

# Az XML modell

Az előadás anyaga

Prof. Dr. Kovács László: Adatkezelés XML  
környezetbe

diasor alapján készült el

# Témakörök

1. Az XML modell kialakulása.
2. Az XML modell alapelemei.
3. XML dokumentum szerkezete.
4. Helyesen formált XML.
5. XDM modell.
6. Speciális karakterek használata.

# Témakörök

7. Névterek szerepe.
8. Névterek használata.
9. ER modell konverziója XDM modellre.

# Igényelt kompetenciák

- XML hierarchia alkalmazása.
- Helyesen formált XML előállítása.
- Névterek használata.
- XDM modell készítése.

*Fejlesztőkörnyezet:* XML szerkesztő (Oxygen, VS Code, IntelliJ IDEA Editor 2021, Eclipse stb....)

<https://github.com/BLaszlo2020/NEPTUNKODWebXMLLev>

# Adatmennyiség robbanása

- Internet of Things.
- Intelligens szenzorok.
- Intelligens gyártás.
- Intelligens eszközök.
- Személyes adatok.
- Adatbányászat.

2012-ben 1 nap alatt 2,5 exabyte ([maxer.hu](http://maxer.hu))

2025-ben 1 nap alatt 463 exabyte

[https://www.dxc.com/insights/flxwd/78931-big\\_data\\_universe\\_beginning](https://www.dxc.com/insights/flxwd/78931-big_data_universe_beginning)

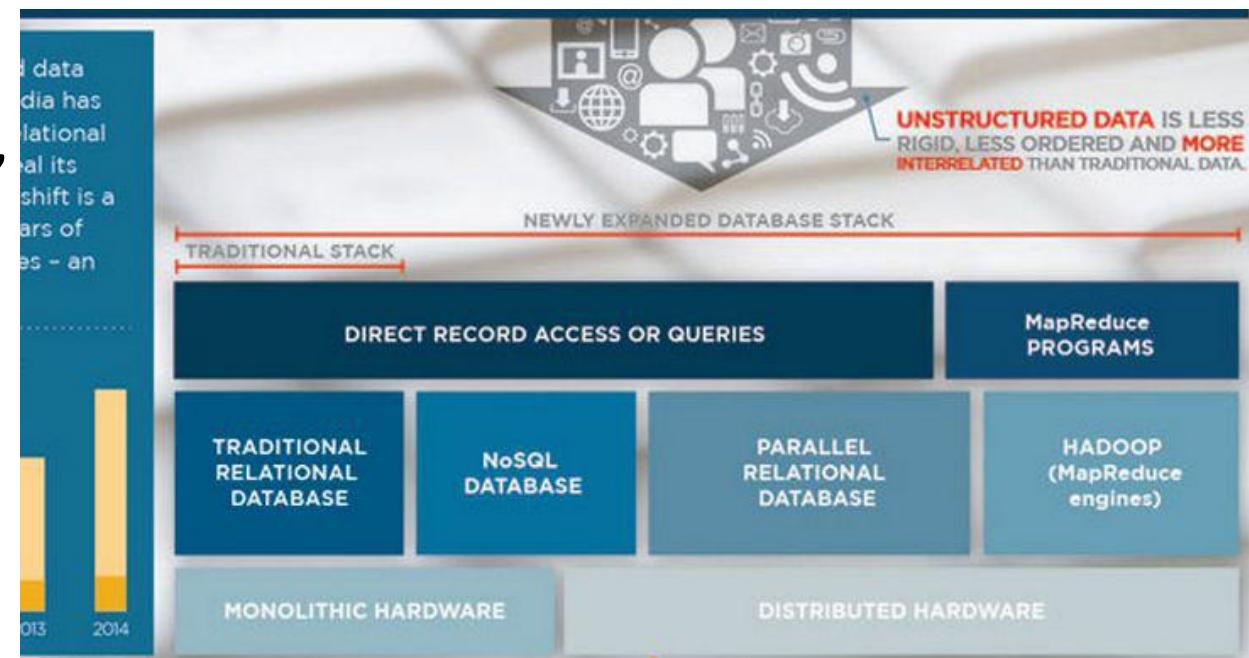


Byte mértékegységek		m · v · sz		
SI-prefixumok		Értéke	Bináris prefixumok	
Név (Szimbólum)	SI Standard		Név (Szimbólum)	Értéke
kilobyte (kB)	$10^3 = 1000^1$	$2^{10}$	kibibyte (KiB)	$2^{10}$
megabyte (MB)	$10^6 = 1000^2$	$2^{20}$	mebibyte (MiB)	$2^{20}$
gigabyte (GB)	$10^9 = 1000^3$	$2^{30}$	gibibájt (GiB)	$2^{30}$
terabyte (TB)	$10^{12} = 1000^4$	$2^{40}$	tebibájt (TiB)	$2^{40}$
petabyte (PB)	$10^{15} = 1000^5$	$2^{50}$	pebibájt (PiB)	$2^{50}$
exabyte (EB)	$10^{18} = 1000^6$	$2^{60}$	exbibájt (EiB)	$2^{60}$
zettabyte (ZB)	$10^{21} = 1000^7$	$2^{70}$	zebibájt (ZiB)	$2^{70}$
yottabyte (YB)	$10^{24} = 1000^8$	$2^{80}$	yobibájt (YiB)	$2^{80}$

# Adatkezelés fejlődése

- állomány kezelés,
- fileszerverek,
- *lokális relációs adatbázisok*,
- elosztott adatbázisok (DDBMS),
- NoSQL adatbázisok,
- felhő alapú adatbázisok.

(Forrás: KovácsL)



# Mi az XML?

- *Szűkebb értelemben:*
  - **Strukturált dokumentum**, mely lehetővé teszi az automatikus feldolgozást (elektronikus dokumentum formátum).
- *Tágabb értelemben:*
  - Egy sereg közös tőről fakadó **specifikációt jelent**, melyeket összefoglaló néven **XML családnak** is neveznek.

# XML - elnevezés

- Az *Extensible Markup Language* (XML) kifejezés szó szerint jelentése „Kiterjeszthető Jelölőnyelv”.
  - Kiterjeszthető, mert *nincs előre definiált címkekészlet* a szerkezeti alkotórészek azonosítására, hanem egy *mechanizmust biztosít* a címkekészletek *definiálásához*.

# Az XML története - előzmény

- Előzménye az SGML.
  - ISO szabvány elektronikus dokumentumok leírására (1980-as évek).
    - *ISO 8879:1986 Information processing – Text and office systems – Standard Generalized Markup Language (SGML)*
  - Bonyolult, webes környezetben történő széleskörű használatra nem alkalmas.

# Az XML története

- előzményei már a hatvanas évek végén megjelentek (GML, SGML) – kevés sikерrel,
- az alapszabvány 1998-ban született meg,
- kiegészítő szabványai azóta folyamatosan,
- ipari elterjedése már 2000-től jelentős (web, e-üzlet, ...).

# Mi az XML? - jellemzők

- Szöveges dokumentum formátum.
- Metanyelv.
- Technológia.
- Szabványcsalád.

# Mi az XML? - Szöveges dokumentum formátum

## *„a) Dokumentum formátum szabványok*

- de facto (gyakorlatban): DOCX, PDF, RTF, ...
- de jure (törvény szerint): HTML, SGML, XML, ODF, ...

## *b) Strukturált dokumentum*

- dokumentum definíció (DTD),
- strukturált dokumentum *tartalom*,
- dokumentum *megjelenés*.

# Mi az XML? - Szöveges dokumentum formátum

## c) Jelölő nyelv

- a struktúra ábrázolása *címkékkel*,
- a címkék típusai: *procedurális* és *leíró* (szematinkus).

*Cél:* az automatikus feldolgozhatóság, ehhez a *jelölőnyelveket* használnak a *szerkezeti alkotórészek azonosítását*.

# Mi az XML? - Szöveges dokumentum formátum

## XML példák:

- a)** <név>Informatika Intézet</név>
- b)** <ember>  
    <név>Informatika Intézet</név>  
  </ember>
- c)** Hétfőn <ember><név>Informatika  
Intézetbe</név></ember> vizsgázott.
- d)** <p>  
    Ez egy <i>dőlt betűvel</i> írt szöveg.  
  </p>

