ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΧΕΙΡΙΣΗΣ ΠΟΛΙΤΙΣΜΙΚΗΣ ΠΛΗΡΟΦΟΡΙΑΣ

Τεχνολογίες XML

Γιώργος Καρυδάκης
Τμήμα Πολιτισμικής Τεχνολογίας και Επικοινωνίας

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αιγαίου**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Ενότητα 4:
Τεχνολογίες XML

Τεχνολογίες Διαχείρισης Πολιτισμικής Πληροφορίας

Γ. Καρυδάκης

DTD

XML: Document Type Definitions (DTDs)

- Επιτρέπει να ορίσουμε και να χρησιμοποιήσουμε στοιχεία, γνωρίσματα και οντότητες της επιλογής μας.
- Ένα έγγραφο XML είναι *καλά διαμορφωμένο* (*well-formed*) αν:
  - Το έγγραφο ξεκινά με ένα δηλωτικό XML.
  - Διαθέτει στοιχείο ρίζα που περιέχει όλα τα υπόλοιπα στοιχεία.
  - Όλα τα στοιχεία του είναι κατάλληλα εμφωλευμένα.
- Είναι χρήσιμο να τίθενται κοινά αποδεκτοί κανόνες που προδιαγράφουν συγκεκριμένο λεξιλόγιο από επιτρεπτά ονόματα στοιχείων και γνωρισμάτων, και θέτουν περιορισμούς ως προς την πολλαπλότητα εμφάνισης των στοιχείων, την μεταξύ τους σειρά κ.λ.π.
- Κάθε κοινότητα χρηστών μπορεί να προδιαγράψει τη δική της XML διάλεκτο με βάση τις ανάγκες των μελών της.

XML: Document Type Definitions (DTDs)

- Το περιεχόμενο ενός DTD παρέχει (μετα)πληροφορία στα *προγράμματα συντακτικής ανάλυσης* (parsers) των XML τεκμηρίων. Η πληροφορία αφορά τους περιορισμούς σύνταξης που πρέπει να πληρούν τα τεκμήρια ώστε να θεωρούνται *έγκυρα* ως προς το συγκεκριμένο DTD.
- *Έγκυρο* (valid) XML τεκμήριο: αν συνοδεύεται από ένα DTD και είναι δομημένο σύμφωνα με τους κανόνες που ορίζει το DTD.
- Ένα DTD λειτουργεί ως *γραμματική* (grammar) για μια κατηγορία XML τεκμηρίων, αφού παρέχει ένα λεξιλόγιο (αποδεκτά ονόματα στοιχείων και γνωρισμάτων) καθώς και σύνολο από κανόνες που διέπουν τη σειρά εμφάνισης

Παράδειγμα DTD

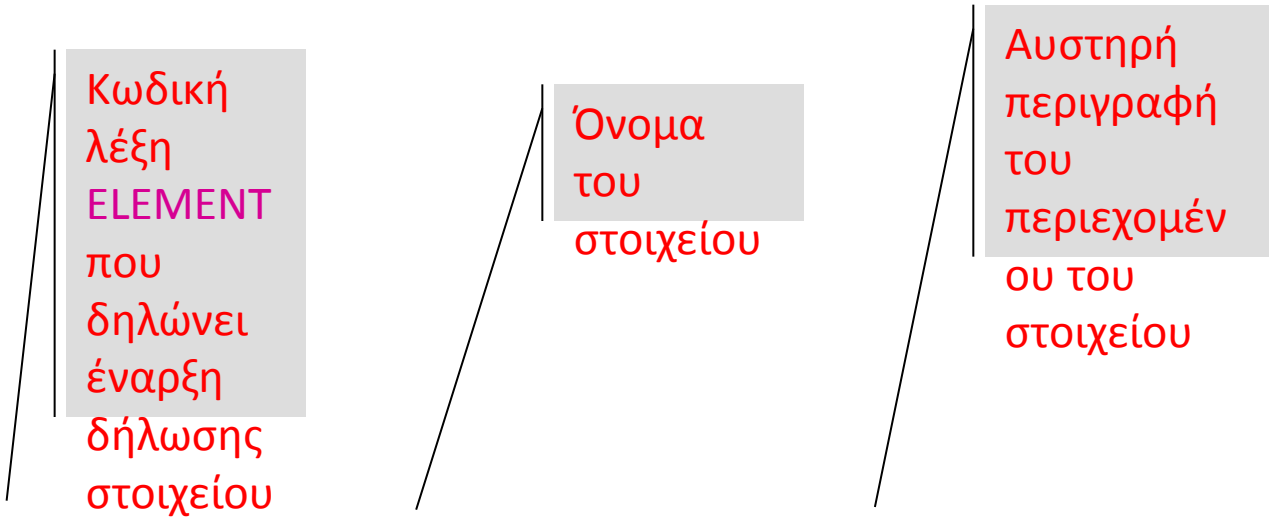
- Παράδειγμα XML τεκμηρίου που κωδικοποιεί στοιχεία φοιτητών του TAB:

```
<TAB>
  <φοιτητής>
            <όνομα> Νίκος </όνομα>
            <επώνυμο> Νικολάου </επώνυμο>
  </φοιτητής>
  <φοιτητής> … </φοιτητής>
  …
</TAB>
```

- Ένα DTD για το πιο πάνω τεκμήριο:

```
<!DOCTYPE TAB [                                  -Το κεντρικό στοιχείο είναι το TAB…
 <!ELEMENT TAB (φοιτητής*)>                      -Αποτελείται από στοιχεία φοιτητής…
 <!ELEMENT φοιτητής (όνομα, επώνυμο)> -Το στοιχείο φοιτητής περιλαμβάνει τα στοιχεία όνομα και επώνυμο…
 <!ELEMENT όνομα (#PCDATA)>                      -Το όνομα περιλαμβάνει χαρακτήρες
 <!ELEMENT επώνυμο (#PCDATA)>                    -Το επώνυμο περιλαμβάνει
```

