

ELSŐ MELLÉKLET

XML ALAPISMERETEK

Az XML-fájlok előállításához, módosításához használhatunk egyszerűbb szövegszerkesztőt, mint például a Notepad++, de érdemes inkább olyan eszközt választani, amely a szintaktikai kiemelésen túl további szolgáltatásokat is nyújt (kódkiegészítés, érvényesség-ellenőrzés, struktúraspecifikus keresés/csere, kifejezések végrehajtása stb.). A kereskedelmi szoftverek közül az egyik legjobb az Oxygen XML Editor (30 napos licenc kérhető a kipróbáláshoz, kb. 950 \$), illetve az Altova XMLSpy (kb. 450 €), de számos ingyenes fejlesztőeszközt is alkalmas a feladatra, pl. Visual Studio (ha legalább egy programozási nyelv telepítve van), Visual Studio Code (Red Hat XML bővítménnyel), Eclipse. A kisebb XML-szerkesztők többnyire használhatatlanok. Biztosan nem jó például a Microsoft XML Notepad.

A megfelelő program kiválasztásnál szempont lehet még a helyesírás-ellenőrzés. A Hunspell ellenőrzőmotorral rendelkező programokhoz (mint az Oxygen) a szótárak ingyen tölthetők le, mivel ezek többnyire az OpenOffice-hoz készültek.

XML

Az XML az eXtensible Markup Language (bővíthető jelölőnyelv) rövidítése, amelynek segítségével az adatokat strukturáltan tárolhatjuk szövegfájlokban, így azok az emberek és a programok számára egyaránt értelmezhetők.

Az XML-dokumentumok deklarációkat, elemeket, feldolgozási utasításokat és megjegyzéseket tartalmazhatnak. Ezeknek a komponenseknek egy része kötelező, és vannak, amelyek opcionálisak. A dokumentum szerkezetére tekintsük az alábbi egyszerű mintát:

```
<?xml version="1.0" encoding="UTF-8"?>
<könyv kiadás="magyar">
  <szerző>Ken Follett</szerző>
  <cím>A tűzoszlop</cím>
  <ISBN>978 963 406 506 7</ISBN>
</könyv>
```

Az első sor az ún. XML-deklaráció, amely a jelölőnyelv verzióját és a fájl karakterkódolását specifikálja. Ezt követi a dokumentumelem (más néven gyökérelem), amelyből az XML-dokumentumokban csak egy lehet. Az elemeket nyitó- és zárócímkékkel adjuk meg, tartalmuk lehet szöveg, további elemek, illetve ezek keveréke. A mintában látható `cím` egy szöveges elem. Az elemek nyitórésze jellemzőket is tartalmazhat `név="érték"` szintaxissal.

MEGJEGYZÉS Az üres elemeket rövidített formában is megadhatjuk, például `<cím/>`.

Az XML-dokumentum elemei tetszőleges mélységig egymásba ágyazhatók, szerkezetük egy fastruktúrát alkot.

EGYEDEK (ENTITÁSOK)

Az XML-dokumentumokban néhány karakter speciális jelentéssel bír, például a < a címkék kezdetét jelöli, ezért a szövegben ezek helyett egyedhivatkozásokat kell használni. Az előre definiált egyedhivatkozások az alábbiak:

<	<	kisebb mint
>	>	nagyobb mint
&	&	et jel
'	'	aposztróf
"	"	idézőjel

A karakteregyedek számkóddal is megadhatók, például az Ω hexadecimálisan Ω, decimálisan Ω.

JÓLFORMÁZOTTSÁG

A programok az XML-fájlokat csak akkor tudják feldolgozni, ha megfelelnek az alábbi jólformázottsági követelményeknek:

1. A prológnak, amely az XML-deklarációt és a dokumentumelem típusspecifikációját tartalmazhatja, a dokumentum elején kell szerepelnie. Ezek megadása opcionális.
2. A dokumentumnak pontosan egy dokumentumelemet kell tartalmaznia.
3. Minden elemnek rendelkeznie kell zárócímkével.
4. Az XML-címkék megkülönböztetik a kis- és nagybetűket. A nyitó- és zárócímkéknek betűhelyesen egyezniük kell.
5. Az elemek nyitó- és zárócímkéjének ugyanazon elembe kell elhelyezkedniük. Például hibás az <f><d>szöveg</f></d>, mert a d elem zárórésze nem az f elembe van.
6. A jellemzőket az elem nyitócímkéjében kell megadni. Egy jellemző névből és értékből áll, ahol a név megelőzi az értéket, és a kettőt egy egyenlőségjel választja el egymástól. Két jellemző neve nem lehet azonos, az értéket idézőjelekben adjuk meg.
7. Az elemek belsejében és a jellemzőértékekben a különleges jelentésű karakterek helyett – mint pl. a <, >, &, " és a ' – egyedhivatkozásokat kell használni.
8. A megjegyzések szintaxisa a következő: <!-- szöveg -->. A megjegyzésen belül két kötőjel megadása nem megengedett.

NÉVTEREK

Az elemeket névterekbe csoportosítva különíthetjük el egymástól. A névterek minősített elemneveket használnak (pl. `xml:space`), azaz a név előtt előtagként a névtér neve is szerepel, ezáltal két azonos nevű, de különböző előtaggal rendelkező elem megkülönböztethetővé válik. A névtér deklarálásának szintaxisa `xmlns:névtér_név="URI"`, például

```
xmlns:xml="http://www.w3.org/XML/1998/namespace"
```

A névterek használata nemcsak a névkonfliktusok elkerülése végett hasznos, arra is lehetőséget nyújt, hogy az XML-feldolgozó az eltérő feldolgozási célokra szánt elemeket egymástól megkülönböztethesse.

