

Sikos László:

# **Stíluslapok a weben**

**CSS kézikönyv**



Sikos László:

# **Stíluslapok a weben**

## **CSS KÉZIKÖNYV**

BBS-INFO Kiadó – 2005.

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 9425 03 6  
E-book ISBN 9786156364180

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

<b>AJÁNLÁS</b> .....	<b>11</b>
<b>ELŐSZÓ</b> .....	<b>13</b>
<b>1. A STÍLUSLAPOK ALAPJAI</b> .....	<b>15</b>
1.1. Történeti áttekintés .....	15
1.2. A jelölőnyelvek és a stíluslapok kapcsolata .....	17
1.2.1. A HTML és a CSS .....	17
1.2.1.1. A belső stílus .....	18
1.2.1.2. A külső stíluslap .....	18
1.2.2. Az XML és az XSL .....	20
1.2.3. A CSS feldolgozása .....	22
1.2.3.1. A vászon .....	23
1.2.3.2. A CSS címezése .....	23
1.2.3.3. A CSS tervezése .....	23
<b>2. MEGFELELŐSÉG</b> .....	<b>26</b>
2.1. Ajánlások .....	26
2.1.1. Hibafeltételek .....	27
2.1.2. A text/css tartalomtípus .....	27
<b>3. SZINTAXIS ÉS ADATTÍPUSOK</b> .....	<b>29</b>
3.1. Szintaxis .....	29
3.2. Jelölések .....	29
3.3. Kulcsszavak .....	31
3.3.1. Specifikus kiterjesztések .....	31
3.3.2. Történeti megjegyzések .....	32
3.3.3. Karakterkezelés .....	32
3.3.4. Utasítások .....	33
3.3.5. At-szabályok .....	33
3.3.6. Blokkok .....	34
3.3.7. Szabályhalmazok, deklarációs blokkok, szelektorok .....	34
3.3.8. Deklarációk, tulajdonságok .....	35
3.3.9. Megjegyzések .....	36

3.4. Hibakezelési szabályok .....	36
3.5. Értékek .....	38
3.5.1. Egész és valós számok .....	38
3.5.2. Hosszértékek .....	38
3.5.2.1. Relatív hosszértékek .....	38
3.5.2.2. Abszolút hosszértékek.....	41
3.5.3. Százalék-értékek.....	41
3.5.4. URL, URN, URI értékek.....	42
3.5.5. Számlálók.....	42
3.5.6. Színek.....	43
3.5.7. Stringek .....	45
3.5.8. Nem támogatott értékek.....	45
3.6. Stíluslap-reprezentáció .....	46
3.6.1. Hivatkozás a karakterkódolásban nem ábrázolható karakterre.....	47
<b>4. Szelektorok .....</b>	<b>48</b>
4.1. Egyezés .....	48
4.2. Szintaxis .....	49
4.2.1. Csoportosítás.....	49
4.3. Univerzális szelektor .....	50
4.4. Típuszelektorok.....	50
4.5. Leszármazott szelektorok .....	50
4.6. Gyerekszelektorok .....	51
4.7. Szomszédos testvérszelektorok .....	51
4.8. Attribútumszelektorok .....	52
4.8.1. Attribútum- és attribútumérték-egyezőség .....	52
4.8.2. Alapértelmezett attribútumértékek DTD-kben .....	53
4.8.3. Osztályszelektorok .....	54
4.9. Azonosító szelektorok.....	55
4.10. Álelemek és álosztályok.....	56
4.10.1. Álelemek .....	56
4.10.1.1. A :first-line álelem.....	56
4.10.1.2. A :first-letter álelem .....	59
4.10.1.3. A :before és :after álelemek .....	62
4.10.2. Álosztályok .....	63
4.10.2.1. A :first-child álosztály .....	63
4.10.2.2. A :link és :visited álosztályok.....	63
4.10.2.3. Dinamikus álosztályok .....	64
4.10.2.4. A :lang álosztály .....	65
<b>5. Tulajdonságértékek meghatározása, lépcsőzetesség, öröklődés... 67</b>	
5.1. Megadott, számított és aktuális értékek .....	67
5.1.1. Megadott értékek.....	67
5.1.2. Számított értékek.....	67
5.1.3. Használt értékek .....	68
5.1.4. Aktuális értékek .....	68
5.2. Öröklődés .....	68