DTD:Δηλώσεις Τύπου Στοιχείων



<!ELEMENT όνομα_ στοιχείου τύπος_στοιχείου>

Δηλώσεις Τύπου Στοιχείων

Δήλωση	*Σημασία*
R?	0 ή 1 στιγμιότυπο του R
R+	1 ή περισσότερα στιγμιότυπα του R
R*	0 ή περισσότερα στιγμιότυπα του R
R_1, R_2, … , Rn	1 στιγμιότυπο του R_1 ακολουθούμενο από 1 στιγμ. του

DTD:Δηλώσεις Τύπου Στοιχείων: Παράδειγμα

- Με την έκφραση:

  <!ELEMENT s (a, b?,c*)>

  δηλώνεται ότι: κάθε στοιχείο με ετικέτα **s** που εμφανίζεται σε ένα έγκυρο XML τεκμήριο, περιλαμβάνει ένα ακριβώς στοιχείο με ετικέτα a ακολουθούμενο προαιρετικά από ένα το πολύ στοιχείο με ετικέτα b, και στη

DTD:Δηλώσεις Τύπου Στοιχείων: Παράδειγμα

- Με την έκφραση:

<!ELEMENT e ((c? , d)* | (d , c)*)>

δηλώνεται ότι κάθε στοιχείο με ετικέτα e που εμφανίζεται σε ένα έγκυρο XML τεκμήριο, περιλαμβάνει:

–μια ακολουθία από (μηδέν ή περισσότερα) ζεύγη στοιχείων c, d από τα οποία το c είναι προαιρετικό

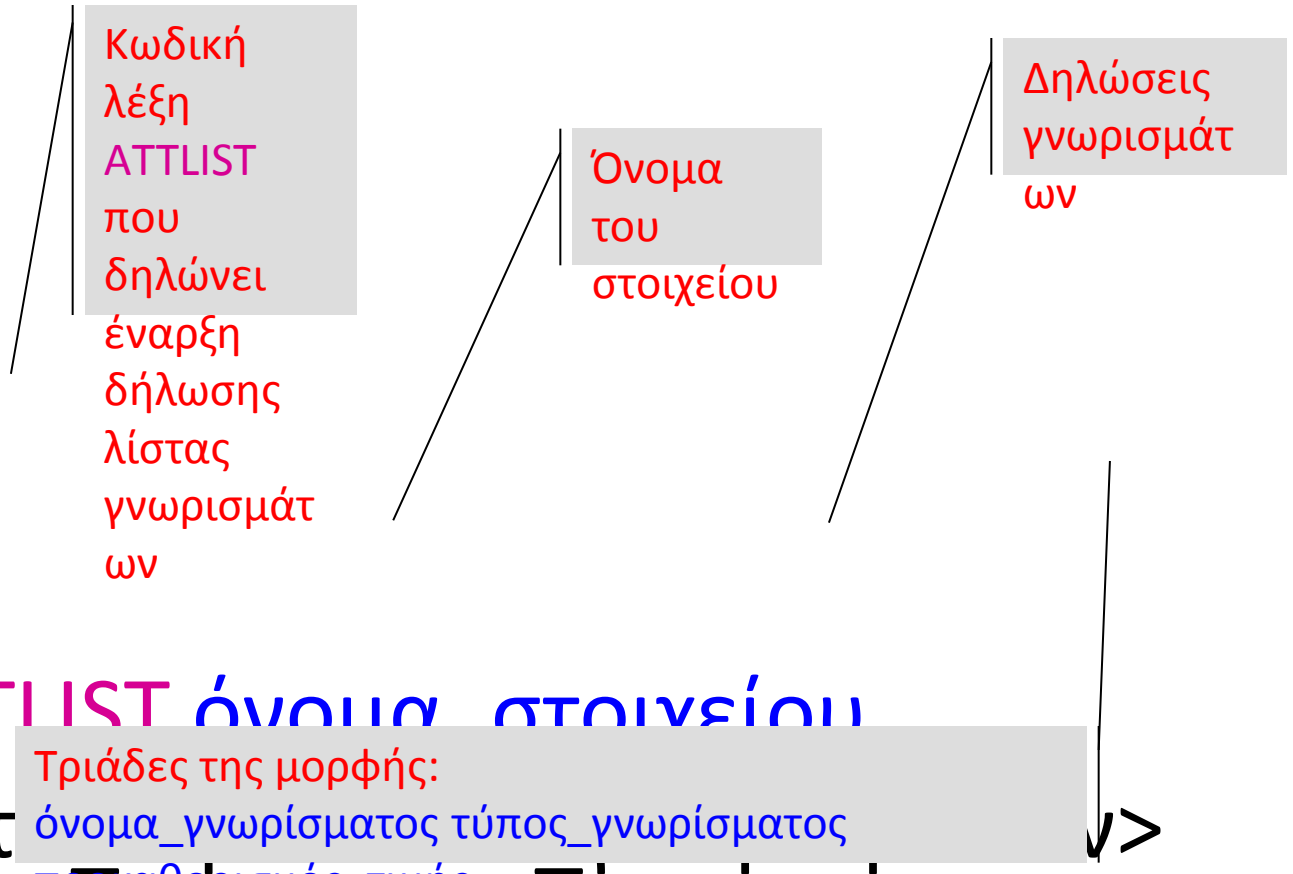
Δηλώσεις Τύπου Στοιχείων

- Για να δηλώσουμε ότι το περιεχόμενο ενός στοιχείου είναι ακολουθία χαρακτήρων χρησιμοποιούμε δηλώσεις της μορφής:

<!ELEMENT *όνομα_στοιχείου* (#PCDATA)>

- Η παράσταση *τύπος_στοιχείου* είναι επίσης δυνατό να πάρει μια από τις τιμές EMPTY και ANY που σημαίνουν το *κενό στοιχείο*, και το *στοιχείο με οποιοδήποτε περιεχόμενο* αντίστοιχα.