# Mi az XML? - Szöveges dokumentum formátum

## XML példák:

**a)** Hétfőn <ember><név>az Informatika  
Intézetbe</név></ember> **<i>jelesre</i>**  
vizsgázott.

**b)** Hétfőn <ember><név>Informatika  
Intézetbe</név></ember>  
**<érdemjegy>jelesre</érdemjegy>** vizsgázott.

# Mi az XML? - Szöveges dokumentum formátum

XML példa

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<wsdl:definitions name="DiscoService"
    targetNamespace="http://disco.mit.bme.hu/service" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:tns="http://disco.mit.bme.hu/service" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
    <wsdl:types>
        <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            targetNamespace="http://disco.mit.bme.hu/service">
            <xsd:complexType name="DiscoLanguage">
                <xsd:sequence>
                    <xsd:element name="id" type="xsd:int"></xsd:element>
                    <xsd:element name="locale" type="xsd:string"></xsd:element>
                    <xsd:element name="lang_id" type="xsd:int"></xsd:element>
                    <xsd:element name="name" type="xsd:string"></xsd:element>
                </xsd:sequence>
            </xsd:complexType>
            <xsd:complexType name="DiscoLanguageList">
                <xsd:sequence>
                    <xsd:element ref="tns:DiscoLanguage" minOccurs="0" maxOccurs="unbounded"></xsd:element>
                </xsd:sequence>
            </xsd:complexType>
            <xsd:complexType name="DiscoTerm">
                <xsd:sequence>
                    <xsd:element name="id" type="xsd:int"></xsd:element>
                    <xsd:element name="lang_id" type="xsd:int"></xsd:element>
                    <xsd:element name="code" type="xsd:string"></xsd:element>
                    <xsd:element name="name" type="xsd:string"></xsd:element>
```

# Mi az XML? - Metanyelv

*Metanyelv*: lehetővé válik a *jelölt tartalom* gépi „megértése”.

- címkekészlet és struktúra (szókincs és nyelvtan) határoz meg,
- egy *deklaráció* (DTD, XML Schema) rögzíti a nyelvet.

XML deklaráció példák:

a) `<!ELEMENT ember (név)>`

b) `<complexType name="ember">`  
`<sequence>`  
    `<element name="név" type="string"></element>`  
`</sequence>`  
`</complexType>`

# Az XML - mint jelölőnyelv

A **jelölőnyelvek** szöveg annotálására szolgáló számítógépes nyelvek.

- Példák:
  - TeX, LaTeX, <https://www.latex-project.org/>
  - Markdown, <https://daringfireball.net/projects/markdown/>
  - Wikitext, <https://en.wikipedia.org/wiki/Help:Wikitext>

# Mi az XML? - Technológia

*Technológia, mert tartalmaz szabványokat és termékeket a strukturált dokumentumok készítésére, feldolgozására és megjelenítésére.*

Sokféle alkalmazási területen nyújt eszközöket és módszereket a feladatok megoldásához.

- *webes megjelenés* (szerver és kliens oldali transzformációk),
- *adatcsere* (formátum, transzformáció) – e-Business,

# Mi az XML? - Technológia

- szövegek reprezentációja és feldolgozása,
- szövegszerkesztők dokumentum formátuma (OpenOffice, MS Office),
- Web 2.0,
- technikai dokumentációk nyelvezete,
- szoftverek konfigurálása (ant, maven),
- EU önéletrajzok készítése (Europass).
- ....

# Mi az XML? - Szabványcsalád

*Több szervezet által is jegyzett alapszabványokból épül fel.*

Legismertebbek:

*a) ISO/IEC JTC 1 által gondozott szabványok*

- Standard Generalized Markup Language (SGML).
- Document Schema Definition Languages (DSDL) - meghatározza a DTD és XML Schema-t .
- HyperText Markup Language (HTML).

# Mi az XML? - Szabványcsalád

*b)Organization for the Advancement of Structured Information Standards (**OASIS**)*

- Business Centric-Methodology (BCM), a SOA alapjai.
- Universal Description Discovery and Integration (UDDI).
- Security Assertion Markup Language (SAML) - az LDAP leváltásra.
- DocBook, DITA XML - a formátum.

# Mi az XML? - Szabványcsalád

## c) World Wide Web Consortium (**W3C**)

- Extensible Markup Language (XML) (1998, 2004, 2008)
- XML Namespaces (1999, 2004, 2009)
- XML Schema (2001, 2004, 2012)
- DOM (1998, 2000, 2004, 2008)
- XML Path Language (XPath) (1999, 2007, 2010)
- XSL Transformations (XSLT) (1999, 2007, 2010)
- XHTML (2000, 2001, 2007, 2010)
- XQuery: An XML Query Language (2007, 2010),, (Forrás: Adamko)

# XML - Előnyök

- Könnyen *alkalmazható webes rendszerekben*.
- Keresés (web) helyett *lekrdezéssel* (DB) juthatunk információhoz.