5.2.1. Az örökölt érték .....	69
5.3. Az @import szabály .....	69
5.4. A lépcsőzetesség .....	70
5.4.1. A sorrend a lépcsőzetességben.....	71
5.4.2. Az !important szabályok.....	71
5.4.3. Egy szelektor specifikusságának számítása .....	72
5.4.4. Nem CSS-es jelölések precedenciája .....	73
<b>6. Médiatípusok.....</b>	<b>74</b>
6.1. Mik azok a médiatípusok?.....	74
6.2. Médiafüggő stíluslapok.....	74
6.2.1. Az @media szabály .....	75
6.3. Felismert médiatípusok .....	75
6.3.1. Média csoportok .....	76
<b>7. A dobozmodell.....</b>	<b>77</b>
7.1. A doboz kiterjedése .....	77
7.2. Margók, helykitöltés, keretek .....	78
7.2.1. Margó-tulajdonságok.....	80
7.2.1.1. A margók meghatározása egyesével .....	81
7.2.1.2. A margók egyidejű meghatározása .....	81
7.2.1.3. A margók összevonása .....	82
7.2.2. A helykitöltés tulajdonságai.....	82
7.2.3. Keret-tulajdonságok .....	83
7.2.3.1. Keretszélesség .....	83
7.2.3.2. Keretszín .....	84
7.2.3.3. Keretstílus .....	85
7.2.3.4. Rövidített kerettulajdonságok .....	87
7.3. Az inline elemek dobozmodellje kétirányú szövegeknél.....	88
<b>8. Vizuális formázási modell .....</b>	<b>89</b>
8.1. Dokumentumfa készítése vizuális médiához .....	89
8.1.1. A viewport.....	89
8.1.2. Tartalmazó blokkok .....	90
8.2. Vezérlődoboz-generálás .....	90
8.2.1. Blokk szintű elemek és blokkdobozok.....	90
8.2.1.1. Névtelen blokkdobozok.....	90
8.2.2. Inline szintű elemek és inline dobozok .....	91
8.2.2.1. Névtelen inline dobozok.....	92
8.2.3. Betoldott dobozok.....	92
8.2.4. A display tulajdonság .....	93
8.3. Pozicionálás .....	94
8.3.1. Pozicionálási séma választása a position tulajdonsággal.....	94
8.3.2. Doboz-eltolás: top, right, bottom, left .....	95
8.4. Normál elhelyezés.....	97
8.4.1. Blokkformázás .....	97
8.4.2. Inline formázás.....	97