DTD:Δηλώσεις Λίστας Γνωρισμάτων



Δηλώσεις Λίστας Γνωρισμάτων: Παράδειγμα

```
<!ATTLIST φοιτητής ΑΜ CDATA #REQUIRED
                   ΑΔΤ CDATA #IMPLIED
>
```

- Το στοιχείο **φοιτητής** έχει δύο γνωρίσματα με ονόματα **ΑΜ** και **ΑΔΤ**.
- Και τα δύο γνωρίσματα είναι του τύπου **CDATA**.
- Η παρουσία του γνωρίσματος **ΑΜ** είναι υποχρεωτική σε κάθε εμφάνιση του στοιχείου φοιτητής (λόγω του

Δηλώσεις Λίστας Γνωρισμάτων

- Ανάμεσα στις πιθανές τιμές που μπορεί να πάρει η παράμετρος *τύπος_γνωρίσματος* περιλαμβάνονται και οι ακόλουθες: CDATA, ID, IDREF, IDREFS, ENTITY, ENTITIES, NMTOKEN, NMTOKENS.
  - Τα γνωρίσματα του τύπου CDATA μπορούν να πάρουν για τιμή οποιοδήποτε κείμενο (ακολουθία χαρακτήρων).
  - Τα γνωρίσματα του τύπου NMTOKEN είναι ειδική περίπτωση των γνωρισμάτων τύπου CDATA και μπορούν να πάρουν για τιμή μια απλή λέξη.
  - Τα γνωρίσματα του τύπου NMTOKENS μπορούν να πάρουν για τιμή πολλαπλές τιμές τύπου NMTOKEN που χωρίζονται μεταξύ τους με κενά.

Δηλώσεις Λίστας Γνωρισμάτων: Παράδειγμα

- Στη δήλωση λίστας γνωρισμάτων που ακολουθεί:

```
<!ATTLIST book color
(red|green|blue) "blue">
```

ορίζεται ότι:

–Το στοιχείο book έχει ένα γνώρισμα με όνομα color.

–Το γνώρισμα αυτό μπορεί να πάρει μια

Δηλώσεις Λίστας Γνωρισμάτων

- Τιμές της παράμετρου *προκαθορισμός_τιμής:*
  - Μπορεί να πάρει σαν τιμή μια από τις πιθανές τιμές του γνωρίσματος, με τη σημασία που αναφέραμε προηγούμενα.
  - Η τιμή #REQUIRED η οποία επιβάλει την υποχρεωτική εμφάνιση του γνωρίσματος στο αντίστοιχο στοιχείο.
  - Η τιμή #IMPLIED η οποία υποδηλώνει ότι δεν παρέχεται κάποια προκαθορισμένη τιμή (και δεν είναι

Δηλώσεις Λίστας Γνωρισμάτων: Παράδειγμα

- Με τη δήλωση:

  <!ATTLIST form method CDATA #FIXED "POST">

- ορίζεται ότι:

  –το στοιχείο form διαθέτει το γνώρισμα method το οποίο είναι τύπου CDATA και έχει πάντα τη τιμή POST.

Οι τύποι γνωρίσματος ID, IDREF, και IDREFS

- Ο τύπος ID δηλώνει ότι το γνώρισμα παίζει ρόλο ταυτότητας για το στοιχείο, παίρνει δηλαδή μοναδική τιμή που προσδιορίζει μονοσήμαντα το κάθε στοιχείο.
  - Δεν επιτρέπεται να υπάρχουν περισσότερα του ενός στοιχεία σε ένα έγκυρο XML τεκμήριο τα οποία να διαθέτουν γνώρισμα τύπου ID και να έχουν την ίδια τιμή στο γνώρισμα αυτό.
- Ο τύπος IDREF δηλώνει γνωρίσματα που παίρνουν σαν τιμή τη τιμή του γνωρίσματος τύπου ID κάποιου άλλου στοιχείου.
- Ένα γνώρισμα του τύπου ID συμπεριφέρεται όπως ένα “*κλειδί*” σε μια σχεσιακή βάση δεδομένων, ενώ ένα γνώρισμα του τύπου IDREF σαν ένα “*ξένο κλειδί*” σε μια σχεσιακή βάση δεδομένων.
- Ως IDREFS δηλώνονται γνωρίσματα τα οποία παίρνουν σαν τιμή μια λίστα τιμών του τύπου IDREF οι οποίες χωρίζονται μεταξύ τους με κενά

Οι τύποι γνωρίσματος ID, IDREF, και IDREFS: Παράδειγμα DTD

- Ένα DTD:

```
<!DOCTYPE οικογένεια [
<!ELEMENT οικογένεια (πρόσωπο)*>
<!ELEMENT πρόσωπο (όνομα, επώνυμο)>
<!ATTLIST πρόσωπο ΑΔΤ  ID #REQUIRED
                       μητέρα   IDREF   #IMPLIED
                       πατέρας IDREF   #IMPLIED
                       παιδιά    IDREFS #IMPLIED>
<!ELEMENT όνομα (#PCDATA)>
<!ELEMENT επώνυμο (#PCDATA)>
]>
```

Οι τύποι γνωρίσματος ID, IDREF, και IDREFS: Παράδειγμα

- Ένα XML τεκμήριο:

```
<οικογένεια>
  <πρόσωπο ΑΔΤ = "Κ123456" παιδιά = "Μ345678  Ν456789">
        <όνομα> Πέτρος </όνομα>
        <επώνυμο> Πέτρου </επώνυμο>
  </πρόσωπο>
  <πρόσωπο ΑΔΤ = "Λ234567" παιδιά = "Μ345678  Ν456789">
        <όνομα> Μαρία </όνομα>
        <επώνυμο> Πέτρου </επώνυμο>
  </πρόσωπο>
  <πρόσωπο ΑΔΤ = "Μ345678" πατέρας = "Κ123456" μητέρα= "Λ234567" >
        <όνομα> Γιώργος </όνομα>
        <επώνυμο> Πέτρου </επώνυμο>
  </πρόσωπο>
  <πρόσωπο ΑΔΤ = “Ν456789” πατέρας = "Κ123456" μητέρα= "Λ234567" >
        <όνομα> Άννα </όνομα>
        <επώνυμο> Πέτρου-Ιωάννου </επώνυμο>
  </πρόσωπο>
  ....    ....
</οικογένεια>
```