- *Univerzális adatcsere formátum*.
- Gyártófüggetlenség.
- Platformfüggetlenség.
- Az iparban de-facto szabvány.....

# XML - Hátrányok

- Az eredeti szabvány *gyenge deklarációs rendszert* tartalmaz (DTD).
- A deklaráció tervezési hibáinak javítása *igen költséges*.
- *Bőbeszédű* és *nehézkesen* használható szintaxis.
- Nagy tárigény.
- Bonyolultság.
  - Se szeri, se száma az XML-hez *kötődő specifikációknak*.
- Mindezek ellenére fontos, együtt kell élni vele.

# XML - Alternatíva

- JSON (*JavaScript Object Notation*) <http://www.json.org/>
  - Könnyűsúlyú szöveges nyelvfüggetlen *adatcsere formátum*.
  - *Strukturált adatok ábrázolására* szolgál, esetükben jelenti az XML alternatíváját.
  - Nagyjából ugyanazokat az előnyöket kínálja, mint az XML, azonban annak hátrányai nélkül.

# BSON („Binary JSON”)

Bináris *adatcsere* formátum.

- Specifikáció: <http://bsonspec.org/>

A *MongoDB* NoSQL adatbázis-kezelő rendszer használja.

URL: <https://www.mongodb.org/>

- *Adattárolás* és *hálózati adatátvitel*.

# YAML formátum

*YAML formátum* (YAML - Nem Markup Language), amely egy újabb módja az *adatok tárolásának*.

A fájl kiterjesztése: .yml

A YAML egy adat **sorosító nyelv** (szerIALIZÁció), - közvetlenül olvasható és írható emberi szemmel.

*Célja:* a memóriában tárolt adatok egyszerű lemezre mentése és visszatöltése.

# Az XML család – tágabb értelemben

**Az XML-lel kapcsolatos specifikációk:** Az XML lehetőségeit bővítik.

- Lehetővé teszik XML dokumentumok szerkezetére és *tartalmára vonatkozó megszorításokat* (XML sémanyelvek).
- Lehetővé teszik XML dokumentumokból *információ kinyerését* (lekérdező nyelvek).
- Lehetővé teszi XML dokumentumok más *formába alakítását* (transzformációs nyelvek).

# Az XML család

- **Alkalmazásprogramozási interfészek (API-k):**
  - Lehetővé teszik XML dokumentumok feldolgozását programnyelvekből (pl.: DOM, JAXB, JAXP, JDOM, SAX, StAX).

# XML sémanyelvek

- XML dokumentumok szerkezetére és tartalmára vonatkozó megszorítások kifejezésére szolgálnak.
- Adott sémá(k)nak megfelelő XML dokumentumot érvényesnek (validate) mondunk.
  - A sémá(k)nak való megfelelés ellenőrzésének folyamata az érvényesítés.

# XML sémanyelvek

Kortárs XML séma nyelvek:

- *Dokumentumtípus-definíció* (DTD),
- *W3C XML Schema*, <https://www.w3.org/XML/Schema>
- RELAX NG <https://relaxng.org/>
- Schematron [http://www.schematron.com/”](http://www.schematron.com/)

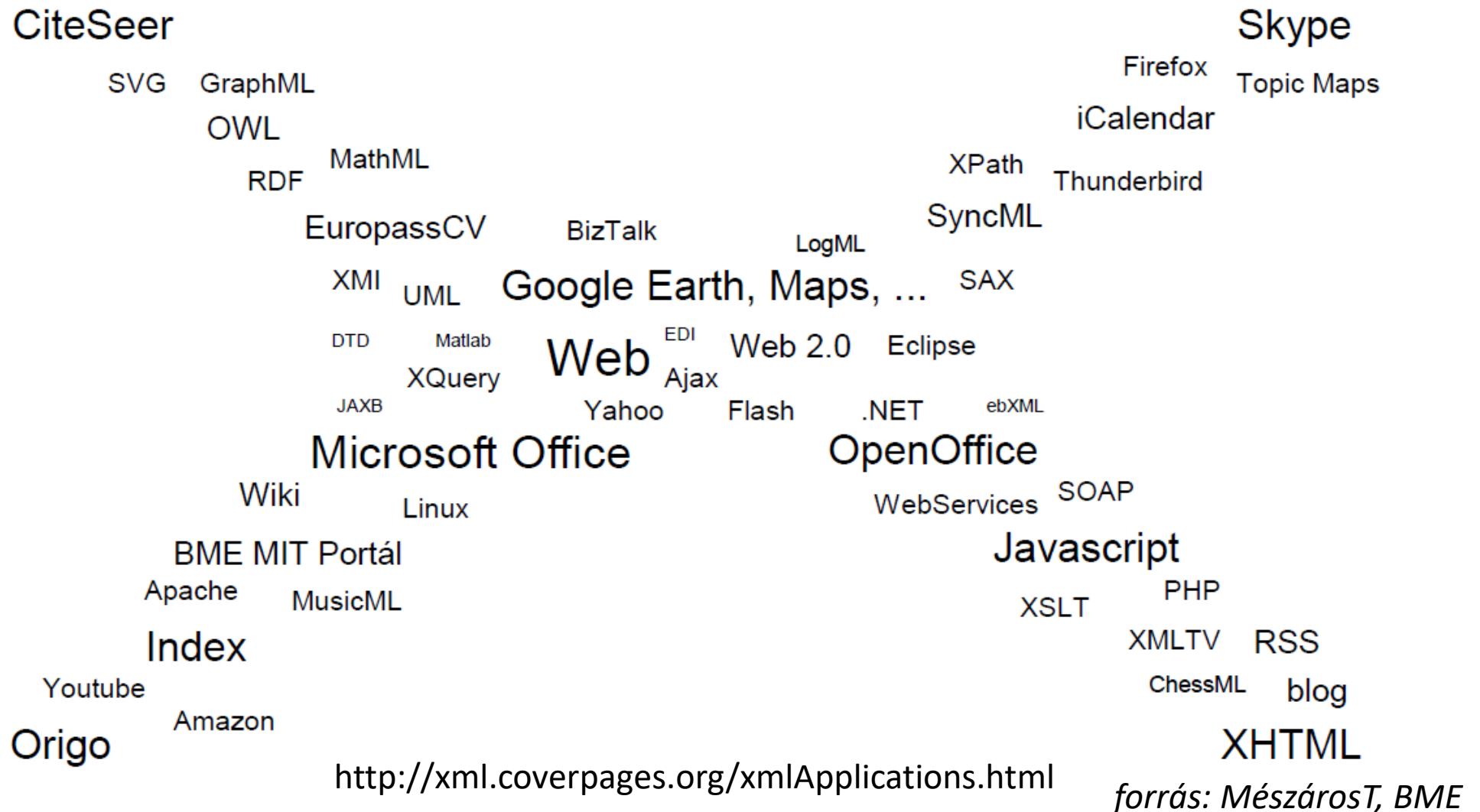
# Lekérdező nyelvek

- *XML Path Language (XPath) Version 1.0* (W3C ajánlás, 1999. november 16.)  
<https://www.w3.org/TR/1999/REC-xpath-19991116/>
- *XML Path Language (XPath) 3.1* (W3C ajánlás, 2017. március 21.)  
<https://www.w3.org/TR/xpath-31/>
- *XQuery 3.1: An XML Query Language* (W3C ajánlás, 2017. március 21.)  
<https://www.w3.org/TR/xquery-31/>

# Transzformációs nyelvek

- *XSL Transformations (XSLT) Version 1.0* (W3C ajánlás, 1999. november 16.)  
<https://www.w3.org/TR/1999/REC-xslt-19991116>
- *XSL Transformations (XSLT) Version 3.0* (W3C ajánlás, 2017. június 8.)  
<https://www.w3.org/TR/xslt-30/>
- *XQuery 3.1: An XML Query Language* (W3C ajánlás, 2017. március 21.)  
<https://www.w3.org/TR/xquery-31/>

# Ki használ XML-t?

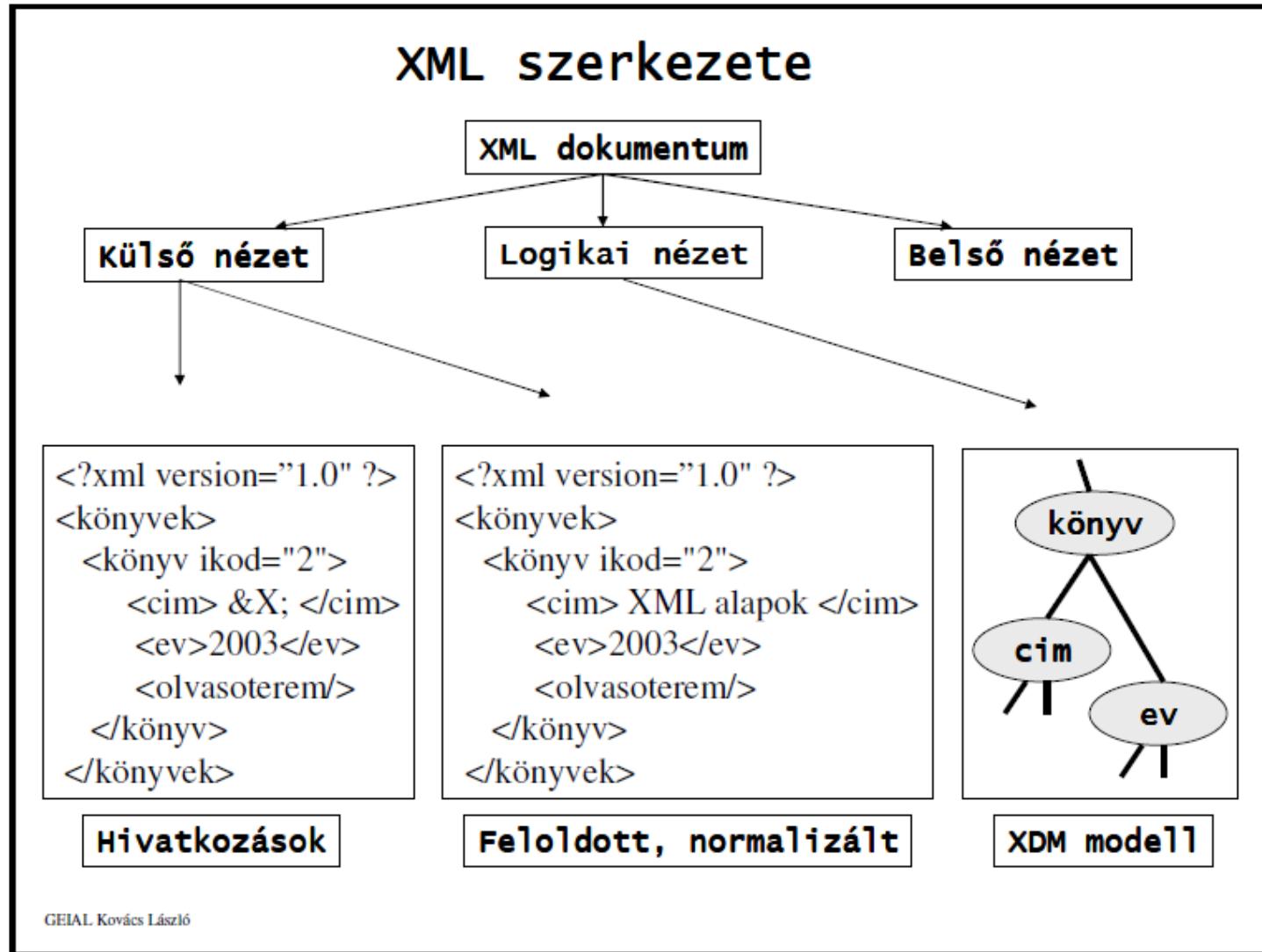


# Irodalom

- Dr. Adamkó Attila: Fejlett Adatbázis Technológiák - Jegyzet, 2013.
- Jeszenszky Péter: XML bevezető, DE, 2019.
- Mészáros Tamás: Az XML technológia, BME, 2014.

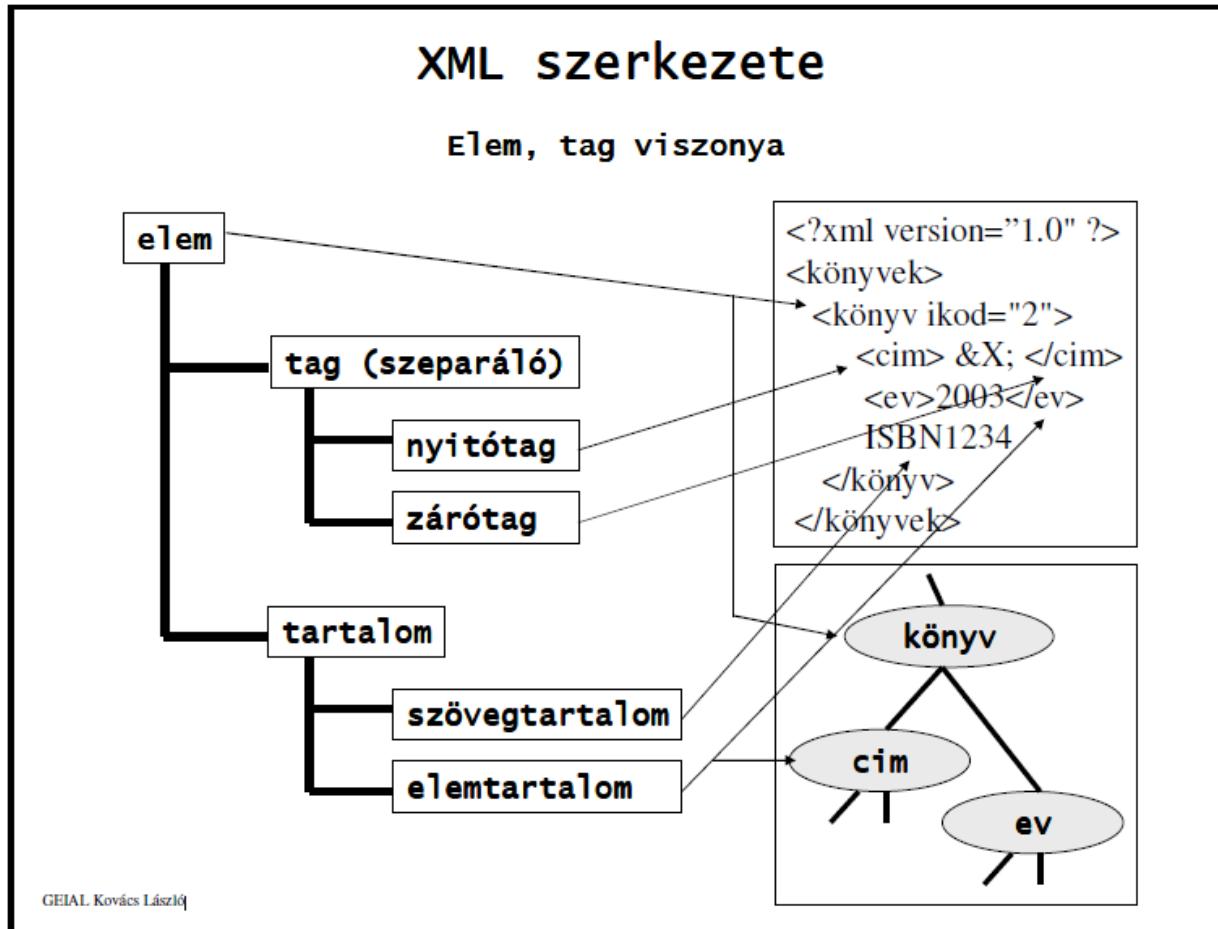
# Az XML dokumentum szerkezete

# XML dokumentum szerkezete



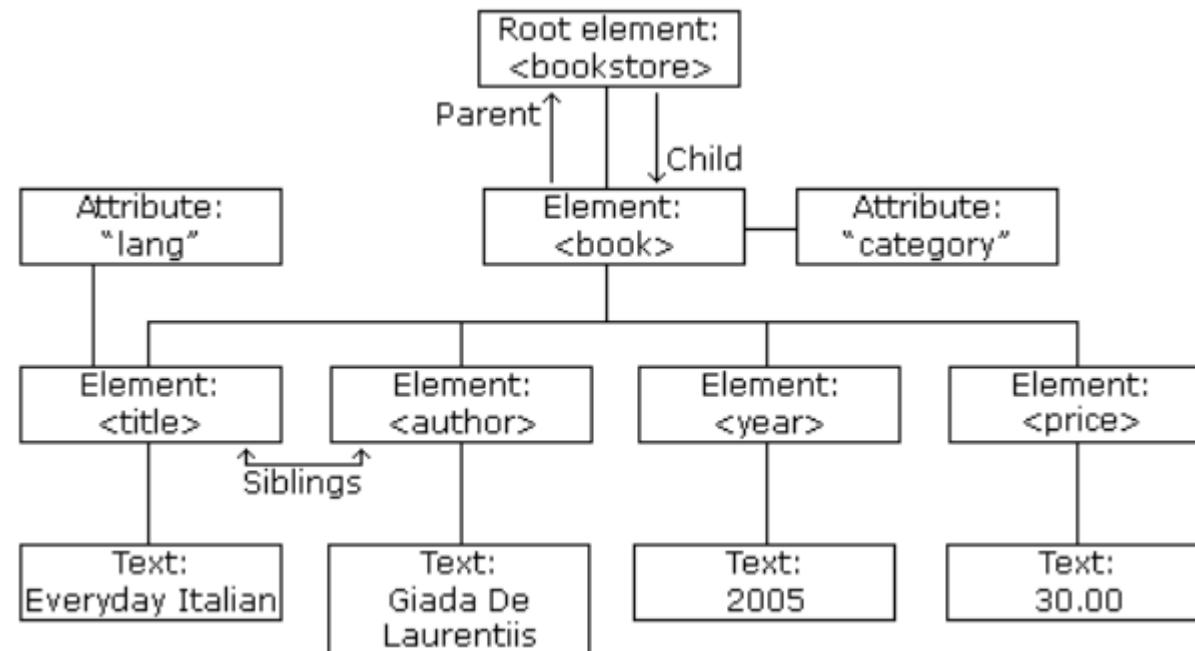
# XML dokumentum szerkezete

Elem,  
tag  
viszonya



# XML fa felépítése

Az XML dokumentumok *egy fastruktúrát* alkotnak, amely a "gyökérnél" kezdődik és a "levelekhez" ágazik.



# XML fa felépítése

Példa XML dokumentumra:

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
    <book category="cooking">
        <title lang="en">Everyday Italian</title>
        <author>Giada De Laurentiis</author>
        <year>2005</year>
        <price>30.00</price>
    </book>
    <book category="children">
        <title lang="en">Harry Potter</title>
        <author>J. K. Rowling</author>
        <year>2005</year>
        <price>29.99</price>
    </book>
    <book category="web">
        <title lang="en">Learning XML</title>
        <author>Erik T. Ray</author>
        <year>2003</year>
        <price>39.95</price>
    </book>
</bookstore>
```

