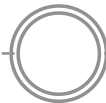




A nyelvtechnológia alapjai



2. Gyakorlat

Dokumentumformátumok,
jelölőnyelvek, XML



- Sima szöveges formátum: TXT
 - tömör, esetleges szerkezet
 - nagyon hatékony feldolgozó eszközök (pl. Linux környezetben)
 - a sorvége kódolása: CR (carriage return, \r) és LF (line feed, \n); win (CR LF), unix (LF), mac (CR)
 - karakterkódolás
- Lapleíró nyelvek: PS (PostScript), PDF (Portable Document Format)
 - cél a szép és gyors nyomtatás
 - nem (nagyon nehezen) módosítható
 - a megjelenést alacsony szinten írja le



- Szövegszerkesztő formátumok: DOC / RTF
 - cél: szép megjelenésű dokumentumok előállítása
 - módosítható tartalom
 - könnyen módosítható forma (WYSIWYG)
 - elsikkad a szerkezet
 - docx, odt: XML-alapú
- LaTeX
 - cél: szép megjelenésű dokumentumok előállítása
 - szöveges formátum, a tartalom között kódolva jelenik meg a szerkezet és a forma
 - dokumentumsablonok – újat elég nehéz létrehozni
 - PS, PDF lesz belőle



- Történet
 - 60-as évek: válasszuk el a dokumentum tartalmát és szerkezetét a megjelenítéstől! →
 - 1969: **GML** (Generalized Markup Language)
 - Goldfarb, Mosher, Lorie
 - 1986: ISO szabvány lesz: **SGML** (Standard GML)
 - 1989: **HTML** (Hypertext ML) – egy SGML alkalmazás, amit mindenki ismer
 - 1998: **XML** (Extensible ML) 1.0 verziója



- **kódolás (markup):**
 - a szöveg egy interpretációjának reprezentálására használt módszer; minden, ami a dokumentumban nem tartalom; címkézés
- **jelölőnyelv (markup language):**
 - a szövegek kódolására használt jelölési szabályok összessége
- **címke (tag, element):**
 - adott szövegrész megjelölésére szolgáló eszköz, a szövegrészt nyitó és záró címke közé zárjuk; a címkék szigorúan elkülönülnek a dokumentum szövegétől
- Példa:
 - **LaTeX:** Ez most `\emph{nagyon}` fontos.
 - **HTML:** Ez most `<i>nagyon</i>` fontos.



- Procedurális és deskriptív markup
 - **procedurális markup**: mit kell vele csinálni
 - példa: `<b>Kovács János</b>`
 - **deskriptív markup**: mit jelent
 - példa: `<author>Kovács János</author>`
- Struktúráltság
 - a címkék egymásba ágyazhatók → ábrázolható a dokumentum struktúrája (fa-struktúra)
 - a címkék nem fedhetik át egymást
 - példa:
 - jó: `<dátum><év>2004</év><hó>2</hó><nap>17</nap></dátum>`
 - rossz: `<i>dőlt <b>dőlt és félkövér</i> félkövér</b>`

- olyan szabályrendszer, mely jelölőnyelvek általános szabályszerűségeit definiálja
- azt, hogy hogyan kell megadnunk . . .
 - egy jelölőnyelv címkekészletét, és
 - a címkek egymásba ágyazhatóságának szabályait, azaz a dokumentum megengedett struktúráját
- segítségével céljainknak megfelelő strukturált jelölőnyelvek készíthetők

metanyelv	konkrét nyelv
SGML	pl. HTML
XML	pl. XHTML



- **HTML:**

- konkrét jelölőnyelv – címkekészlete előre definiált:
- `<html>` `<head>` `<body>` `<title>` `<p>` ...
- cél a webes *megjelenítés* – inkább procedurális
- egyszerű, olcsó, sok eszköz, a hivatkozásokat jól kezeli
- rosszul kereshető, mert hiányzik a szerkezet



- Követelmények:
 - rugalmas, bővíthető
 - alkalmazások széles körét támogassa
 - egyszerűen használható
 - könnyen elemezhető, feldolgozható, kereshető
 - ember által olvasható, világos szerkezetű dokumentumok
- Jellemzői:
 - metanyelv: szabadon definiálható címkekészlet és dokumentumstruktúra
 - az SGML-hez képest kötöttebb, egyszerűbb, pl. SGML-ben nem kötelező lezárni bizonyos tag-eket.
 - könnyen elemezhető: *általános* (minden XML fájlt feldolgozó!) elemzők készíthetők