ÉRVÉNYESSÉG-ELLENŐRZÉS

Séma megadásával előírhatjuk az XML-dokumentumban használható jelölőelemeket, azok struktúráját, illetve az értékekre vonatkozó korlátozásokat. A professzionális XML-szerkesztők a séma alapján ellenőrizni tudják a dokumentum érvényességét, valamint a kódkiegészítő funkciójuk révén környezettől függően ajánlják fel az adott helyre beszúrható elemek és attribútumok nevét, illetve azok értékét.

XPATH

Az XPath nyelv segítségével kiválaszthatjuk az XML-dokumentum egyes részeit, vagy egyszerűbb műveleteket végezhetünk azokon (például számlálás, összegzés). Az XPath a dokumentum fastruktúrájában hét ún. csomóponttípust különböztet meg:

1. gyökér (root)
2. elem (element)
3. jellemző (attribute)
4. szöveg (text)
5. megjegyzés (comment)
6. feldolgozási utasítás (processing instruction)
7. névtér (namespace)

Az XPath fának egyetlen gyökércsomópontja van, ez tartalmazza a fa összes többi csomópontját (például a dokumentumelemet). A szülőcsomópont magába foglalja a gyerekcsomópontokat. A jellemző- és a névtércsomópontok információt tartalmaznak a szülőcsomópontról.

Minden csomópont rendelkezik sztring reprezentációval (helyi név), illetve egyes csomópontoknak további jellemzője az ún. kiterjesztett név, amely a névtér nevéből és a helyi névből áll. (Ezek az egyes csomóponttípusok esetén eltérő módon képződnek.)

Azt, hogy miként juthatunk el az egyik csomópontból egy másikba az XPath-fán, elérési út segítségével adhatjuk meg. Ez a kifejezés elérési lépésekből áll, melyeket / karakter választ el egymástól, kiértékelésük balról jobbra történik. Az elérési lépés három összetevőt tartalmazhat: irány, csomópontvizsgálat és predikátum.

Az alábbi példa két lépésből áll. Az első kiválasztja a `feladatlap` nevű gyerekelemeket, a második ezek utolsó `feladat` elemét. A lépésekben az irányt `::` választja el a csomóponttól, a predikátum `[]`-ben található.

```
child::feladatlap/child::feladat[last()]
```

Az *irány* azt jelöli ki, hogy a környezeti csomóponthoz képest melyik csomópontokat kell bevonni a keresésbe. Az irányokat az alábbi táblázat foglalja össze. Ezek egy része előre, míg másik részük hátra felé mutató keresést ír elő meghatározva a csomópontok sorrendjét.

Irány	Leírás
self	Maga a környezeti csomópont.
parent	A szülőcsomópont, ha létezik.
child	Gyerekcsomópontok, ha léteznek.
ancestor	Ősök, ha léteznek.
ancestor-or-self	A környezeti csomópont és ősei
descendant	Leszármazottak, ha léteznek.
descendant-or-self	A környezeti csomópont és leszármazottjai.
following	A környezeti csomópontot követő csomópontok a leszármazottakat nem beleértve.
following-sibling	A környezeti csomópontot követő testvércsomópontok.
preceding	A környezeti csomópontot megelőző csomópontok az őseket nem beleértve.
preceding-sibling	A környezeti csomópontot megelőző testvércsomópontok.
attribute	Jellemző csomópontok.
namespace	Névtér csomópontok.

Minden iránynak van elvi csomóponttípusa, amely megadja, hogy milyen csomóponttípusok legyenek kiválasztva. Jellemző, illetve névtér irány esetén ez jellemző, illetve névtér típus, minden más esetben elem.

A *csomópontvizsgálat* az irány által meghatározott csomóponthalmazt szűkíti.

Csomópontvizsgálat	Leírás
*	Az elvi típussal megegyező csomópontok.
<code>node()</code>	Típustól függetlenül az összes csomópont.
<code>text()</code>	A szöveg típusú csomópontok.
<code>comment()</code>	A megjegyzés típusú csomópontok.
<code>processing-instruction()</code>	A feldolgozási utasítás típusú csomópontok.
<i>csomópontnév</i>	Az adott nevű csomópontok.

Az elérési út néhány esetben egyszerűsített szintaxissal is megadható.

Normál alak	Rövid alak	Leírás
<code>child::feladat</code>	<code>feladat</code>	A környezeti csomópont <code>feladat</code> nevű gyerekelemei.
<code>attribute::pont</code>	<code>@pont</code>	A környezeti csomópont <code>pont</code> nevű jellemzője.
<code>self::node()</code>	<code>.</code>	A környezeti csomópont.
<code>parent::node()</code>	<code>..</code>	A szülőcsomópont.
<code>/descendant-or-self::node()</code>	<code>//</code>	Az összes csomópont a dokumentum gyökerétől indulva.

A *predikátum* tovább szűkítheti az eredményhalmazt, vagyis az irány és csomópontvizsgálat által meghatározott csomópontokból csak azok lesznek kiválasztva, amelyekre a feltételes kifejezés igaz értéket ad. A kifejezésekben literálok, csomópontok, változóreferenciák és függvényhívások szerepelhetnek operátorokkal összekapcsolva.

MEGJEGYZÉS A függvényekről a https://www.w3schools.com/xml/xsl_functions.asp, az operátorokról a https://www.w3schools.com/xml/xpath_operators.asp oldalon talál részletes leírást.