# XML dokumentum felépítése – elemek közötti kapcsolatok

- Az XML fa egy **gyököből** indul, a gyermek elembe.
- Az *elemek közötti kapcsolatok* a testvér kifejezéseket jelentik.
- A szülőknek vannak gyermekai. A testvérek a szülők gyermekai.
- minden elem tartalma van, és attribútumokat (írunk).

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="web">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

gyökérből  
gyermek és  
gyermek (egy szülője  
testvérek).  
at (Harry Potter)

# XML dokumentum felépítése

- Az első sor a feldolgozás

<?xml version='

- A következő sor a dokumentum

<

A következő sor egy <book>

<book ca

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J. K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="web">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

# XML dokumentum felépítése

A `<book>` elemeknek 4 g

```
<title>, <auth  
<title lang="en">Ever  
<author>Giada De Laur  
<year>2005</year>  
<price>30.00</price>
```

A következő sor befejezi a /

```
<?xml version="1.0" encoding="UTF-8"?>  
<bookstore>  
  <book category="cooking">  
    <title lang="en">Everyday Italian</title>  
    <author>Giada De Laurentiis</author>  
    <year>2005</year>  
    <price>30.00</price>  
  </book>  
  <book category="children">  
    <title lang="en">Harry Potter</title>  
    <author>J. K. Rowling</author>  
    <year>2005</year>  
    <price>29.99</price>  
  </book>  
  <book category="web">  
    <title lang="en">Learning XML</title>  
    <author>Erik T. Ray</author>  
    <year>2003</year>  
    <price>39.95</price>  
  </book>  
</bookstore>
```

# XML szintaxis szabályok

Az XML **szintaxis szabályai** nagyon *egyszerűek* és *logikusak*.

Az XML dokumentumoknak tartalmazniuk kell  
egy **gyökér** elemet, amely az összes többi elem **szülője**.

A példában a **<note>** a gyökérelem:

```
<?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 szintaxis szabályok - The XML Prolog

Ezt a sort **XML Prolognak** hívják.

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

- Az XML prolog *opcionális*. Ha létezik, akkor elsőnek kell lennie a dokumentumban.
- Az XML dokumentumok *nemzetközi karaktereket tartalmazhatnak*, pl.: norvég øæå vagy francia èèé.
- A hibák elkerülése érdekében meg kell adnia a használt kódolást, vagy az XML fájlokat *UTF-8 fájlként* kell mentenie.

# XML szintaxis szabályok – nyitó és záró címke

*Minden XML elemnek rendelkeznie kell záró címkével.*

Minden elemnek rendelkeznie **kell** záró címkével:

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

- Az XML-címék megkülönböztetik a kis- és nagybetűket. A `<Title>` címke eltér a `<title>` címkétől.
- A nyitó és záró címéket ugyanazzal a betűvel kell írni:

```
<message>This is correct</message>
```

# XML szintaxis szabályok – nyitó és záró címke

Az XML *elemeket megfelelően be kell ágyazni.*

Az XML-ben minden elemnek **megfelelően be** kell ágyazódnia egymásba:

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

# XML szintaxis szabályok - attribútumértékek

*Az XML attribútumértékeket mindig idézőjelbe kell tenni.*

- Az XML elemeknek **név = érték** párokban lehetnek attribútumai, akárcsak a HTML-ben.
- XML-ben az attribútumértékeket mindig idézőjel kell tenni:

Példa:

```
<note date="8/09/2022">
    <to>Péter</to>
    <from>Jani</from>
</note>
```

# XML szintaxis szabályok - entitás hivatkozások

## *Entitás hivatkozások*

- Egyes karaktereknek *különleges jelentése* van az XML-ben.
- Ha egy olyan karaktert, mint "<", elhelyez egy XML elem belsejében, az hibát generál, mert az elemző *egy új elem kezdete*ként értelmezi.
- Pl.: <message>salary < 1000000</message>
- Cseréljük le a "<" karaktert egy **entitás hivatkozással** :  
`<message>salary &lt; 1000000</message>`

# XML szintaxis szabályok - entitás hivatkozás

Öt előre definiált *entitás hivatkozás* található az XML-ben:

&lt;	<	less than
&gt;	>	greater than
&amp;	&	ampersand
&apos;	'	apostrophe
&quot;	"	quotation mark

# XML szintaxis szabályok – megjegyzések

## *Megjegyzések XML-ben*

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

Két kötőjel a megjegyzés közepén *nem megengedett*:

```
<!-- This is an invalid -- comment -->
```

# XML szintaxis szabályok

## *Well Formed XML*

Azok a XML dokumentumok, amelyek *megfelelnek az említett szintaxis szabályoknak*, "**Well Formed**" XML dokumentumok.

## *Érvényes*

**Érvényes** XML dokumentumnak nevezzük azon *jól-formázott XML dokumentumokat*, amelyek *logikai felépítése és tartalma teljes mértékben megegyezik* az XML dokumentumban meghatározott szabályoknak.

*Pl.: Eclipse: Click Properties - Validation*

# XML elements

An XML document contains XML Elements.

Az XML *elem* az elem kezdőcímkéjétől (*beleértve*) az elem végcímkéjéig terjed.

```
<price>29.99</price>
```

Egy *element* a következőket tartalmazhatja:

- text,
- attributes,
- other elements,
- or a mix of the above.

# XML elements - példa

```
<bookstore>
  <book category="children">
    <title>Harry Potter</title>
    <author>J. K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="web">
    <title>Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

- A `<title>`, `<author>`, `<year>` és `<price>` **szöveges tartalommal** bír, mert szöveget tartalmaznak.
- A `<bookstore>` és a `<book>` **elemeknek tartalma** van, mert elemeket tartalmaznak.
- A `<book>` **attribútummal rendelkezik** (`category = "gyerekek"`).

# XML elements - Empty XML Elements

A *tartalom nélküli elem* - ez az üres elem.

Ezek általában *megjeleníti információkat tartalmaz*.

Az XML-ben egy ilyen üres elemet jelölhet meg:

```
<element></element>
```

Használhat ún. önzáró címkét is:

```
<element />
```

Az üres elemeknek lehetnek attribútumai!

Pi.: <kep forras="valami.jpg" szelesseg="356" magassag="356" />

# XML elements - Elnevezési szabályok

Az XML *elemeknek* a következő elnevezési szabályokat kell követniük:

- Az elemek neve *megkülönbözteti a kis- és nagybetűket*.
- Az elemek nevénak *betűvel vagy aláhúzással kell kezdődnie*.
- Az elemek neve *nem kezdőhet xml* (vagy XML, vagy Xml stb.)
- Az elemek neve *nem tartalmazhat szóközt*.