Δηλώσεις Οντοτήτων

- **Βασική ιδέα:** οι οντότητες αποτελούν συντομογραφίες σε ένα XML τεκμήριο και επομένως οι δηλώσεις των οντοτήτων στην πραγματικότητα αποτελούν ορισμό των συντομογραφιών αυτών.
- Διακρίνουμε τρία είδη οντοτήτων.
  - Οι *εσωτερικές οντότητες* (internal entities),
  - Οι *εξωτερικές οντότητες* (external entities),
  - Οι *οντότητες παραμέτρων* (parameter entities).
- Η *δήλωση μιας εσωτερικής οντότητας*

Δηλώσεις Οντοτήτων

- *Δηλώσεις εξωτερικών οντοτήτων*: μαζί με το όνομα της οντότητας δηλώνεται και ένα URI το οποίο θα πρέπει να ακολουθηθεί προκειμένου να βρεθεί το κείμενο το οποίο θα δοθεί σαν τιμή στην οντότητα.
- **Παράδειγμα:** Με τη δήλωση:

```
<!ENTITY TABinfo
        SYSTEM
"/ionio/information/info.xml">
```

Οι οντότητες παραμέτρων

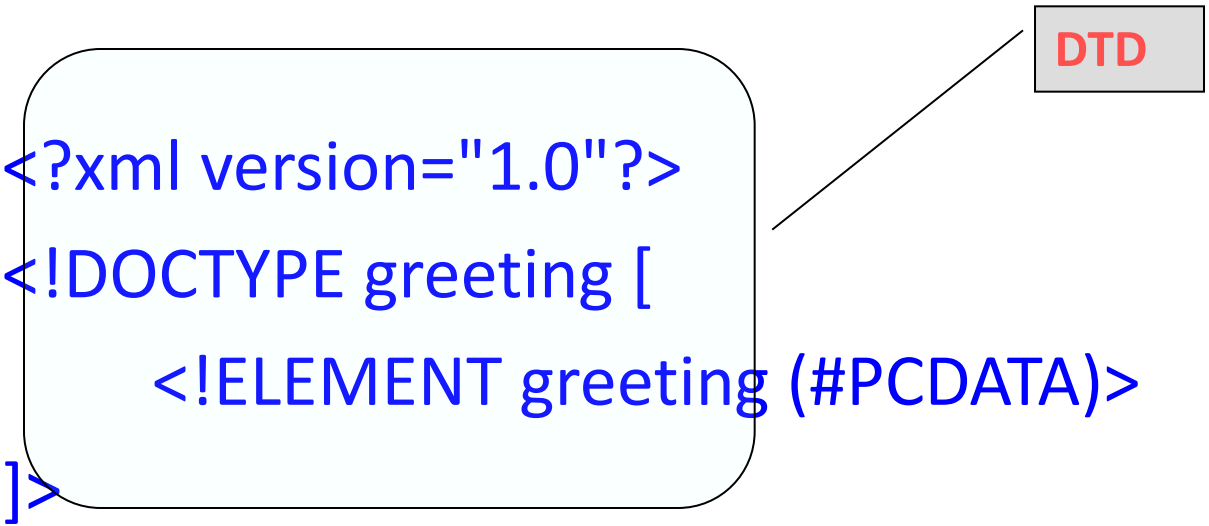
- Οι *οντότητες παραμέτρων* εμφανίζονται μόνο στα πλαίσια ενός DTD.
- Στη *δήλωση μιας οντότητας παραμέτρου* πριν από το όνομα της οντότητας πρέπει να τοποθετείται ο χαρακτήρας %.
- Το ίδιο σύμβολο % εμφανίζεται και στις αναφορές σε οντότητες παραμέτρου αντί για το σύμβολο &.

Σύνδεση XML με DTD

- Προκειμένου να εξεταστεί η εγκυρότητα ενός XML τεκμηρίου ως προς ένα DTD, θα πρέπει τα δύο αυτά να συσχετιστούν.
- Γενικά έχουμε δύο επιλογές.
  - *Πρώτη επιλογή*: να συμπεριλάβουμε το DTD στο ίδιο αρχείο με αυτό που βρίσκεται το XML τεκμήριο.
  - *Δεύτερη επιλογή*: να τοποθετήσουμε τις δηλώσεις του DTD σε ξεχωριστό αρχείο και στη συνέχεια να συσχετίσουμε κατάλληλα τα δύο αρχεία.

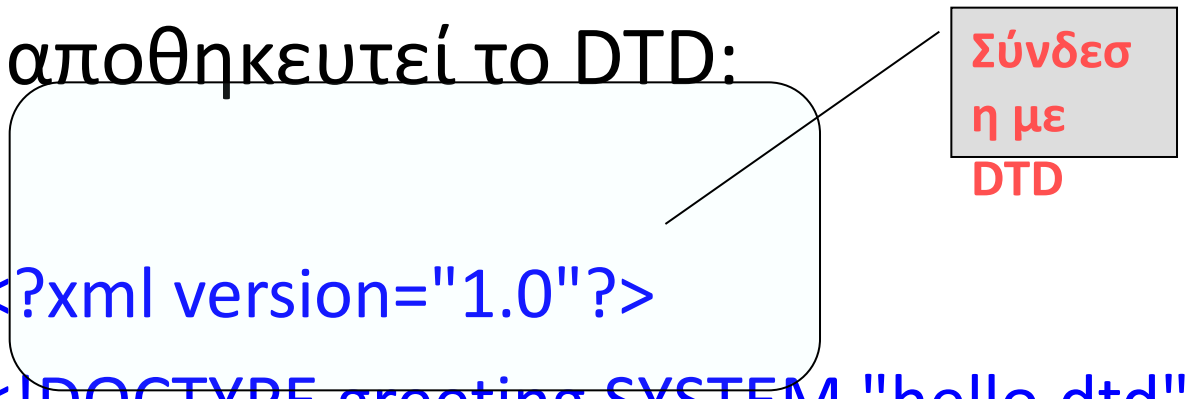
Σύνδεση XML με DTD:Παράδειγμα 1

- DTD ενσωματωμένο στο XML τεκμήριο:



Σύνδεση XML με DTD:Παράδειγμα 2

- Σύνδεση με εξωτερικό αρχείο στο οποίο έχει αποθηκευτεί το DTD:




```
<?xml version="1.0"?>
<!DOCTYPE greeting SYSTEM "hello.dtd">
<greeting>Hello, world!</greeting>
```

Υπό συνθήκη τμήματα

- Ένα *υπό συνθήκη τμήμα* (conditional section) είναι τμήμα ενός εξωτερικού DTD το οποίο συμπεριλαμβάνεται ή δεν συμπεριλαμβάνεται στη λογική δομή του DTD με βάση κάποια *λέξη κλειδί* (keyword) η οποία συνδέεται με αυτό.
- Αν η τιμή της λέξης κλειδί είναι INCLUDE, τότε το περιεχόμενο του υπό συνθήκη τμήματος συμπεριλαμβάνεται στο DTD.
- Αν η τιμή της λέξης κλειδί είναι IGNORE,

Υπό συνθήκη τμήματα:Παράδειγμα

- Παράδειγμα:

```
<!ENTITY % draft 'INCLUDE' >
<!ENTITY % final 'IGNORE' >

<! [%draft; [
<!ELEMENT article (comments*, title, body,
  supplements?)>
]]>
<! [%final; [
<!ELEMENT article (title, body, supplements?)>
]]>
```

- Η συμμετοχή ή όχι των υπό συνθήκη τμημάτων καθορίζεται από την τιμή των

Schema

Η γλώσσα *XML Schema*

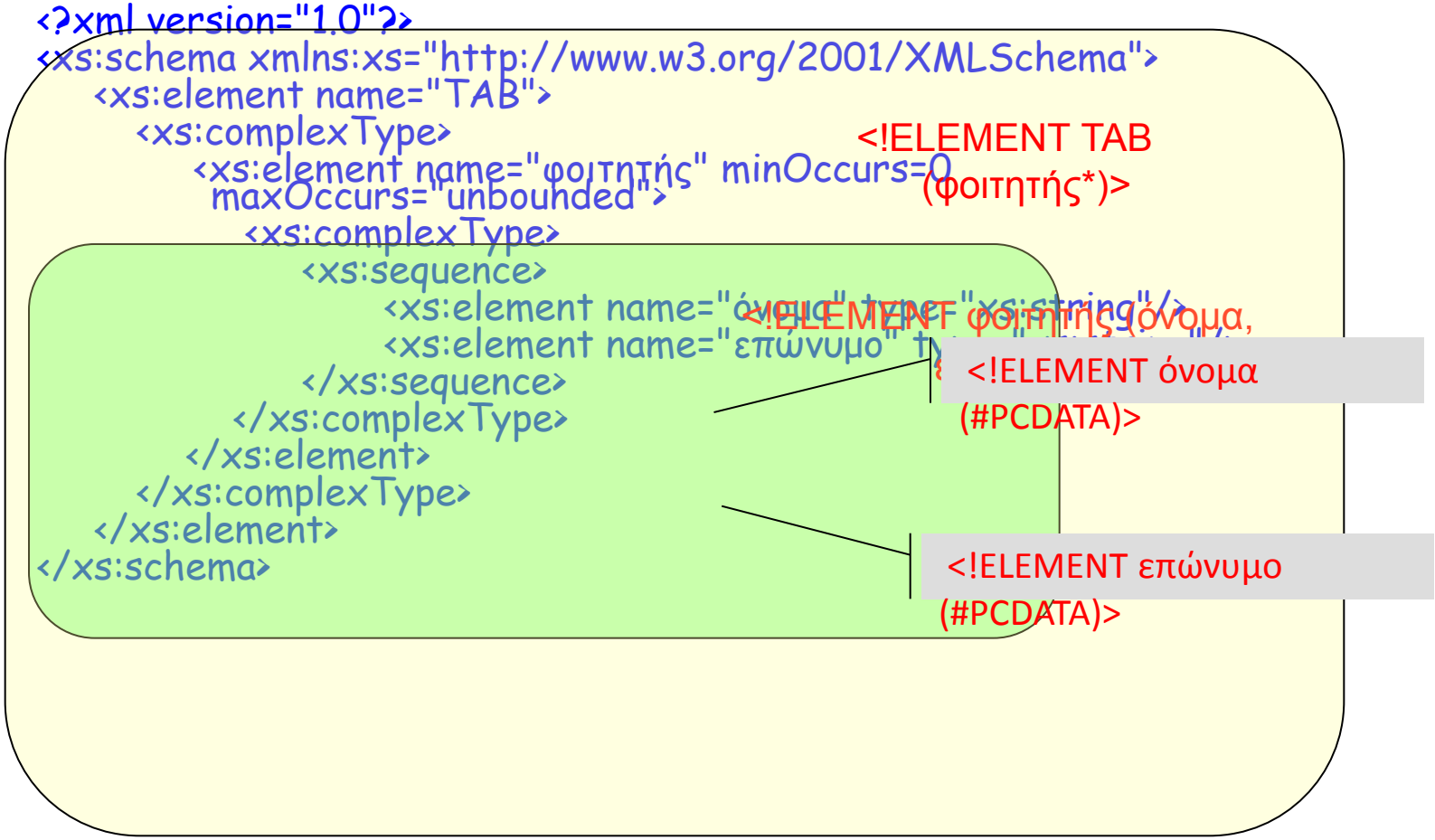
- Η γλώσσα *XML Schema* είναι μια γλώσσα XML κατάλληλη για την περιγραφή της δομής XML τεκμηρίων.
- Η XML Schema (όπως και τα DTD) είναι γλώσσα περιγραφής σχήματος.
- Η XML Schema προσφέρει χαρακτηριστικά και δυνατότητες, ισχυρότερα αυτών που παρέχονται από τα DTD.

