

Sikos László:

XHTML

A HTML megújulása XML alapokon

Sikos László:

XHTML

A HTML megújulása XML alapokon

BBS-INFO Kiadó – 2004.

Minden jog fenntartva! A könyv vagy annak oldalainak másolása, sokszorosítása csak a kiadó írásbeli hozzájárulásával történhet.

A könyv nagyobb mennyiségben megrendelhető a kiadónál:
BBS-INFO Kft. 1630 Bp. Pf. 21. Tel.: 407-17-07

A könyv megírásakor a szerző és a kiadó a lehető legnagyobb gondossággal járt el. Ennek ellenére a könyvben előfordulhatnak hibák. Az ezen hibákból eredő esetleges károkért sem a szerző sem a kiadó semmiféle felelősséggel nem tartozik, de a kiadó szívesen fogadja, ha e hibákra felhívják figyelmét.

ISBN 963 86392 7 X
E-book ISBN 9786156364227

Kiadja a BBS-INFO Kft.
1630 Budapest, Pf. 21.
Felelős kiadó: a BBS-INFO Kft. ügyvezetője
Készült a debreceni Kinizsi nyomdában
Felelős vezető: Bördős János ügyvezető igazgató

TARTALOMJEGYZÉK

AJÁNLÁS	13
ELŐSZÓ	15
1. MI AZ AZ XHTML?	17
1.1. Mi az a HTML?.....	18
1.2. Mi az az XML?	19
1.3. Miért van szükség az XHTML-re?.....	19
2. A HTML és az XHTML eltérései	20
2.1. Áttérés HTML-ről XHTML-re	20
2.1.1. Jólformáltság.....	20
2.1.2. Kisbetűs megadás	20
2.1.3. A lezáró tag kötelező.....	21
2.1.4. Az attribútum-értékek megadása	21
2.1.5. Attribútum-minimalizáció	21
2.1.6. Üres elemek megadása	22
2.1.7. Whitespace attribútum-értékekben	22
2.1.8. Script és style elemek.....	22
2.1.9. Az id és a name attribútum	23
2.1.10. Attribútumok beépített értékhalommal.....	23
2.1.11. Egyed-hivatkozások hexa értékekkel	23
2.2. Kompatibilitás a HTML-lel	24
2.2.1. A műveletek feldolgozása	24
2.2.2. Üres elemek	24
2.2.3. Elemek minimalizálása.....	25
2.2.4. Beágyazott stíluslapok és scriptek.....	25
2.2.5. Sortörés attribútum-értékekben.....	25
2.2.6. Isindex.....	25
2.2.7. A lang és az xml:lang attribútumok.....	25
2.2.8. Részazonosítók	25
2.2.9. Karakterkódolás	26
2.2.10. Logikai értékek	26
2.2.11. A DOM és az XHTML.....	27

2.2.12. Az & használata	27
2.2.13. A CSS és az XHTML kapcsolata	27
3. AZ XHTML 1.0.....	29
3.1. Előírások.....	29
3.1.1. Szigorú megfelelési feltételek	29
3.1.2. XHTML használata más névvel	31
3.2. XHTML 1.0 dokumentumtípus-definíciók	32
3.2.1. Az XHTML 1.0 Strict DTD	32
3.2.1.1. Karakterkészlet-definiálások	32
3.2.1.2. Importált nevek	33
3.2.1.3. Általános attribútumok.....	34
3.2.1.4. Szöveges elemek.....	36
3.2.1.5. Blokkszintű elemek	38
3.2.1.6. Tartalmi modellek.....	40
3.2.1.7. Dokumentum-szerkezeti elemek	42
3.2.1.8. Dokumentum-fejrész	42
3.2.1.9. Dokumentum-törzs.....	45
3.2.1.10. Bekezdések	45
3.2.1.11. Címsorok	45
3.2.1.12. Felsorolások	45
3.2.1.13. Cím	46
3.2.1.14. Vízszintes határoló.....	46
3.2.1.15. Előformázott szöveg	46
3.2.1.16. Hosszú idézet.....	46
3.2.1.17. Beillesztett és törölt szöveg	47
3.2.1.18. A horgony elem.....	47
3.2.1.19. Inline elemek	48
3.2.1.20. Objektumok.....	49
3.2.1.21. Képek	50
3.2.1.22. Kliens-oldali térképek	51
3.2.1.23. Űrlapok	52
3.2.1.24. Táblázatok.....	56
3.2.2. Az XHTML 1.0 Transitional DTD	60
3.2.2.1. Karakterkészletek	60
3.2.2.2. Importált nevek	61
3.2.2.3. Általános attribútumok.....	62
3.2.2.4. Szövegelemek	65
3.2.2.5. Blokkszintű elemek	68
3.2.2.6. Tartalmi modellek.....	69
3.2.2.7. Dokumentumszerkezet.....	69
3.2.2.8. A dokumentum-fejrész.....	70
3.2.2.9. Keretek	72
3.2.2.10. Dokumentumtörzs	73
3.2.2.11. Bekezdések.....	73
3.2.2.12. Címsorok	73