# XML elements - Az XML elemek kiterjeszthetők

Az XML elemek bővíthetők további információk továbbítására.

```
<note>
```

```
  <to>Péter</to>
```

```
  <from>Jani</from>
```

```
  <heading>Emlékeztető</heading>
```

```
  <body> Ne felejtsd el, amit megbeszéltünk</body>
```

```
</note>
```

# XML elements - Az XML elemek kiterjeszthetők

## *Kibővítés*

```
<note>  
  <date>2021-09-08</date>  
  <to>Péter</to>  
  <from>Jani</from>  
  <heading>Emlékeztető</heading>  
  <body> Ne felejtsd el amit megbeszéltünk</body>  
</note>
```

# XML Attributes

- Az XML elemeknek lehetnek *attribútumai*, akárcsak a HTML-nek.
- Az attribútumok egy *adott elemhez kapcsolódó adatokat tartalmaznak*.
- Az attribútumértékeket mindig *idézőjel közé kell tenni*. Lehet *aposztrófa* vagy *dupla idézőjel*.

Egy személy *neme* esetén a <személy> elem így írható:

```
<person gender="female"> vagy  
<person gender='female'>
```

# XML Attributes - példa

Példák: Ebben a példában a *neme attribútum*.

```
<person gender="female">
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>
```

Ebben pedig a *neme egy elem*.

```
<person>
  <gender>female</gender>
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>
```

# XML attribútumok a metaadatokhoz

Az elemekhez néha *azonosító hivatkozásokat* rendelnek.

Ezekkel az *azonosítóval* osítani az XML elemeket, mindenben.

```
<messages>
  <note id="501">
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
  </note>
  <note id="502">
    <to>Jani</to>
    <from>Tove</from>
    <heading>Re: Reminder</heading>
    <body>I will not</body>
  </note>
</messages>
```

# XML dokumentum

- Az XML dokumentum *szöveges állományban* tárolt, melynek szokásos kiterjesztése az 'xml'.
- Az XML-ben a *metaadatok megadására jelölő elemek* használatosak.
- Az elemek tagjait a '<' és '>' karakterek határolják.
  - *tartalom elem*: a *tartalom tulajdonságait írja le*, a *feldolgozó programnak* szóló információkat tartalmaz,

# XML dokumentum elemei

- *deklarációs elem*: a feldolgozónak szóló instrukciókat tartalmaz,
- *megjegyzés elem*: az olvasónak, a programozónak szóló információkat tartamaz.
- A kéttagú *tartalom elem* szerkeze:  
`<elemnév jellemzők_listája> tartalom</elemnév>`  
Az egytagú *tartalom elem* szerkeze:  
`<elemnév jellemzők_listája/>`

# XML dokumentum elemei

- A *tartalom elemek* neve tetszőleges szó lehet  
`<életkor> 23 </életkor>`  
Az XML elemnevek kisbetű/nagybetű érzékenyek.
- A deklarációs elem szerkezete:  
`<? elemnév jellemzők_listája ?>`
- A megjegyzés elem szerkezete:  
`<! -- megjegyzés -- >`
- Egy attribútum megadása egy párossal történik, a párosban a jellemző **nevét** és az **értékét** társítjuk össze – formája:  
`jellemző_neve = „érték”`

# Helyesen formált XML dokumentum - összefoglalás

## Helyesen formált XML dokumentum

### Megkötések a felépítésre

Egyetlen fő, gyökér jelölő elem létezik

A jelölő elemeknek két formátuma lehet:

kéttagú elem:

`<tagnév ...> tartalom </tagnév>`

egytagú elem: `<tagnév ... />`

Az elem neve egy tetszőleges szó lehet

Jelölő elemek egymásba ágyazhatók, de átlapolás nem lehet

feldolgozási utasítás elem alakja: `<?kezelő ... ?>`

megjegyzés elem alakja: `<!-- ... -->`

```
<?xml version="1.0" ?>
<könyvek>
  <könyv ikod="2">
    <cim> &X; </cim>
    <ev>2003</ev>
    <olvasoterem/>
  </könyv>
</könyvek>
<!-- ez egy minta --&gt;</pre>
```

# Helyesen formált XML dokumentum - összefoglalás

## Helyesen formált XML dokumentum

**Megkötések a felépítésre**

**Feldolgozási utasítás (PI):**  
az XML feldolgozó programnak szól

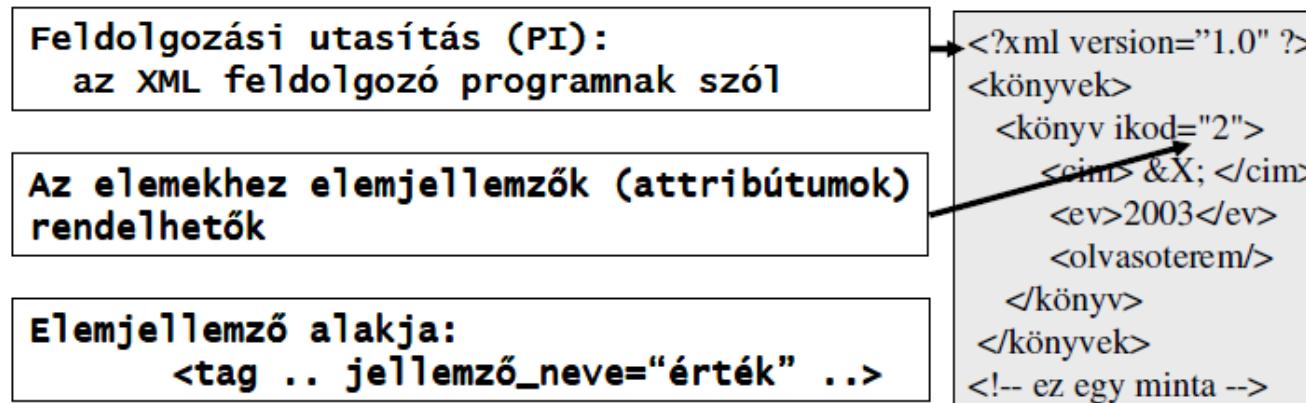
**Az elemekhez elemjellemzők (attribútumok)**  
rendelhetők

**Elemjellemző alakja:**  
`<tag .. jellemző_neve="érték" ..>`

**Egy elemhez több elemjellemző is rendelhető**

**Az elemjellemző értéke elemi**

**Az elemjellemző a elemhez szorosan kötődő értéket tárol**



```
<?xml version="1.0" ?>
<könyvek>
  <könyv ikod="2">
    <cim> &X; </cim>
    <ev>2003</ev>
    <olvasoterem/>
  </könyv>
</könyvek>
<!-- ez egy minta --&gt;</pre>

GEIAL Kovács László


```

# Helyesen formált XML

## Megszorítások:

- csak egyetlen gyökér elem,
- nyitó elemhez kell záró elem,
- lehet üres elem is,
- az elemnév egytagú szó,
- elemnév nem lehet foglalt szó, foglalt karakter,
- az elemek egymásba ágyazódnak,

```
<?xml version="1.0" ?>
<könyvek>
  <könyv ikod="2">
    <cim> &X; </cim>
    <ev>2003</ev>
    <olvasoterem/>
  </könyv>
</könyvek>
<!-- ez egy minta -->
```

# Helyesen formált XML

## Megszorítások (folyt.):

- a nyitó tagban vagy üres elemben lehet attribútum
- attribútum megadás fix alakú: név = “érték”
- megjegyzés: <!-- .... -->
- PI egység: <? .... ?>

```
<?xml version="1.0" ?>
<könyvek>
  <könyv ikod="2">
    <cím> &X; </cím>
    <év>2003</év>
    <olvasoterem/>
  </könyv>
</könyvek>
<!-- ez egy minta -->
```

# Helyesen formált XML jellemzői

- Tartalmaz induló *feldolgozó utasítást*.
- *Egy gyökérelemmel* rendelkezik.
- *Tartalmazhat üres elemeket*.
- *Egy szülőnek több gyereke* is lehet.
- *Egy gyereknek csak egy szülője* lehet.
- Az *elemek rendelkezhetnek attribútumokkal*.
- *Egy attribútum csak egyszer szerepelhet egy elemnél*.

# Speciális karakterek kezelése

## Speciális karakterek kezelése

Vannak foglalt karakterek, nem lehetnek normál szöveg részei

```
<?xml version="1.0" ?>
<könyvek>
  <könyv ikod="2">
    <cim> A&B</cim>
    <ev>2003</ev>
    <olvasoterem/>
  </könyv>
</könyvek>
<!-- ez egy minta --&gt;</pre>
```

Foglalt karakterek szerepelhetetése:  
helyettesítő szimbólumok (egyedek)

Előre definiált helyettesítő jelek:

<	: &lt;
>	: &gt;
&	: & amp;
'	: &apos;
"	: &quot;

Tetszőleges karakter: &#nnn;

Hosszabb egyedi karaktersorozat:  
<! [CDATA[ szöveg ]]>

# Névtér - Szabványok

# Mi az XML névtér

Egy *XML névtér* XML dokumentumokban *elemek* és *attribútumok neveként* használt nevek egy olyan halmaza, melyet egy *URI-hivatkozás azonosít*.

- Az *URI-hivatkozás* fogalmát lásd az alábbi szabványban:
  - Tim Berners-Lee, Roy T. Fielding, Larry Masinter, *Uniform Resource Identifier (URI): Generic Syntax*, STD 66, RFC 3986, January 2005.