XML Schema: Παράδειγμα

- Ζητάμε περιγραφή σε XML Schema της δομής των τεκμηρίων της μομφής:

```
<TAB>
   <φοιτητής>
            <όνομα> Νίκος </όνομα>
            <επώνυμο> Νικολάου
</επώνυμο>
   </φοιτητής>
   <φοιτητής> … </φοιτητής>
            …
</TAB>
```

XML Schema: Παράδειγμα (συνέχεια)



Namespaces Σύγκρουση ονομάτων

- <table>
- <tr>
- <td>Apples</td>
- <td>Bananas</td>
- </tr>
- </table>

Σύγκρουση ονομάτων

- <table>
- <name>African Coffee Table</name>
- <width>80</width>
- <length>120</length>
- </table>

Xml:ns

- <root
- xmlns:h="http://www.w3.org/TR/html4/" xmlns:f=http://www.w3schools.com/furniture
- >
  <h:table>
- …
- <f:table>
- </root>

Χαρακτηριστικά της XML Schema

- Ένα τεκμήριο σε XML Schema είναι ένα XML τεκμήριο.
- Τα στοιχεία στο XML Schema του παραδείγματος έχουν το πρόθεμα **xs** το οποίο συνδέεται με το χώρο ονομάτων της XML Schema μέσω της δήλωσης:
  - xmlns:xs="http://www.w3.org/2001/XMLSchema".
- Η XML Schema παρέχει στοιχεία (όπως τα element, sequence, complexType) με συγκεκριμένη σημασία, τα οποία μαζί με αντίστοιχα γνωρίσματα (όπως τα name, type, minOccurs, maxOccurs)

Παράδειγμα (εναλλακτική περιγραφή)

- Η ακόλουθη περιγραφή σε XML Schema περιγράφει ακριβώς την ίδια κατηγορία τεκμηρίων με την προηγούμενη περιγραφή:

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="TAB" type="TABtype"/>
    <xs:complexType name="TABtype">
        <xs:element name="φοιτητής" type="studentType"
                    minOccurs=0 maxOccurs="unbounded"/>
    </xs:complexType>
    <xs:complexType name="studentType">
        <xs:sequence>
            <xs:element name="όνομα" type="xs:string"/>
            <xs:element name="επώνυμο" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

Δηλώσεις στοιχείων

- Στοιχεία της XML:
  - Σύνθετου τύπου: περιέχουν υποστοιχεία ή διαθέτουν γνωρίσματα.
  - Απλού τύπου: δεν έχουν υποστοιχεία, ούτε διαθέτουν γνωρίσματα.
- Τα γνωρίσματα θεωρούνται ότι είναι απλού τύπου.
- Η XML Schema παρέχει το στοιχείο complexType για τη δημιουργία νέων σύνθετων τύπων. Οι σύνθετοι τύποι που δημιουργούνται αποκτούν όνομα μέσω του προαιρετικού γνωρίσματος name.
  - Οι τύποι TABtype και studentType είναι σύνθετοι.
- Με την ακόλουθη έκφραση ορίζεται ο σύνθετος τύπος TABtype :

```
<xs:complexType name="TABtype">
        ...
</xs:complexType>
```

  Το TABtype χρησιμοποιείται σε άλλο σημείο της περιγραφής σχήματος για να δηλώσουμε ότι ένα στοιχείο είναι τύπου TABtype.
  - **Παράδειγμα:** Στη δήλωση:

```
<xs:element name="TAB" type="TABtype"/>
```

    δηλώνεται ότι το στοιχείο TAB είναι τύπου TABtype.
- Το όνομα ενός σύνθετου τύπου μπορεί χρησιμοποιηθεί στη δήλωση πολλών στοιχείων μειώνοντας έτσι το μέγεθος των περιγραφών.

Δηλώσεις γνωρισμάτων

- Για τη δήλωση γνωρισμάτων χρησιμοποιείται το στοιχείο attribute. Ένα γνώρισμα δηλώνεται ως εξής:

```
<xs:attribute name=" ... " type="..."  .... />
```

  - **Παράδειγμα:** Με την έκφραση

```
<xs:attribute name="ηλικία"
type="xs:positiveInteger"
    use="required"/>
```

    δηλώνεται το γνώρισμα ηλικία το οποίο παίρνει τιμές του τύπου positiveInteger.
- Ο τύπος positiveInteger, είναι απλός τύπος και αντιπροσωπεύει τους θετικούς ακέραιους αριθμούς.
- Η παράσταση use="required", δηλώνει ότι η εμφάνιση του συγκεκριμένου γνωρίσματος είναι υποχρεωτική.