3.2.2.13.	Felsorolások	74
3.2.2.14.	Cím.....	76
3.2.2.15.	Vízszintes tagoló	76
3.2.2.16.	Előformázott szöveg	76
3.2.2.17.	Hosszú idézet	76
3.2.2.18.	Szöveg-igazítás.....	77
3.2.2.19.	Beillesztett és törölt szöveg	77
3.2.2.20.	A horgony elem.....	77
3.2.2.21.	Inline elemek	78
3.2.2.22.	Objektumok	80
3.2.2.23.	Java alkalmazások	81
3.2.2.24.	Képek	82
3.2.2.25.	Kliens-oldali térképek	82
3.2.2.26.	Úrlapok	83
3.2.2.27.	Táblázatok.....	87
3.2.3.	Az XHTML 1.0 Frameset DTD	92
3.2.3.1.	Karakterkészlet-definiálások	92
3.2.3.2.	Importált nevek	93
3.2.3.3.	Általános attribútumok.....	95
3.2.3.4.	Szöveges elemek.....	97
3.2.3.5.	Blokkszintű elemek	100
3.2.3.6.	Tartalmi modellek.....	101
3.2.3.7.	Dokumentum-szerkezet	101
3.2.3.8.	Dokumentum-fejrész	102
3.2.3.9.	Keretek	104
3.2.3.10.	Dokumentumtörzs.....	106
3.2.3.11.	Bekezdések	107
3.2.3.12.	Címsorok.....	107
3.2.3.13.	Felsorolások	107
3.2.3.14.	Cím	109
3.2.3.15.	Vízszintes határoló	109
3.2.3.16.	Előformázott szöveg.....	109
3.2.3.17.	Hosszú idézet	110
3.2.3.18.	Szöveg-igazítás.....	110
3.2.3.19.	Beillesztett és törölt szöveg.....	110
3.2.3.20.	A horgony elem	111
3.2.3.21.	Inline elemek	111
3.2.3.22.	Objektumok	113
3.2.3.23.	Java alkalmazások	114
3.2.3.24.	Képek	115
3.2.3.25.	Kliens-oldali térképek	116
3.2.3.26.	Úrlapok.....	117
3.2.3.27.	Táblázatok	121
3.3.	Egyedkészletek.....	126
3.3.1.	Latin-1 karakterkészlet	126
3.3.2.	Különleges karakterek	129