8.4.3. Relatív pozicionálás .....	98
8.5. Lebegő elhelyezés .....	99
8.5.1. A float tulajdonság .....	100
8.5.2. A clear tulajdonság .....	101
8.6. Abszolút pozicionálás .....	101
8.6.1. Rögzített pozicionálás .....	102
8.7. A display, a position és a float kapcsolata .....	103
8.8. Rétegelt megjelenítés .....	104
8.8.1. A z-index tulajdonság .....	104
8.9. Szövegrány: a direction és a unicode-bidi tulajdonságok .....	105
<b>9. Vizuális hatások .....</b>	<b>107</b>
9.1. Körülfolytatás és csonkolás .....	107
9.1.1. Az overflow tulajdonság .....	107
9.1.2. A clip tulajdonság .....	108
9.2. Láthatóság, a visibility tulajdonság .....	109
<b>10. Generált tartalom, felsorolás, számozás .....</b>	<b>110</b>
10.1. A :before és az :after álelemek .....	110
10.2. A content tulajdonság .....	110
10.3. Idézőjelek .....	111
10.3.1. Idézőjelek megadása a quotes tulajdonsággal .....	111
10.3.2. Idézőjelek beszúrása a content tulajdonsággal .....	112
10.4. Automatikus számozások .....	113
10.4.1. Beágyazott számozások, hatáskör .....	114
10.4.2. Számozás-stílusok .....	115
10.4.3. Számozás display:none-os elemekben .....	115
10.5. Felsorolások .....	115
10.5.1. A felsorolás stílusa: list-style-type .....	116
10.5.2. Képes felsorolásijel: list-style-image .....	117
10.5.3. A felsorolásijel pozíciója: list-style-position .....	118
10.5.4. Összesített meghatározás: list-style .....	119
<b>11. A lapozós médiatípus .....</b>	<b>121</b>
11.1. Oldaldobozok: az @page szabály .....	121
11.1.1. Oldalmargók .....	121
11.1.2. Oldalszelektorok .....	122
11.1.3. Az oldaldobozon kívüli tartalom .....	123
11.2. Oldaltörések .....	123
11.2.1. Oldaltörés-tulajdonságok .....	123
11.2.2. Elemeken belüli törések: orphans, widows .....	124
11.2.3. Engedélyezett oldaltörések .....	124
11.2.4. Kikényszerített oldaltörések .....	125
11.2.5. A „legjobb” oldaltörések .....	125
11.3. Lépcsőzetesség az oldalak vonatkozásában .....	126



<b>12. Színek, háttérszínek.....</b>	<b>127</b>
12.1. Az előtér színe, a color tulajdonság .....	127
12.2. A háttérszín .....	127
12.2.1. A háttér tulajdonságai.....	128
12.2.1.1. A background-color tulajdonság .....	128
12.2.1.2. A background-image tulajdonság.....	128
12.2.1.3. A background-repeat tulajdonság .....	129
12.2.1.4. A background-attachment tulajdonság .....	130
12.2.1.5. A background-position tulajdonság .....	131
12.2.1.6. A background tulajdonság.....	131
12.3. Gamma-korrekción .....	132
<b>13. Betűtípusok.....</b>	<b>133</b>
13.1. Betűtípus-egyezőség.....	133
13.2. Betűtípus-család, a font-family tulajdonság .....	134
13.3. Betűstílus, a font-style tulajdonság.....	134
13.4. Kisméretű nagybetűk, a font-variant tulajdonság.....	135
13.5. Félkövér betűk, a font-weight tulajdonság .....	135
13.6. Betűméret, a font-size tulajdonság.....	137
13.7. A font tulajdonság .....	138
<b>14. Szövegformázás .....</b>	<b>140</b>
14.1. Szöveg-behúzás, a text-indent tulajdonság .....	140
14.2. Szövegigazítás, a text-align tulajdonság.....	140
14.3. Díszítés, kiemelés: text-decoration .....	141
14.4. Betűköz és szóköz: letter-spacing, word-spacing .....	142
14.5. Nagybetűssé alakítás, a text-transform tulajdonság .....	143
14.6. A whitespace tulajdonság .....	143
<b>15. Táblázatok.....</b>	<b>144</b>
15.1. A táblázat-modell.....	144
15.1.1. Névtelen táblázat-objektumok .....	145
15.2. Oszlopok.....	146
15.3. Táblázatok a vizuális formázási modellben .....	147
15.3.1. Feliratok igazítása.....	148
15.4. A táblázat-tartalom vizuális megjelenése.....	148
15.4.1. Táblázat-rétegek, átlátszóság .....	148
15.4.2. Táblázat-szélesség: a table-layout tulajdonság.....	151
15.4.3. Táblázat-magasság.....	151
15.4.4. Vízszintes igazítás egy oszlopban .....	152
15.4.5. Dinamikus hatások .....	152
15.5. Keretek.....	153
15.5.1. Különálló keretek.....	153
15.5.1.1. Üres cellák kerete, háttere: az empty-cells tulajdonság....	154
15.5.2. Összevont keretek .....	155
15.5.3. Keret-stílusok.....	156