<https://tools.ietf.org/html/rfc3986>

# Névtér - Szabványok

- *Namespaces in XML 1.0 (Third Edition)* (W3C ajánlás, 2009. december 8.)  
<https://www.w3.org/TR/xml-names/>
- *Namespaces in XML 1.1 (Second Edition)* (W3C ajánlás, 2006. augusztus 16.)  
<https://www.w3.org/TR/xml-names11/>

Az XML 1.1 ajánláshoz.

# XML névterek - Névütközések

Miért használunk névteret?

Az XML névterek módszert kínálnak az *elemnév ütközések elkerülésére*.

- Az XML-ben az *elemneveket a fejlesztő határozza meg.*

Ez gyakran konfliktushoz vezet, amikor különböző XML-alkalmazásokból származó XML-dokumentumokat próbál keverni.

# XML névterek – Névütközések - példa

Ez az XML tartalmazza a HTML-tábla adatait:

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

Ez az XML információ tartalmaz egy bútor:

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

Ha ezeket az XML-töredékeket összeadnák, névütközés állna fenn.  
Mindkettő tartalmaz egy `<table>` elemet, de az elemek tartalma és jelentése eltérő.

# XML névterek – Névütközés megoldása

A névütközés megoldása *előtag használatával*:

- Az XML névkonfliktusai könnyen elkerülhetők egy *név előtag használatával*.
- Ez az XML információt bútorról:

```
<h:table>
  <h:tr>
    <h:td>Apples</h:td>
    <h:td>Bananas</h:td>
  </h:tr>
</h:table>
```

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

```
<table>
  <name><f:table>
  <widtl><f:name>African Coffee Table</f:name>
  <lengt><f:width>80</f:width>
  </table>  <f:length>120</f:length>
            </f:table>
```

# XML névterek - Az xmlns attribútum

Az előtagok XML-ben történő használatakor meg kell határozni az előtag névterét.

- A névtér egy *elem kezdő* található **xmlns** attribút

```
<root>
<h:table xmlns:h="http://www.w3.org/TR/html4/">
  <h:tr>
    <h:td>Apples</h:td>
    <h:td>Bananas</h:td>
  </h:tr>
</h:table>
```

- A névtér *deklaráció* a kö

xmlr

```
<f:table xmlns:f="https://www.w3schools.com/furniture">
  <f:name>African Coffee Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>

</root>
```

Ikezik

# XML névterek - Az xmlns attribútum

- Az *első* <table> elem xmlns attribútuma a h:

*előtagnak minősített név*

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

- A *második* <table> elem *előtagnak minősített név*

```
<h:table>  
  <h:tr>  
    <h:td>Apples</h:td>  
    <h:td>Bananas</h:td>  
  </h:tr>  
</h:table>
```

- Ha egy elemhez meg van azonos előtaggal rendelkezni, ugyanahhoz a névtérhez

```
<f:table>  
  <f:name>African Coffee Table</f:name>  
  <f:width>80</f:width>  
  <f:length>120</f:length>  
</f:table>
```

# XML névterek

A névterek deklarálhatók az XML gyökérelemben is:

*Egységes erőforrás-azonosító (URI)*

- Az **egységes erőforrás-azonosító (URI)** egy karakterlánc, amely *azonosítja az internetes erőforrást*.
- A leggyakoribb URI az **Uniform Resource Locator (URL)**, amely *azonosítja az internetes tartomány címét*.
- Az URI másik, nem annyira elterjedt típusa az **Egységes erőforrásnév (URN)**.

# XML névterek - Alapértelmezett névterek

A következő szintaxiszal rendelkeznek:

`xmlns="namespaceURI"`

Ez az XML tartalmazza a HTML-tábla adatait:

Ez az XML információt tartalmaz egy bútorról:

```
<table xmlns="http://www.w3.org/TR/html4/">
  <tr>
    <td>Apples</td>
    <td>Bananas</td>
  </tr>
</table>
```

```
<table xmlns="https://www.w3schools.com/furniture">
  <name>African Coffee Table</name>
  <width>80</width>
  <length>120</length>
</table>
```

# Névterek szerepe - összefoglalás

A névtér szerepe az *elemnevek/attribútumnevek* egyértelműsítése:

- Az *elemnév* önmagában *nem elegendő* egyértelmű azonosításra.
- *Megoldás*: a név kibővítése a feldolgozó azonosításával.
- *Névtér*: a feldolgozó program azonosítója.

```
<cim>  
  Budapest Nagy u 23  
</cim>
```

```
<cim>  
  Egri csillagok  
</cim>
```

```
http://a.b.hu/konyvek  
  books
```

# Névterek szerepe - összefoglalás

- Névtér szokásos alakja: URI
- Egy dokumentumban több névtér is élhet.

A névterek elemhez kötése *alias nevekkel* történik.

A különböző XML feldolgozókhöz más és más saját névtér tartozik.

<a href="http://www.w3.org/2000/10/XMLSchema">http://www.w3.org/2000/10/XMLSchema</a>	: W3C XML Schema névtere
<a href="http://www.w3.org/1999/XSL/Transform">http://www.w3.org/1999/XSL/Transform</a>	: XSL feldolgozó névtere
<a href="http://www.w3.org/1999/xlink">http://www.w3.org/1999/xlink</a>	: XLink névtere

# Névterek használata

## Névterek használata

### Névtér használat lépései

**Névtér és alias összekötése:**

```
<tagnév ... xmlns:alias="URI" ... >
```

**Névtér kötése elemhez:**

```
<alias:tagnév ...> ... </alias:tagnév>
```

**Névtér kötése elemjellemzőhöz:**

```
<tagnév ... alias:jellemző="érték" ...>
```

```
<books:cim xmlns:books="http://a.b.hu/konyvek">  
    Egrí csillagok  
</books:cim>
```

<a href="http://www.w3.org/2000/10/XMLSchema">http://www.w3.org/2000/10/XMLSchema</a>	: W3C XMLSchema névtere
<a href="http://www.w3.org/1999/XSL/Transform">http://www.w3.org/1999/XSL/Transform</a>	: XSL feldolgozó névtere
<a href="http://www.w3.org/1999/xlink">http://www.w3.org/1999/xlink</a>	: XLink névtere

# Névterek használata

## Névterek használata

**Van alapértelmezési névtér is, ahol az alias üres**

**Névtér és alias összekötése:**

```
<tagnév ... xmlns="URI" ... >
```

**Névtér kötése elemhez:**

```
<tagnév ...> ... </tagnév>
```

```
<tagnév ... jellemző="érték" ...>
```

**Az aliasnév érvényességi köre azon elemre és annak befoglalt elemeire terjed ki, amelyben definiálták.**

**Egy belső elemben a kívül értelmezett aliasnév felülírható egy új értelmezéssel**

```
<cim xmlns="http://a.b.hu/konyvek">  
Egri csillagok  
</cim>
```

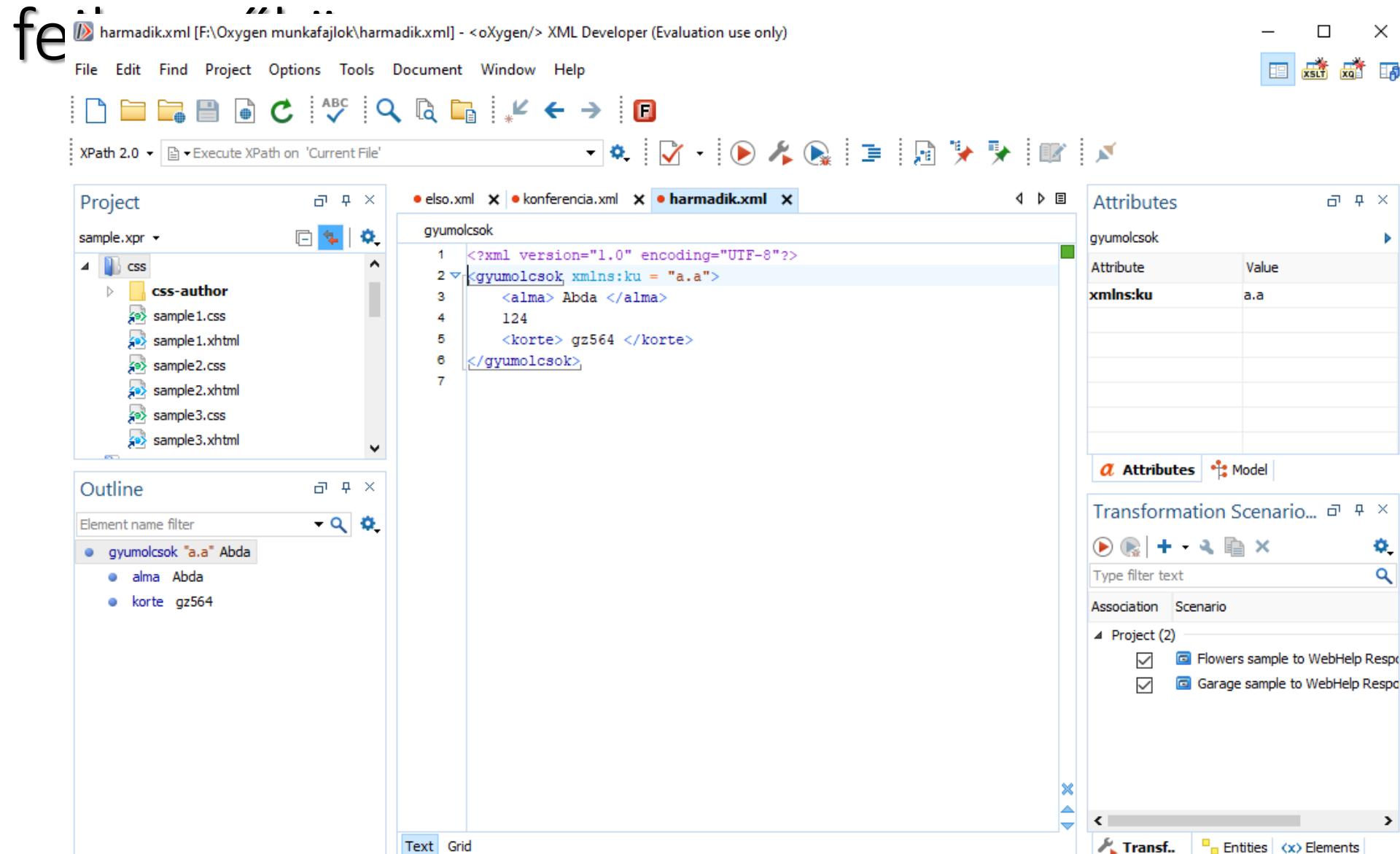
# XML dokumentumok létrehozása - fejlesztőkörnyezet

- Az XML dokumentumot *tetszőleges szövegszerkesztővel* elkészthetjük, amely alkalmas szövegállományok előállítására (Word, Notepad++..)
- A létrejött XML dokumentummal kapcsolatban az egyik legelső teendőnk a *dokumentum helyesen formáltságának ellenőrzése*.
- a dokumentum helyesen formált, ha teljesíti a szerkezet leíró alfejezetben felsorolt tulajdonságokat.