3.3.2.1.	Vezérlőkarakterek és alapvető latin karakterek	129
3.3.2.2.	Latin kiegészítő karakterek („A” kiegészítés)	129
3.3.2.3.	Tagolás-módosító karakterek	129
3.3.2.4.	Általános írásjelek	130
3.3.2.5.	Pénznem szimbólum	130
3.3.3.	Szimbólumok	130
3.3.3.1.	Latin kiegészítő karakterek („B” kiegészítés)	131
3.3.3.2.	Görög karakterek	131
3.3.3.3.	Általános írásjelek	132
3.3.3.4.	Írásjel-szimbólumok	133
3.3.3.5.	Nyilak	133
3.3.3.6.	Matematikai operátorok	133
3.3.3.7.	Vegyes szakmai jelek	135
3.3.3.8.	Geometriai alakzatok	135
3.3.3.9.	Vegyes szimbólumok	135
3.4.	Elem-megkötések	135
4.	XHTML Basic	136
4.1.	Az XHTML Basic alapjai	136
4.2.	Előírások	137
4.3.	XHTML Basic dokumentum tervezése	137
4.3.1.	Stíluslapok	138
4.3.2.	Scriptek és események	138
4.3.3.	Megjelenés	138
4.3.4.	Űrlapok	138
4.3.5.	Táblázatok	139
4.3.6.	Keretek	139
4.4.	Megfelelőség	139
4.5.	Az XHTML Basic dokumentumtípus	139
4.6.	Az XHTML Basic használata	140
4.7.	Az XHTML Basic dokumentumtípus-definíciója	141
4.7.1.	Az SGML nyílt katalógus-formájú definíció	141
4.7.2.	Az XHTML Basic driver	141
4.7.2.1.	A cím-funkciós deklarációk	142
4.7.2.2.	A kép modul	143
4.7.2.3.	A táblázat modul	143
4.7.2.4.	Űrlap modul	143
4.7.2.5.	A hivatkozás-elem modulja	143
4.7.2.6.	A dokumentum metainformációs modulja	144
4.7.2.7.	Base-elemek modulja	144
4.7.2.8.	Param elemek modulja	144
4.7.2.9.	Beágyazott objektum-modul	144
4.7.3.	XHTML Basic rendelkezések	144

5. XHTML 1.1	147
5.1. Az XHTML 1.1 dokumentumtípus-definíciók	147
5.1.1. Az SGML nyílt katalógus-formájú definíció	147
5.1.2. Az XHTML 1.1 driver	147
5.1.2.1. Szerkezet előtti újradeklarációs helyfenntartó	148
5.1.2.2. A dokumentum modell-modul példányosítása a szerkezetben	149
5.1.2.3. A moduláris szerkezeti modul	149
5.1.2.4. Szerkezet utáni újradeklarációs helyfenntartó	149
5.1.2.5. A szöveges modul	149
5.1.2.6. A hipervivatkozás modul	149
5.1.2.7. A felsorolás modul	150
5.1.2.8. A szerkesztés modul	150
5.1.2.9. A szövegírány-felülbíró modul	150
5.1.2.10. A Ruby modul	150
5.1.2.11. A prezentációs modul	150
5.1.2.12. Hivatkozás-elem modul	151
5.1.2.13. Dokumentum-metainformációs modul	151
5.1.2.14. Base elemek modulja	151
5.1.2.15. A script modul	151
5.1.2.16. A stíluslap modul	151
5.1.2.17. A kép modul	152
5.1.2.18. A kliens-oldali térkép modulja	152
5.1.2.19. A szerver-oldali térkép modulja	152
5.1.2.20. Param elemek modulja	152
5.1.2.21. Beágyazott objektum-modul	152
5.1.2.22. Táblázat modul	153
5.1.2.23. Űrlap modul	153
5.1.2.24. Örökölt jelölés	153
5.1.2.25. Dokumentum-szerkezeti modul	153
5.1.3. XHTML 1.1 rendelkezések	153
5.1.3.1. Az XHTML 1.1 dokumentum-modell	154
5.1.3.2. A modell kiterjesztése	154
5.1.3.3. Opcionális elemek a fej részben	154
5.1.3.4. Vegyes elemek	155
5.1.3.5. Inline elemek	155
5.1.3.6. Blokkos elemek	156
5.1.3.7. Az összes tartalmi elem	157
6. XHTML 2.0	158
6.1. Az XHTML 2.0 dokumentumtípus	158
6.1.1. Visszafele kompatibilitás	159
6.1.2. Az XHTML 2.0 és a megjelenítés	160
6.1.3. Tervezési célok	160
6.2. Megfelelőségi feltételek	161
6.3. Az XHTML 2.0 modulok	161