Δηλώσεις γνωρισμάτων

- **Παράδειγμα:** Το στοιχείο φοιτητής στην παρακάτω δήλωση εμφανίζεται να διαθέτει το γνώρισμα ηλικία:

```
<xs:element name="φοιτητής" minOccurs=0
 maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="όνομα"
type="xs:string"/>
      <xs:element name="επώνυμο"
type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="ηλικία"
type="xs:positiveInteger"
    use="required"/>
  </xs:complexType>
```

Συχνότητα εμφάνισης στοιχείων

- Η XML Schema επιτρέπει να προδιαγράψουμε το πλήθος των εμφανίσεων των στοιχείων με εξαιρετική ακρίβεια. Αυτό γίνεται μέσω δύο γνωρισμάτων του element:
  - Του minOccurs μέσω του οποίου δηλώνουμε τον ελάχιστο αριθμό εμφανίσεων του στοιχείου,
  - Του maxOccurs μέσω του οποίου δηλώνουμε τον μέγιστο αριθμό εμφανίσεων του στοιχείου.
  - Οι τιμές των γνωρισμάτων αυτών είναι μη αρνητικοί ακέραιοι.
  - Όταν κάποιο από τα minOccurs ή maxOccurs παραλείπεται (και τα δύο ή ένα από αυτά) τότε

Περιορισμοί συχνότητας εμφάνισης στοιχείων και γνωρισμάτων

- Ένα γνώρισμα μπορεί να εμφανίζεται σε ένα στιγμιότυπο στοιχείου μία φορά ή να μην εμφανίζεται καθόλου. Δεν μπορεί όμως το ίδιο γνώρισμα να εμφανίζεται στο ίδιο στοιχείο περισσότερες από μια φορές (με τη ίδια ή διαφορετική τιμή).
- Η XML Schema επιτρέπει να προσδιορίσουμε την υποχρεωτικότητα ή μη της εμφάνισης ενός γνωρίσματος

Απλοί τύποι

- Η XML Schema διαθέτη πλούσια συλλογή ενσωματωμένων απλών τύπων όπως οι byte, integer, positiveInteger, negativeInteger, int, decimal, long, float, double, boolean, date, dateTime, ID, string, IDREF, IDREFS, ENTITY, ENTITIES, NMTOKEN, NMTOKENS.
- Η XML Schema επιτρέπει να ορίσουμε νέους απλούς τύπους μέσω του στοιχείου simpleType. Οι δηλώσεις περιγράφουν τον τρόπο που παράγονται οι νέοι τύποι από άλλους απλούς (ενσωματωμένους ή παραγόμενους) τύπους.
  - **Παράδειγμα.** Με τη δήλωση:

```
<xs:simpleType name="myInteger">
    <xs:restriction base="xs:integer">
                <xs:minInclusive value="0"/>
```

Απλοί τύποι

- Η XML Schema παρέχει πλούσια ποικιλία «*όψεων*» (facets) (όπως οι minInclusive και maxInclusive) για την επιβολή περιορισμών κατά τον ορισμό νέων τύπων.
- Μια χρήσιμη όψη είναι το στοιχείο enumeration που περιορίζει έναν απλό τύπο σε ένα σύνολο διακριτών τιμών:
  - **Παράδειγμα.** Στην περιγραφή:

```
<xs:simpleType name="νόμισμα">
    <xs:restriction base="xs:string">
        <xs:enumeration value="GRD"/>
            <xs:enumeration value="EURO"/>
            <xs:enumeration value="USD"/>
                <!-- κ.λ.π. ... -->
    </xs:restriction>
```

Ορισμός γνωρισμάτων σε στοιχεία απλού τύπου

- Στοιχεία απλού τύπου δηλώνονται όπως στο παράδειγμα:
  <xs:element name="ονοματεπώνυμο" type="xs:string"/>
- Οι απλοί τύποι (στοιχεία απλού τύπου) δεν επιτρέπεται να έχουν γνωρίσματα. Η επισύναψη γνωρισμάτων σε στοιχεία απλού τύπου (π.χ. το γνώρισμα ΑΔΤ-αριθμός δελτίου ταυτότητας στο στοιχείο ονοματεπώνυμο) γίνεται με τον ορισμό σύνθετου τύπου όπως στο επόμενο παράδειγμα. Το περιεχόμενο του στοιχείου παραμένει απλού τύπου (του τύπου string στην περίπτωση μας).

```
<xs:element name="ονοματεπώνυμο">
<xs:complexType>
 <xs:simpleContent>
      <xs:extension base="xs:string">
            <xs:attribute name="ΑΔΤ" type="xs:string"/>
      </xs:extension>
 </xs:simpleContent>
</xs:complexType>
</xs:element>
```

Το simpleContent υποδηλώνει στοιχεία με απλό περιεχόμενο χωρίς υποστοιχεία (στο παράδειγμα ακολουθία απλών χαρακτήρων). Για τον ορισμό του νέου τύπου, επεκτείνουμε τύπο string. Η επέκταση συνίσταται στην

Ομαδοποίηση στοιχείων

- Η XML Schema παρέχει τη δυνατότητα να ορίζουμε ομάδες στοιχείων (στις οποίες μπορούμε να δίνουμε και ονόματα). Στόχος η χρήση αυτών των ομάδων για τη δόμηση του περιεχόμενου των συνθέτων τύπων.

Ομαδοποίηση στοιχείων με το sequence

- Το στοιχείο sequence χρησιμοποιείται για να δηλώσει (διατεταγμένη) ακολουθία (υπο)στοιχείων.
- **Παράδειγμα.** Στο παρακάτω:

```
<xs:complexType name="studentType">
    <xs:sequence>
        <xs:element name="όνομα"
type="xs:string"/>
        <xs:element name="επώνυμο"
type="xs:string"/>
        <xs:element name="πατρώνυμο"
type="xs:string"/>
    </xs:sequence>
</xs:complexType>
```

δηλώνεται ότι κάθε στοιχείο τύπου studentType πρέπει να περιλαμβάνει ένα υποστοιχείο όνομα

Ομαδοποίηση στοιχείων με το choice

- Το στοιχείο choice επιτρέπει την εμφάνιση κάθε φορά ενός μόνο από τα (υπο)στοιχεία που περιλαμβάνει, ως περιεχόμενο του στοιχείου που ανήκει στον συγκεκριμένο τύπο.
- **Παράδειγμα.** Στο παρακάτω τμήμα περιγραφής:

```
<xs:complexType name="studentType">
    <xs:choice>
        <xs:element name="ονοματεπώνυμο" type="xs:string"/>
        <xs:sequence>
            <xs:element name="όνομα" type="xs:string"/>
            <xs:element name="επώνυμο" type="xs:string"/>
        </xs:sequence>
    </xs:choice>
</xs:complexType>
```

δηλώνεται ότι κάθε στοιχείο του τύπου studentType έχει ως περιεχόμενο είτε το στοιχείο ονοματεπώνυμο είτε ένα στοιχείο όνομα ακολουθούμενο από ένα στοιχείο επώνυμο.