# XML dokumentumok létrehozása - fejlesztőkörnyezet

- A legtöbb XML megjelentő program elvégzi az ellenőrzést (IE).
- Az IE mellett léteznek speciális XML-szerkesztők is, ilyen az **Oxygen XML Editor**.
- A termék honlapja:  
<http://www.oxygenxml.com>
- A honlapról egy ingyenes ideiglenes tesztrendszer is letölthető, amelyhez egy 30 napos próbaidő tartozik.

# XML dokumentumok létrehozása -



# XML dokumentumok létrehozása- fejlesztőkörnyezet - Eclipse

Gyakran használt fejlesztőkörnyezet az Eclipse.

Különféle platformokra letölthető:

- MacOS [x86\\_64](#)
- Windows [x86\\_64](#)
- Linux [x86\\_64](#) | [AArch64](#)

URL: <https://www.eclipse.org/downloads/>

Mind a kettőt használhatjuk.

# XML dokumentumok létrehozása- fejlesztőkörnyezet - Eclipse

Csomagból történő letöltés: *Eclipse IDE 2022-06 R Packages*

[Eclipse IDE for Java Developers](#), amely szintén különféle platformokra tölthető le.

- Windows [x86 64](#)
- MacOS [x86 64](#)
- Linux [x86 64](#) | [AArch64](#)

Minden Java fejlesztő számára nélkülözhetetlen eszközök, beleértve a *Java IDE*-t, a *Git client*, az *XML-Editor*, a *Maven* és a *Gradle* integrációt.

# XML létrehozása - összefoglalás

## XML létrehozása

### A létrehozás főbb lépései

**Adattartalom meghatározása**

**Adatok tagolása**

**Adatokhoz név, jelentés hozzárendelése**

**Adatok hierarchiájának meghatározása**

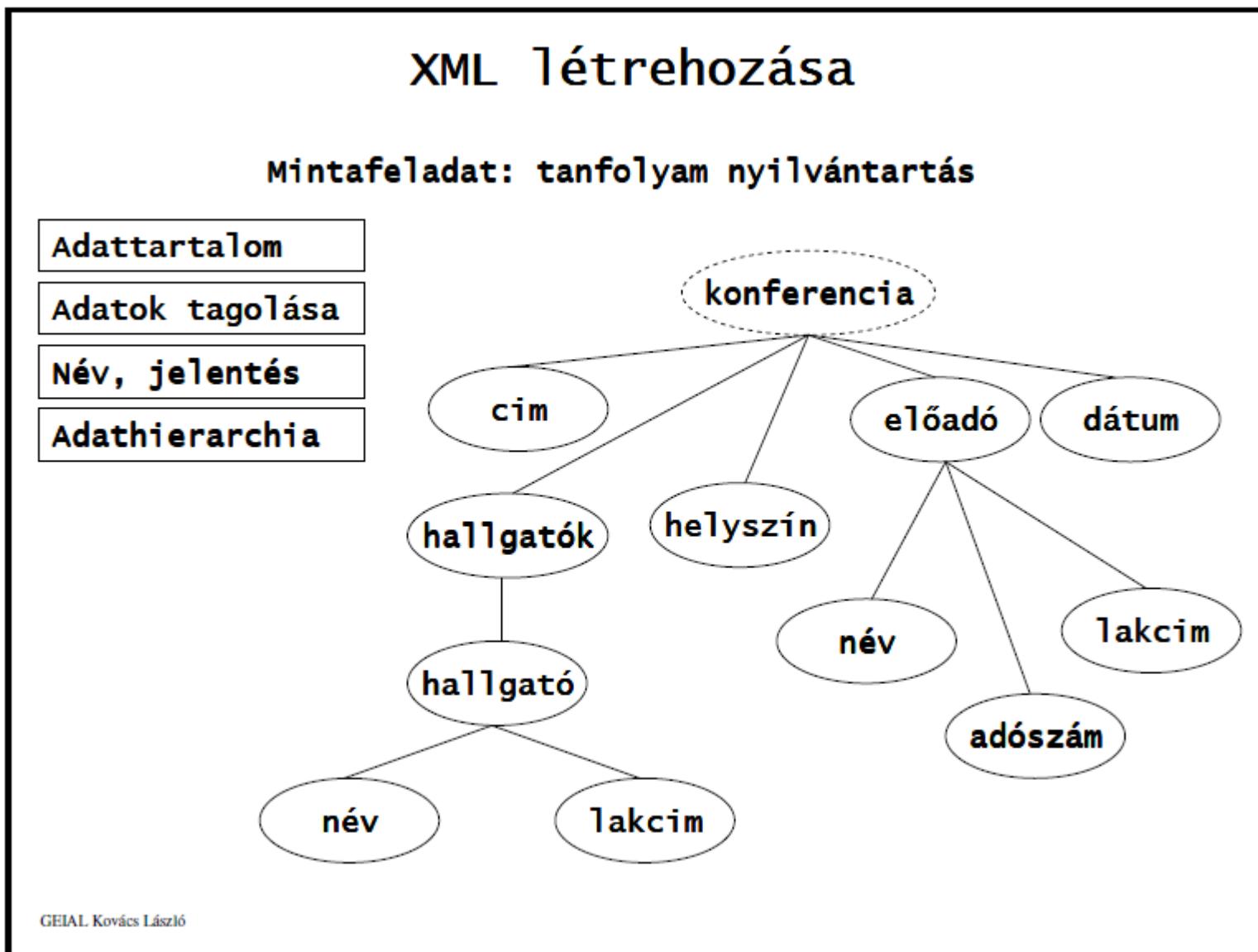
**Gyökérelem létrehozása**

**Adatelem hozzárendelése jellemzőhöz vagy elemhez**

**Prolog PI megadása**

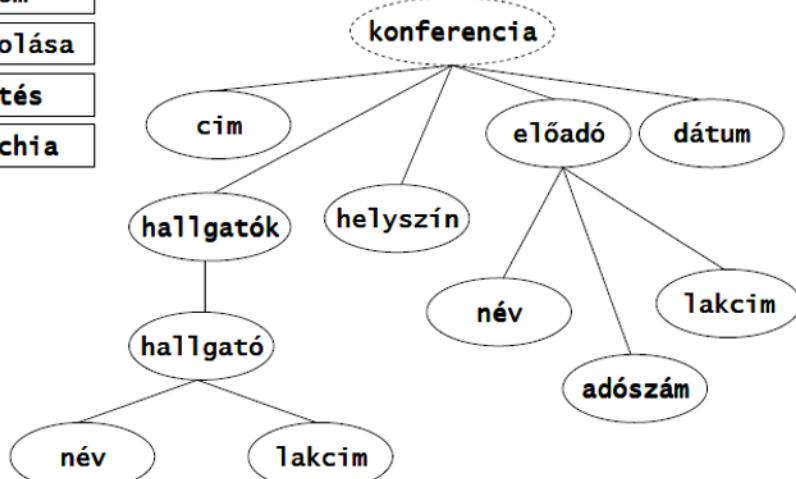
**Elemhierarchia megadása**

# XML létrehozása - mintapélda



# XML létrehozása - mintapélda

Adattartalom  
Adatok tagolása  
Név, jelentés  
Adathierarchia



## XML létrehozása

Mintafeladat: tanfolyam nyilvántartás

Gyökérelem  
Jellemzők, elemek  
Prolog PI  
Elemhierarchia

```
<?xml version="1.0"?>
<konferencia>
<cím> XML alapok </cím>
<helyszin> XXVI ea. </helyszin>
<oktato adsz="256154">
    <nev> Komor Tibor </nev>
    <lakcím> Zoro ut 45 </lakcím>
</oktato>
<hallgatok>
    <hallgato>
        <nev> Okosodo Olga</nev>
        <lakcím> Kun u 6 </lakcím>
    </hallgato>
</hallgatok>
</konferencia>
```

# XDM modell

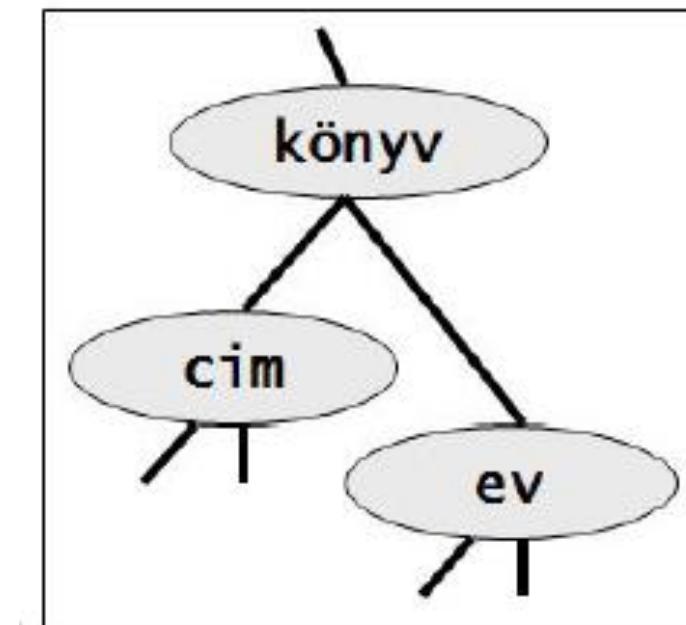
Az XDM (XML Data Modell) a hierarchia leírására szolgál, az XML dokumentumot fa alakban írja le.