6.3.1. A szerkezeti modul	162
6.3.2. Blokk-szöveg modul	162
6.3.3. Inline szöveg modul	163
6.3.4. Hiperszöveg modul.....	164
6.3.5. Felsorolás modul.....	164
6.3.6. Hivatkozás modul.....	164
6.3.7. Metainformációs modul.....	165
6.3.8. Objektum modul	165
6.3.9. Ruby modul.....	166
6.3.10. Script modul	166
6.3.11. Stílus-attribútum modul	166
6.3.12. Stíluslap modul	166
6.3.13. Táblázat modul	167
6.3.14. XForms modul	168
6.3.15. XML események modulja	168
6.4. Relax NG XHTML 2.0 Driver	168
6.5. XHTML modul-implementációk	169
6.5.1. Attribútum-gyűjtemények	169
6.5.2. Szerkezet	171
6.5.3. Blokk-szöveg	172
6.5.4. Soron belüli szöveg	173
6.5.5. Hiperszöveg	175
6.5.6. Felsorolás.....	175
6.5.7. Hivatkozás.....	176
6.5.8. Metainformáció	176
6.5.9. Objektum	176
6.5.10. Scriptek.....	177
6.5.11. Stílus-attribútum	177
6.5.12. Stíluslap	177
6.5.13. Táblázatok.....	178
6.6. AZ XHTML RELAX NG támogató moduljai	179
6.6.1. Adattípusok	179
6.6.2. Események.....	181
6.6.3. Paraméterek.....	181
6.6.4. Felirat	182
6.7. A RELAX NG külső moduljai	182
6.7.1. Ruby.....	182
6.7.2. Ruby Driver a teljes Ruby-jelöléshez	183
6.7.3. XML események.....	183
7. PÉLDATÁR	185
7.1. HTML dokumentum XHTML-lé frissítése	186
7.2. Névhelyenként egy séma-dokumentum	190
7.3. Adicionális attribútumok definiálása DTD-ben	190
7.4. XML séma meghatározása	191

Tartalomjegyzék	11
FÜGGELÉK	192
A. XHTML attribútum-típusok	192
B. Az XHTML 2.0 modulok és elemeik	196
C. XHTML esemény-modulok.....	197
TÁRGYMUTATÓ	198
WEBES KISSZÓTÁR	200

AJÁNLÁS

Ajánlom ezt a könyvet kedvesemnek, barátaimnak és mindenkinek, aki buzdított a könyv írására és erőt adott, hogy képes legyek kitartani a könyvírás és a példaprogramok készítése jegyében töltött nehéz órákban.

Sokszor a kívülállók számára már-már kórosan sok „gépezés” volt az, ami meghozta a várt eredményt. A munka nem mindig volt megszakítható, ezért néha már-már összetűzésbe kerültem olyanokkal is, akikkel nem szerettem volna. Ezúttal is köszönöm türelmüket és megértésüket.

Köszönöm továbbá a szeretetet és támogatást, mely nélkül ez a könyv nem jelenhetett volna meg.

A könyvben található valamennyi példaprogram (és számos más webes dokumentum is) fellelhető az interneten és ingyenesen letölthető. Az oldalra a kiadó honlapján (www.bbs.hu) a Szerzők menüben a Sikos László menüpont alatt található linkkel juthat el.

Az XHTML tanulmányozásához elengedhetetlen a HTML ismerete. Ehhez jó kiindulópont lehet a Szerző Bevezetés a HTML-be című fordítása, mely szintén az oldalról érhető el.

Ugyanitt található a különféle webes technológiák dokumentumaiból egy válogatás, mely hivatalos W3C-s fordításokat, hasznos linkgyűjteményt, valamint kliens- és szerver-oldali JavaScripteket is tartalmaz.

Ha bármilyen kérdése vagy észrevétele van a könyv témájával kapcsolatban, akkor az oldalon felveheti a kapcsolatot a Szerzővel e-mailben.

ELŐSZÓ

Manapság sokan vannak, akik valamilyen szinten megbarátkoztak a weblapfejlesztéssel, honlapokat publikálnak az interneten, akár komoly anyagi ellenszolgáltatás fejében is dolgoznak.