Ομαδοποίηση στοιχείων με το all

- Με το στοιχείο all δηλώνεται ότι κάθε στοιχείο της ομάδας εμφανίζεται υποχρεωτικά (ή προαιρετικά αν συνοδεύεται με το γνώρισμα minOccurs=0). Η σειρά εμφάνισης δεν παίζει ρόλο. Επιτρεπτές τιμές των minOccurs και maxOccurs στις δηλώσεις των στοιχείων του all είναι οι 0 και 1.
  - **Παράδειγμα.** Στο παρακάτω τμήμα περιγραφής:

```
<xs:complexType name="studentType">
    <xs:all>
      <xs:element name="όνομα"
type="xs:string"/>
```

Ομαδοποίηση στοιχείων με το all (συνέχεια)

- Το all πρέπει να εμφανίζεται ως το μοναδικό υποστοιχείο του complexType.
  - **Παράδειγμα.** Δεν είναι επιτρέπονται δηλώσεις της μορφής:

```
<xs:complexType name="studentType">
    <xs:all>
        <xs:element name="όνομα"
type="xs:string"/>
        <xs:element name="επώνυμο"
type="xs:string"/>
        <xs:element name="πατρώνυμο"
type="xs:string"/>
    </xs:all>
    <xs:sequence>
        <xs:element name="ηλικία"
type="xs:positiveInteger"/>
```

Ομαδοποίηση γνωρισμάτων

- Το στοιχείο attributeGroup επιτρέπει ομαδοποίηση γνωρισμάτων.
- **Παράδειγμα.** Εδώ ορίζεται ομάδα γνωρισμάτων με όνομα personAttributes. Περιλαμβάνει το γνώρισμα ΑΔΤ που είναι τύπου string, το ΑΦΜ που είναι τύπου positiveInteger, και το φύλο για το οποίο ορίζεται ανώνυμος τύπος που περιλαμβάνει τις τιμές άνδρας και γυναίκα:

```
<xs:attributeGroup name="personAttributes">
  <xs:attribute name="ΑΔΤ"  type="xs:string" use="required"/>
  <xs:attribute name="ΑΦΜ" type="xs:positiveInteger"/>
  <xs:attribute name="φύλο">
       <xs:simpleType>
            <xs:restriction base="xs:string">
                 <xs:enumeration value="άνδρας"/>
                 <xs:enumeration value="γυναίκα"/>
            </xs:restriction>
       </xs:simpleType>
  </xs:attribute>
</xs:attributeGroup>
```

Ομαδοποίηση γνωρισμάτων (συνέχεια)

- Το όνομα ομάδας γνωρισμάτων μπορεί να χρησιμοποιηθεί για να δηλωθεί ότι ένα στοιχείο διαθέτει τα γνωρίσματα που περιλαμβάνει η συγκεκριμένη ομάδα.
- **Παράδειγμα.** Στην παρακάτω περιγραφή γίνεται επίκληση της δήλωσης του συνόλου γνωρισμάτων personAttributes, μέσω του γνωρίσματος ref, προκειμένου να δηλώσουμε ότι το στοιχείο φοιτητής διαθέτει τα γνωρίσματα του συνόλου personAttributes:

```
<xs:element name="φοιτητής" minOccurs=0 maxOccurs="unbounded" >
    <xs:complexType>
        <xs:sequence>
            <xs:element name="όνομα" type="xs:string"/>
            <xs:element name="επώνυμο" type="xs:string"/>
        </xs:sequence>
       <xs:attributeGroup ref="personAttributes"/>
   </xs:complexType>
</xs:element>
```

Ομαδοποίηση γνωρισμάτων (συνέχεια)

- Για να ορίσουμε και άλλα στοιχεία που διαθέτουν την ίδια ομάδα γνωρισμάτων (π.χ. τα στοιχεία μέλοςΔεπ, εργαζόμενος κ.λ.π.), χρησιμοποιούμε δηλώσεις ορισμού στοιχείων παρόμοιες με αυτήν του στοιχείου φοιτητής, στις οποίες θα υπάρχει απλή αναφορά στην ίδια ομάδα γνωρισμάτων, χωρίς να χρειάζεται να οριστεί ξανά αυτή.
- Χρησιμοποιώντας ομάδες γνωρισμάτων διαμορφώνουμε πιο ευανάγνωστες περιγραφές σχήματος και διευκολύνουμε τη μελλοντική ενημέρωση του σχήματος, αφού μια ομάδα γνωρισμάτων μπορεί να

Περιληψη

- Το *XML Σχήμα* (*XML Schema*) ενοποιεί προηγούμενες προτάσεις για περιγραφή σχήματος.
- Γενικεύει τα DTDs.
- Χρησιμοποιεί τη σύνταξη της XML.
- Τεχνικά εγχειρίδια:
  - «*XML Schema Part 0: Primer*» βρίσκεται στη διεύθυνση: *http://www.w3.org/TR/xmlschema-0*
  - «*XML Schema Part 1: Structures*»

Validation

Εγκυρότητα

- Συντακτική
  - Well-formed
- Γραμματική – Σχήματος
  - Ικανοποιεί τους περιορισμούς του σχήματος

Άσκηση

- Για το xml που αναπτύξατε στο προηγούμενο εργαστήριο
  - Ελέγξτε το συντακτικά
  - Δημιουργήστε αντίστοιχα
    - DTD
    - XSD
  - Εφαρμόστε το και ελέγξτε την εγκυρότητα του
  - Εάν προκύψουν σφάλματα διορθώστε τα μέχρι να είναι έγκυρο