**Építőkövek:**

- elem (ellipszis)
- szöveg (téglalap)
- attribútum (rombusz)

Az XDM alapvetően *előfordulást ír le* és nem sémát.

```
<?xml version="1.0" ?>
<könyvek>
  <könyv ikod="2">
    <cím> &X; </cím>
    <ev>2003</ev>
    ISBN1234
  </könyv>
</könyvek>
```



# XDM modell séma elemei

Az XDM kiegészítése séma elemekkel:

- Többszörös előfordulás (dupla ellipszis),
- Opcionális elem (ellipszis szaggatott vonallal),
- Kötelező elem (ellipszis vastag vonallal),
- Elsődleges kulcs (attribútum neve aláhúzva),
- Idegen kulcs (attribútum neve aláhúzva szaggatott vonallal).

A feldolgozó programok nagy része az XDM modellre épül.

# ER konverziója XDM modellre

Áttérés ER modellről XDM-re	
ER (Entity-Relation)	XDM (XML Data Model)
Egyed	Elem
Tulajdonság	Gyerekelem
Kulcs tulajdonság	Attribútum
Lényegi egyértékű tulajdonság	Attribútum
1:N kapcsolat	PCR
N:M kapcsolat	2 PCR vagy kapcsoló érték
Séma	Hierarchia

# ER modell konverziója XMLSchema-ra

## Konverziós szabályok

**egyed** ⇒ **elem**

**elemi tulajdonság** ⇒ **szöveg elem**

**kulcs tulajdonság** ⇒ **elemjellemző + kulcs megkötés**

**összetett tulajdonság** ⇒ **elemeket tartalmazó gyerekelem**

**többértékű tulajdonság** ⇒ **gyerekelem, ismétlődéssel**

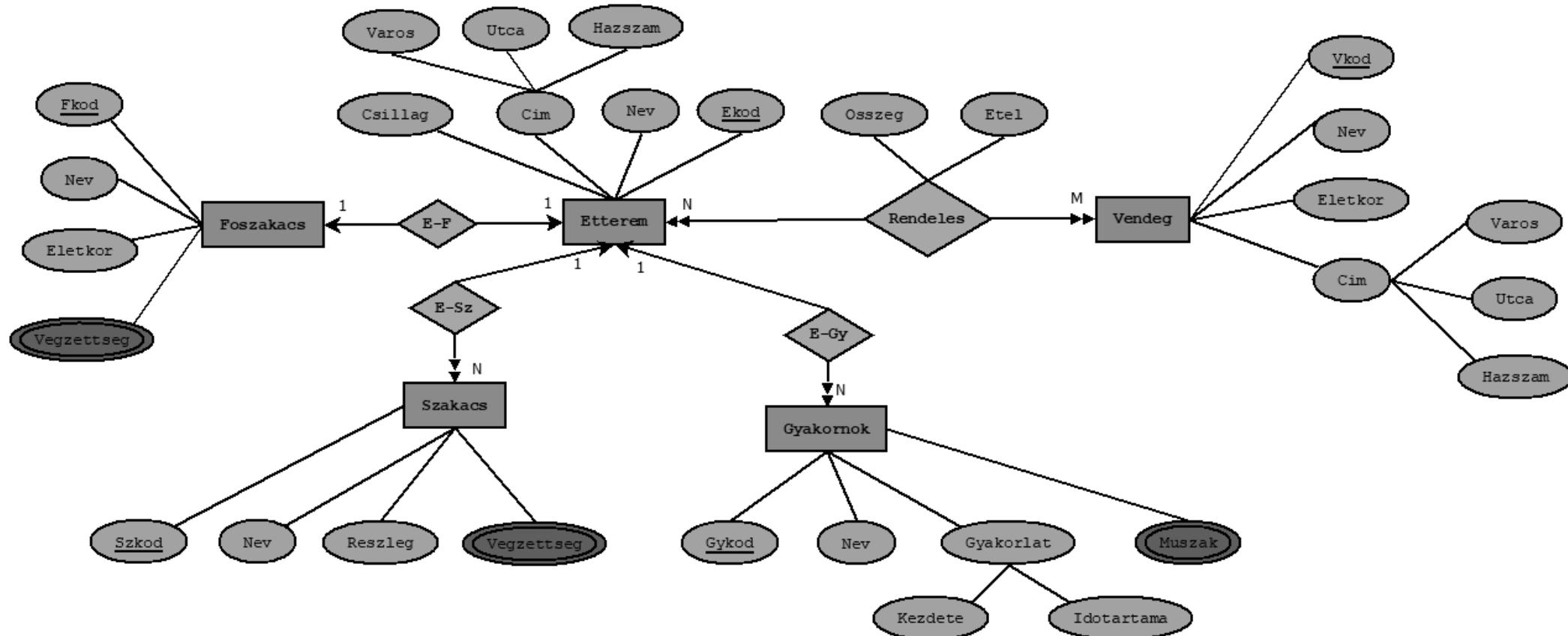
**kapcsoló tulajdonság** ⇒ **elemjellemző + idegen kulcs megkötés**

**1:N kapcsolat** ⇒ **elemjellemző + kulcs + idegen kulcs megkötés**

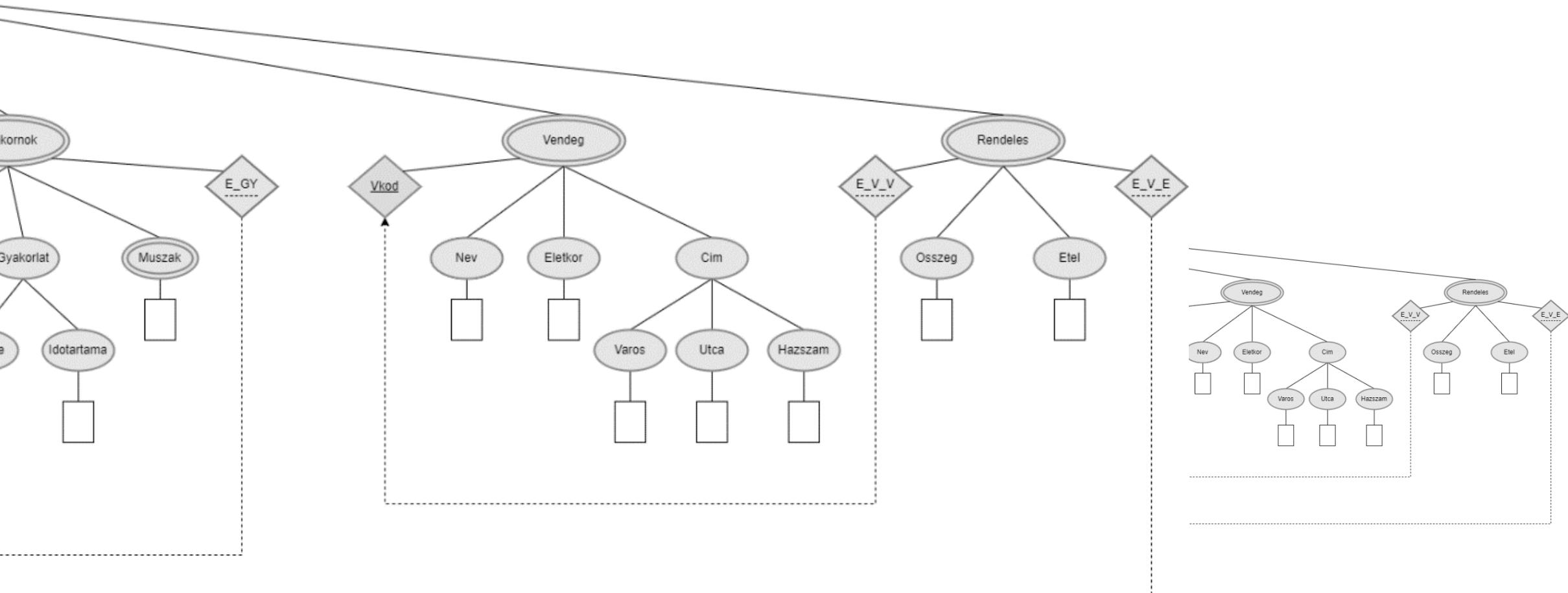
**N:M kapcsolat** ⇒ **külön kapcsoló elem és idegen kulcsok minden két oldalra**

# ER konverziója XDM modellre - mintapélda

## AZ ADATBÁZIS ER MODELL



# ER konverziója XDM modellre - mintapélda



# Irodalom

Kovács László: Adatkezelés XML környezetbe, ME

Tantárgyi honlap: [moodle.iit.uni-miskolc.hu](http://moodle.iit.uni-miskolc.hu)