Ha valaki feltünteti a névjegykártyáján a weblapfejlesztő szót, számos ember egyszerűen csak kineveti, mondván, hogy ma már a weblapok létrehozása ugyanolyan szintre kezd érni, mint ahol 4-5 éve a szövegszerkesztés volt, tudniillik, hogy lassan már mindenki ért hozzá. Ez azonban nem így van! Ha valaki dolgozik, nemcsak egy művelet elvégzéséhez szükséges idő számít, hanem az igényelt eszközök, a felmerülő költségek és nem utolsósorban a munka minősége is. Ez a honlapok készítésénél úgy néz ki, hogy a legtöbben nekiállnak egy csillogó-villogó grafikus felületen fejleszteni, létrehoznak mindenféle „jó megjelenésű” weblapot, közben pedig még azt sem tudják, milyen dokumentumtípust hoztak létre. Valamilyen HTML-t. Ez így nem elfogadható. Milyen verziót? Érvényes-e a dokumentumunk? Betartottuk-e az előírt szabványokat, ajánlásokat? Milyen gyorsan jelenik meg az oldalunk egy nagyon lassú interneteléréssel rendelkező gépen? Milyen böngészőkön hogy fog kinézni a honlap? Egyáltalán működik-e a legtöbb böngészőn vagy egyszerűen csak egyetlen programban teszteltük az oldalt (mondjuk Internet Explorer 6.0-ban)? Biztosítottuk-e az oldal elérhetőségét? Hogyan jelenik meg az oldal egy tenyérszámítógép vagy egy mobiltelefon kifelbontású kijelzőjén? Egyáltalán meg tud-e jelenni? Mi történik, ha a weblapot egy nem-vizuális vagy netán egy akusztikai böngészővel szeretnénk megnyitni? Minimális-e a forráskód hossza? Milyen karakterkódolást alkalmaztunk a dokumentumban? És végül, de nem utolsósorban: értjük-e a forrás minden egyes karakterét? Ugye-ugye, ne is olyan egyszerű az a weblapszerkesztés!

Az interneten jelenleg uralkodó káosznak csak egyetlen módon lehetne véget vetni: a szabványosítással. Valahol persze az is jó, hogy egy kisebb-nagyobb hibákkal tűzdelt forráskódú dokumentumot is meg lehet jeleníteni, de így azok, akik csak összeszórják az oldalakat (és a legtöbb ember ezt csinálja), nem foglalkoznak ezzel a dologgal, nem is értik a problémát.

Ha így is működik, minek kínlódjunk a szabványokkal meg az ajánlásokkal? Hát azért, hogy a fenti kérdésekre választ kapjunk és olyan dokumentumokat tudjunk forrásszinten készíteni, melyek szabványosak, betartják az ajánlásokat és így a legtöbb böngészőben megjelennek, elérhetőségük biztosított, Japánban egy Macintuson is teljesértékű magyar nyelvű dokumentumként olvashatók, a mobiltelefon kijelzőjén is ugyanazt az információt biztosítják, mint a számítógép hatalmas monitorán stb.

A problémák elkerülésére, a szabványosításra, a bővíthető dokumentumtípusok létrehozására több próbálkozás is volt és van ma is. Az egyik lehetséges irány az XML és az ezen nyugvó XHTML dokumentumcsalád széleskörű elterjesztése és mindennapos használata. Jelen könyv tehát korunk egyik legfejlettebb webes nyelvét, az XHTML-t mutatja be, annak különféle verzióit felsorakoztatva, dokumentumtípus-definícióikat kielemezve, mintegy referenciát adva a profi, forrásszinten dolgozó XHTML-es weblapfejlesztők kezébe.