<b>16. Felhasználói interfész .....</b>	<b>157</b>
16.1. Kurzorok, a cursor tulajdonság .....	157
16.2. A CSS rendszerszínek.....	158
16.3. Dinamikus körvonalak, az outline tulajdonság .....	161
<b>FÜGGELÉK.....</b>	<b>163</b>
A. A HTML 4.0 alapértelmezett stíluslapja .....	163
B. Összesített tulajdonságlista.....	165
C. A CSS dobozmodellje .....	178
D. A CSS 2.1 nyelvtana.....	179
E. A CSS és az XSL kapcsolata .....	181
<b>TÁRGYMUTATÓ.....</b>	<b>182</b>
<b>KISSZÓTÁR .....</b>	<b>186</b>

## AJÁNLÁS

Ajánlom ezt a könyvet kedvesemnek, családomnak, barátaimnak és mindenkinek, aki bíztatott a könyv írására, a példaprogramok és ábrák készítése alatt eltöltött nehéz órákban. Sokszor a kívülállók számára érthetetlenül sok „gépezés” hatására készült el egy-egy nehezebb, szárazabb fejezet. Ezúttal is köszönöm mindazok türelmét és megértését, akik nélkül ez a könyv nem készülhetett volna el.

Ajánlom a könyvet továbbá minden lelkes amatőr és profi weblapfejlesztőnek, aki a jelölőnyelvek ismerete, a szakmai angol nyelvtudás, a grafikák készítésének képessége és számos egyéb felhalmozott tudás és kompetencia mellett sem igazodnak el a stíluslapok útvesztőin.

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](http://www.bbs.hu)) a Szerzők menüben a Sikos László menüpont alatt található linkkel juthat el.

A CSS megismeréséhez elengedhetetlen a jelölőnyelvek, a HTML/XHTML ismerete. Ehhez jó kiindulópont lehet Bártfai-Kovács: Weblapkészítés házilag c. könyve, a Szerző Bevezetés a HTML-be című fordítása, (mely szintén az oldalról érhető el), de további hasznos olvasmány lehet az érdeklődőknek a Szerző XHTML – A HTML megújulása XML alapokon című könyve is.

Szintén a honlapon 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

Napjainkban a web különféle területein is krízishelyzet alakult ki. Köszönhető ez a hozzá nem értő vagy hanyag fejlesztőknek, az ajánlások nem ismerésének, be nem tartásának, a szabványosítás szinte teljes hiányának. Hiába a sok új technológia, mely számos lehetőséget rejteget, ha még a régebbiekkel sem boldogulunk! A weblapok megfelelő, egységes megjelenése stíluslapok nélkül elképzelhetetlen. Míg néhány éve az egyszerű, karakteres, kódolástól függően esetleg helytelenül megjelenő, fényképeket, táblázatokat nem tartalmazó dokumentumok voltak jellemzők a számítógépek körében, addig mára a helyzet gyökeresen megváltozott. Az új technológiák új igényeket vetettek fel mind a fejlesztők, mind a felhasználók körében.

A weblapfejlesztők egyre inkább a grafikus felületű „kattintgatás” irányába mennek el, csak néhány igazi szakember marad meg a forrásszintnél, az elérhetőség biztosításánál, a tiszta forráskódnál.

Sokan vannak azonban közülük is olyanok, akik nincsenek tisztában a stíluslapok használatának lehetőségeivel és korlátaival, hiszen a korábbiakban kizárólag vagy nagy részben a jelölőnyelv irányából szemlélték a honlapokat.

