinates>
            -77.05668055019126,38.87154239798456,100
            -77.05542625960818,38.87167890344077,100
            -77.05485125901024,38.87076535397792,100
            -77.05577677433152,38.87008686581446,100
            -77.05691162017543,38.87054446963351,100
            -77.05668055019126,38.87154239798456,100
          </coordinates>
        </LinearRing>
      </innerBoundaryIs>
    </Polygon>
  </Placemark>
</kml>
```

Polygons (Πολύγωνα)

Μπορείτε να χρησιμοποιήσετε Polygons για να δημιουργήσετε απλά κτίρια και άλλα σχήματα.

Σύνολο συντεταγμένων (άκρες πολυγώνου) μέσα στο tag <coordinates>

Οι συντεταγμένες αποθηκεύονται σε ένα tag που ονομάζεται <LinearRing>

Το οποίο είναι εμφωλευμένο μέσα στο tag <outerBoundaryIs>