A nyelvvel való megismerkedéshez azonban hosszú és rögös út vezet. Az XHTML nyelvi elemeivel való megismerkedéshez mindenképp szükséges az alapos HTML-ismeret. Ha nem is tudjuk az összes elemet és azok attribútumait fejből, a legfontosabbakat feltétlenül ismernünk kell ahhoz, hogy átlássuk az XHTML nyelvcsalád felépítését. A nyelvvel való ismerkedéshez nem szükséges, de előnyös más webes technológiák, például a platformfüggetlen JavaScript nyelv alapszintű ismerete.¹ Ha valaki a szerver oldalon is programozott már, az további előnyt jelent.²

Ha úgy érzi a kedves Olvasó, hogy birtokában van a szükséges ismereteknek, szeretné kihasználni az internetben rejlő összes lehetőséget és korunk legfejlettebb webes nyelvén szeretné kifejezni magát, olvassa végig a könyvet, elemezze a példákat, mélyedjen el a dokumentumtípus-definíciókban, gyakoroljon bátran, mert a fejlesztést csak így lehet tanulni! Ehhez a korántsem egyszerű, de idővel nagyon hasznossá, sőt akár szórakoztatóvá is váló tevékenységhez kívánok sok kitartást:

A szerző

¹ Ehhez ajánlható Sikos László: JavaScript kliens oldalon című könyve

² Jó kiindulópont a szerző Szerver oldali webprogramozás című könyve

1. MI AZ AZ XHTML?

Ahhoz, hogy megismerhessük az XHTML különböző változatait, nem árt tisztában lennünk néhány fogalommal.

Az XHTML a jelenlegi és jövőbeli dokumentumtípusok olyan családja, mely a HTML 4-es verzióját reprodukálja, részhalmozokra bontja és kibővíti. Az XHTML dokumentumtípus-család XML alapú és ennek megfelelően XML alapú felhasználói rendszerekkel való munkához lett tervezve.

Az XHTML 1.0, mint neve is mutatja, az XHTML család első dokumentumtípusa. A HTML 4 három dokumentumtípusának megújulása XML 1.0 alkalmazásával. Segítségével olyan tartalmakat készíthetünk, melyek mind az XML, mind a HTML dokumentumokat támogató készülékeken megtekinthetők. Az XHTML 1.0-ra váltó weblapfejlesztők az alábbi előnyöket élvezhetik:

- Mivel az XHTML dokumentumok XML-konformosak, egyszerűen megtekinthetők, szerkeszthetők és kiértékelhetők a hagyományos XML eszközökkel.
- Az XHTML dokumentumokat úgy meg lehet írni, hogy ugyanúgy vagy akár a korábbinál jobban működjenek a meglévő HTML-böngészőkkel éppúgy, mint az XHTML 1.0 böngészőkkel.
- Az XHTML dokumentumok felhasználhatnak a HTML dokumentum objektum-modellre és az XML dokumentum objektum-modellre épülő alkalmazásokat (scripteket, appleteket) is.
- Az XHTML nyelvcsalád fejlődésével az XHTML 1.0 előírásainak megfelelő dokumentumok egyre tökéletesebben együtt tudnak működni a különféle XHTML környezetekben.

A webes dokumentumok következő lépcsője az XHTML. Az XHTML-re való áttéréssel párhuzamosan a weblapfejlesztők érvényes XML dokumentumokat készítenek, így nagy előnyre tesznek szert a HTML-dokumentumok szerkesztőivel szemben. Emellett nyugodtak lehetnek, hogy oldalaik „időtálló” lesznek, azaz egyidejűleg kompatibilisek visszafelé és előre felé is.

1.1. Mi az a HTML?

A HTML a Hypertext Markup Language terminológia rövidítése, azaz egyfajta hiperhivatkozásokat használó jelölőnyelvről van szó. A HTML egyébként az SGML (Standard Generalized Markup Language) nyelvcsaládba tartozik, így megfelel az ISO 8879 előírásainak (emiat széles körben elfogadott és támogatott). Tulajdonságaiból kifolyólag alkalmas a webes dokumentumok leírásához.

Az SGML az 1980-as évek közepe felé született és meglehetősen stabil maradt. Rendkívüli stabilitása a nyelv tulajdonság-gazdagságából és rugalmasságából ered. Ez az előnyös tulajdonság azonban bonyolultsággal párosult, így világviszonylatban nem terjedhetett el.