- XML dokumentum felépítése
 - *címke (tag, element)*:
 - `<s>`Ez egy mondat.`</s>`
 - mindig van záró tag, üres címke jele: `<br/>`
 - beágyazás lehetséges, átfedés nem
 - *attribútumok (attribute)*:
 - `<s id="1">`Ez az "1" azonosítójú mondat.`</s>`
 - attribútum érték mindig idézőjelek között
 - a fájl elején: *prológ*
 - `<?xml version="1.0" encoding="ISO-8859-2"?>`
 - majd pontosan egy gyökérelem, ami tartalmazza az egész dokumentumot



- Megfelelőség
 - **Jól formázottság**
 - Egy *jól formázott (well-formed)* XML dokumentum megfelel a szabványban rögzített összes szintaktikai szabálynak, de nem köteles megfelelni a dokumentum típus deklarációnak.
 - **Érvényesség**
 - Egy *érvényes (valid)* XML dokumentum jól formázott és ezen kívül a dokumentum típus deklarációban meghatározott struktúra követelményeknek is megfelel.



- **DTD**

- DTD (Document Type Declaration): XML szerkezetének ellenőrzésére szolgál, a dokumentum típus deklarációi; meghatározott struktúra követelmények

- **XML Schema**

- XML sémanyelv, XSD
- segítségével felírhatóak olyan szabályrendszerek, melynek meg kell hogy feleljen egy XML dokumentum ahhoz, hogy *valid* legyen az adott sémában.
- XDR (Microsoft XML Schema Reduced) sémából alakult
- névterek, adattípusok



- DTD fájl elemei
 - *címke*: `<!ELEMENT név (tartalom-modell)>`
 - tartalom modell: elemek, szöveg (`#PCDATA`), `EMPTY`, `ANY`
 - tartalom modell operátorok: `*` `+` `?` `,` `|`
 - példa:
 - `<!ELEMENT név (vezeteknev , keresztnev+)>`
 - *attribútum*: `<!ATTLIST címke név típus default>`
 - típus: `CDATA`, `ID`, felsorolás . . .
 - default: `#REQUIRED`, `#IMPLIED`, adott érték



- DTD hátrányai
 - Nem xml formátumú
 - Nem ismeri a névtereket
 - Nem bővíthető
 - Csak egy teljes dokumentumra alkalmazható
 - Nem ismeri az adattípusokat

- Dokumentum struktúráját, a benne található adok típusát és értelmezési tartományait definiálja.
 - Definiálja az elemek egymásba ágyazását
 - Definiálja a gyermekelemek sorrendjét
 - Definiálja a gyermekelemek számát
 - Definiálja, hogy egy elem üres, szöveget, vagy további gyermekelemeket tartalmaz
 - Definiálja az elemek és attribútumok adattípusát
 - Alapértelmezett értéket definiál elemekhez és attribútumokhoz



- XML Schema Definition – XSD
 - XML, SQL, .NET kompatibilis
 - névterek, adattípusok
 - öröklődés, csoportosítás
- Felhasználás
 - XML dokumentum formális leírása
 - Forráskód generálás

- Végigmegyünk az összes elemen és definiáljuk őket.
- Szerkezet:
 - `<?xml version="1.0" encoding="UTF-8"?>`
 - `<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">`
 - `<xsd:element name="elem">`
 - `<xsd:complexType>`
 - `<xsd:sequence>`
 - `<xsd:attribute name="attrb" type="xsd:string"/>`



- Adattípusok
 - string, token, date, boolean, int, positiveInteger stb.
- Korlátozások
 - enumeration, length, maxLength, pattern stb.
- Elemek
 - element, group, list, sequence, complexType, attribute, unique stb.



- `complexType`
 - elements and attributes
- `simpleType`
 - no elements or attributes
- `complexContent`
 - extensions or restrictions on a complex type
- `simpleContent`
 - extensions or restrictions on a text-only complex type

- XML Tutorial

<http://www.w3schools.com/XML/default.asp>

- DTD Tutorial

<http://www.w3schools.com/dtd/default.asp>

- XML Schema

https://www.w3schools.com/xml/schema_intro.asp



Köszönöm a figyelmet



ध्यान देने के लिए आपका धन्यवाद
आप अपना ध्यान के लिए धन्यवाद
थैंक यू फॉर योर अटेंशन

Thank you for your attention
Thank you for your attention
Thank you for the attention

Merci de votre attention
Je vous remercie pour votre attention
Merci pour l'attention

感谢您的关注
谢谢你的关注
谢谢您的注意