Ők azok, akik el-ellesnek egy-egy megoldást már meglévő weblapokról, számos ötletük van, de nem tudják megvalósítani. Időről időre összekevernek egyes tulajdonságokat, fogalmakat s összetévesztik az egyes tulajdonság-értékek lehetséges értékeit is. Ennek eredményeképpen készülhet ugyan valamilyen forráskód, de a kívánt eredmény elérhetetlen marad. Az értékadás kettőspontos megadása, a tulajdonság-nevek kötőjeles írása idegen tőlük, hiszen nem ezt szokták meg a leírónyelvben.

A CSS megismerése azért rendkívül fontos, mert napjaink weblapjaihoz (ha másért nem, hát a méretből kifolyólag) feltétlenül szükséges stíluslapokat alkalmazni. Ezek közül a leggyakoribb a külső CSS fájl használata. A régebbi leírónyelven készült dokumentumok éppúgy használhatják, mint a jelenlegi vagy a legújabb (pl. XHTML 1.1) nyelvű weblapok.

A CSS-nek több szintje is kialakult az idők folyamán. Jelen könyv elsősorban a legújabb, teljesen kiforrott verziót, a CSS 2.1-et veszi alapul, min-

den helyen kitérve a régebbi (vagy leendő jövőbeli) verziókhoz képest jellemző esetleges eltérésekre, sajátosságokra.

Még a tapasztaltabb weblaposoknak is számos meglepetést tartogathat a CSS. Jé, még ilyet is lehet csinálni a honlapon? Ráadásul ilyen egyszerűen? A ráeszmélés néha letaglózó, ekkor érzi igazán az ember, hogy a weblapokhoz soha sem lehet igazán felnőni. Százasaival vannak olyan lehetőségek, melyeket akár több éves internetezés alatt sem láthatunk sehol, pedig a lehetőség adva van. Csak éppen nem használja ki senki sem.

A szövevényes, nyakatekert, megtévesztő megfogalmazások, az angol rövidítések közötti eligazodást a könyv egyes fejezetei, a végén található függelék és kiegészítő könnyíti meg.

Ha úgy érzi a kedves Olvasó, hogy birtokában van a szükséges ismereteknek, szeretné kihasználni az internetben rejlő összes lehetőséget és körünk egyik legkorszerűbb webes stíluslapját, a CSS 2.1-et szeretné használni, olvassa végig a könyvet, elemezze a példákat, gyakoroljon bátran, írja át a példasorokat, készítsen saját forráskódokat, mert a fejlesztést csak így lehet tanulni! Ehhez a korántsem egyszerű, de idővel nagyon hasznossá, sőt akár szórakoztatóvá is váló tevékenységhez kívánok sok kitaratást:

*A szerző*

# 1. A STÍLUSLAPOK ALAPJAI

## 1.1. Történeti áttekintés

A webes stíluslapok története az 1994-es esztendőben kezdődött. A Webet eredetileg elektronikus publikálási célokra használták. Ehhez azonban kezdetben hiányzott egy fontos dolog: nem volt megoldva a dokumentumok formázása, stílusokkal való ellátása (például nem lehetett egy újságoldalhoz hasonló oldalt webes formában megjeleníteni). Hákon Wium Lie újság-prezentációkkal foglalkozott az MIT médialaboratóriumában. Többek között ő volt az, aki felismerte, hogy szükség van egy webes stíluslap-nyelvre.

A böngészőkben használatos stíluslapok ötlete nem volt teljesen új. A HTML leírónyelv célja már 1990-ben az volt, hogy a dokumentum szerkezetét és kinézetét elválassza egymástól. Tim Berners-Lee is úgy írta meg a NeXT nevű böngésző-fejlesztő eszközt, hogy a stílust egy egyszerű stíluslappal le lehessen írni. A stíluslapok szintaxisát azonban nem publikálta, gondolva arra, hogy a böngésző döntse el, mi a legjobb megjelenés a felhasználóknak. Más böngészők, például az 1992-es Pei Wei's Viola vagy az 1993-as Harmony Browser is hasonló stílusnyelvekkel rendelkezett.