Eredetileg a HTML-t tudományos és műszaki dokumentumok készítéséhez tervezték, olyan felhasználóknak, akiknek nem kell ismerniük a dokumentumok szerkesztésének minden csínját-bínját. A HTML megoldotta az SGML-nél felmerülő bonyolult szerkezet problémáját is, mert a nyelvbe strukturáló és szemantikai elemek kis készletét építették. Ebből kifolyólag meglehetősen egyszerű dokumentumokat lehetett létrehozni.

A dokumentumok szerkezeti egyszerűsítésére a HTML kibővítette a hiperszöveges támogatást. Ekkor a HTML még nem biztosított teljesértékű multimédiás felületet, erre csak a későbbi fejlesztések nyomán lett képes.

A HTML nagyon rövid idő alatt széles körben népszerűvé vált. Ezzel párhuzamosan túlhaladta eredeti célját.

Már kezdetben is próbálkoztak különlegesebb elemek HTML-be olvasztásával, de mivel akkor még nem a speciális igények kiszolgálása volt az elsődleges cél, így pár évnek el kellett telnie, mire fokozatosan teret hódítottak különleges célú kiterjesztései is. A kiterjesztések széles körű elterjedést viszont akadályozta, hogy minél szélesebb rétegeket céloztak meg, annál nehezebb volt a felmerülő kompatibilitási problémák legyőzése.

Az internet rendkívüli fejlődésével párhuzamosan gyorsan terjedt a HTML nyelv is. Egy idő után azonban kezdett túlhaladottá válni, mert bár a nyelv rendkívüli képességekkel rendelkezik (közönséges szöveg alapú felépítés, full multimédiás felületek megjelenítése, scriptek, appletek futtatása stb.), az egyes szoftver- és hardverkörnyezetekben egyre kevésbé lehetett biztosítani, hogy a dokumentumok az eredetileg megszerkesztett dokumentumhoz hasonlóak legyenek. Sokszor már azt nehéz (vagy lehetetlen) volt elérni, hogy egyáltalán megjeleníthetők legyenek dokumentumaink. Egy szó, mint száz, a helyzet tarthatatlanná vált.

Akik tehát jelenleg is HTML dokumentumokat készítenek, azoknak előbb-utóbb át kell térniük egy fejlettebb nyelvre, ha fel akarják venni a

versenyt a jövőbeli honlapokkal. Ideig-óraig még tartható az állapot, de mindenkinek be kell látnia: a weben paradigmaváltásra van szükség.

1.2. Mi az az XML?

Az XML az Extensible Markup Language (Kiterjeszhető Leíró Nyelv) kezdőbetűiből alkotott betűszó.

Az XML eredeti feladata az SGML hatékonyságának visszanyerése, annak bonyolultsága nélkül. Bár az XML az SGML szűkített formája, megőrzi az SGML nyelvi gazdagságának nagy részét, így az XML-ben sem kell lemondanunk az SGML általánosan használt tulajdonságairól.

Mivel az előnyös tulajdonságok megtartásával párhuzamosan a bonyolult sajátosságokat mellőzni kell, a fejlesztéshez felhasználható szoftverek előállítására nem egyszerű feladat.

1.3. Miért van szükség az XHTML-re?

Az XHTML nyelvcsaládra történő áttérésnek számos előnye van. Nézzünk néhányat közülük:

- A dokumentumok és a böngészők tervezői folyamatosan kutatják az új kifejezési lehetőségeket. Az XML-ben viszonylag egyszerű az új elemek vagy attribútumok bevezetése. Az XHTML család úgy lett tervezve, hogy ezeket a kiterjesztéseket hozzáillessze az XHTML modulokon és technikákon keresztül az újonnan kifejlesztendő XHTML-konform modulokhoz. Ezek a modulok lehetővé teszik a létező és új tulajdonság-készletek kombinációját a tartalomfejlesztés és böngészőtervezés során.
- Folyamatosan kerülnek bevezetésre alternatív megoldások az internethez történő hozzáférésre. Az XHTML család együttműködik a legszélesebb körben elterjedt böngészőkkel. Egy új böngészőtől kezdve a dokumentumprofil kialakításán át a szerverekkel, proxykkal, meglévő böngészőkkel bezárólag valamennyi eszköz a legjobb teljesítménnyel végezheti el a tartalom átalakítását. Végül, de nem utolsó sorban lehetséges saját XHTML-konform tartalom fejlesztése, mely használható bármely XHTML-konform böngészővel.

2. A HTML és az XHTML eltérései

2.1. Áttérés HTML-ről XHTML-re

Mivel az XHTML egy XML alkalmazás, egyes technikai megoldások, melyek teljesen legálisak voltak az SGML alapú HTML-ben, megérették a cserére.

2.1.1. Jólformáltság

A jólformáltság az XML-lel bevezetett új koncepció. Ez a gyakorlatban azt jelenti, hogy az összes elemet le kell zárni lezáró tag-gel vagy egy speciális megadási formával, valamint az elemeket csak következetesen ágyazhatjuk egymásba.

Bár az átlapolás az SGML-ben sem megengedett, a napjainkban használt böngészők széles körben támogatják ezt a gyatra megadási formát. Az tehát megengedett, hogy egymásba ágyazzuk az elemeket, de amelyik elemet először szerepeltetjük, értelemszerűen azt kell legutoljára lezárni. (Aki következetesen szerkesztette HTML forrásait, annak ez nem okoz gondot.) Nézzünk egy példát!

```
<p> álljon itt egy <em>kiemelt szövegrész</em></p>
```

Aki rossz szokásként eddig esetleg használta az átlapolást, az most sürgősen felejtse el! Különben is hogy néz ki ez a forráskód?!

```
<p> álljon itt egy <em>kiemelt szövegrész</p></em>
```

Ez tehát az XHTML-ben helytelen megadás!

2.1.2. Kisbetűs megadás

Az XHTML dokumentumokban az összes HTML elemet és attribútumnevet kisbetűvel kell írni! Ez a változtatás azért volt szükséges,

mert az XML kisbetű-nagybetű érzékeny (ún. case sensitive). Így tehát a `` és a `` tag nem ugyanaz a tag.

A HTML forráskódban sokszor azért szoktunk nagybetűvel tag-eket megadni, hogy azt jól elkülönítsük a szövegtől. Erről mondjunk le, helyette inkább a sortörésekkel és a behúzásokkal tegyük olvashatóvá forráskódunkat. Higgjuk el, nem is olyan zavaró, sőt, egy idő után jó érzés tölt el bennünket, hogy a forráskódban nem felváltva kis- vagy nagybetűvel vannak írva a tag-ek, hanem egységesen kisbetűvel.

2.1.3. A lezáró tag kötelező

Az XHTML-ben mindent le kell zárni, még a nem üres tag-eket is. Az SGML alapú HTML-ben bizonyos elemek záró tag-je elhagyható. Ezek olyan elemek, melyek értelmileg magukban hordozzák a lezárást (például egy fejezetet egy másik követ, az elsőt nem kell lezárni). Az XML nem engedi meg a lezáró tag-ek elhagyását. Minden elemet, ami a DTD-ben nem EMPTY-ként (üresként) van deklarálva, le kell zárni. Azon elemek esetében, melyek a DTD-ben EMPTY-ként vannak deklarálva, lehet alkalmazni a lezáró tag-et vagy az üres elem rövidített jelölését is.

Az XHTML-ben a lezárt elemek megadása tehát a helyes:

```
<p> ez egy fejezet </p><p> egy egy másik fejezet </p>
```

A le nem zárt elemek megadása pedig helytelen:

```
<p> ez egy fejezet <p> egy egy másik fejezet
```

2.1.4. Az attribútum-értékek megadása

Az XHTML-ben az attribútum-értékeket mindig idézőjelek között kell megadni. Ez minden értékre vonatkozik, a numerikus értékekre is.

Az alábbi megadás tehát helyes az XHTML-ben:

```
<td colspan="2">
```

Az alábbi viszont csak a HTML-ben:

```
<td colspan=2>
```

2.1.5. Attribútum-minimalizáció

Az XML nem támogatja az attribútum-minimalizációt. Az attribútum-érték párokat teljesen ki kell írni. Az olyan attribútum-értékek, mint a