Ahelyett azonban, hogy a stílusok fokozatosan továbbfejlődtek volna, a későbbi böngészők egyre kevesebb lehetőséget biztosítottak a stílus befolyásolására. 1993-ban jelent meg az NCSA Mosaic nevű böngésző, mely a Webet igazán népszerűvé tette. Stílus szempontjából azonban ez egy visszalépés volt, hiszen csak meghatározott színek és betűtípusok használatát támogatta.

Időközben a weblapfejlesztők ráeszméltek, hogy nem tudják igazán meghatározni oldalaik végső megjelenését. Ha valaki elkezdett foglalkozni a weblapfejlesztéssel, első kérdései között szerepelt a betűtípusok és színek megváltoztatásának módja. A HTML leírónyelv ekkoriban nem biztosította ezt a funkcionalitást (érthető módon). A levelezőlistákon már 1994 körül szép számmal voltak fellelhetők ezen problémák hosszabb-rövidebb leírásai. Ami egy Word-ben, egy TeX-ben vagy egyéb szövegszerkesztőben triviális volt, az a weben megoldhatatlan. Marc Andreessen, az NCSA

Mosaic egyik programozója is látta a problémát. 1994. október 13-án a www-talk-on már bejelentette a Mozilla nevű böngésző első tesztelhető béta verzióját (később ebből lett a Netscape Navigator). Néhány új tag mellett az új böngésző már támogatta a középre igazítást, melyet a későbbiekben további lehetőségek követtek. Három nappal a bejelentés előtt Håkon közzétette a lépcsőzetes HTML stíluslapok első vázlatát. Valójában Dave Raggett (a HTML 3.0 atyja) sürgette a tervezet nyilvánosságra hozatalát, hogy az még a közelgő chicago-i 'Mosaic és a Web' nevű konferencia előtt megjelenjen. Raggett volt az, aki megállapította, hogy a HTML-nek nem szabad egy oldalleíró nyelvvé válnia és soha nem is fog az lenni. Egy sokkal célorientáltabb mechanizmusra van szükség a fejlesztők igényeinek kielégítésére. Jóllehet a dokumentum első változata még kiforratlan volt, jó alapot biztosított a további fejlesztésekhez.

A CSS első változatának megalkotói között volt Bert Bos is, aki az Argo nevű browsert írta (ez már támogatta a beépülő modulokat, vagy ahogyan ő nevezte, applet-eket, még a Netscape előtt!). Az Argo egy jól testreszabható böngésző volt, ami már kezelte a stíluslapokat. Bos úgy döntött, hogy egyesíti erőit Wium Lie-vel. Bár mindkét javaslat erősen eltér napjaink CSS-étől, a származás vitathatatlan.

Az Argo stílusnyelvének egyik vonása az volt, hogy elég általános volt ahhoz, hogy a HTML-en kívül más jelölőnyelvet is alkalmazzon. Ez lett a CSS tervezésének egyik célja is, a HTML pedig hamarosan el lett távolítva a specifikáció címéből. Az Argo olyan fejlett vonásokkal is rendelkezett, mint az attribútum-szelektorok vagy a generált szöveg, melyek nem kerültek be még a CSS 1-es szintjébe, csak a CSS2-be.

A lépcsőzetes stíluslapok akkoriban nem az egyetlen stílusnyelv volt. A Viola böngészőhöz készült Pei Wei's language (Pei Wei nyelve) vagy az O'Reilly kiadónál dolgozó Robert Raisch nyelve ékes példák erre. Az utóbbi még 1993-ban készült! Nem szabad megfeledkezni az SGML nyelvek nyomtatásához készült ISO komplex stílusról és átalakító nyelvről, a DSSSL-ről sem. A DSSSL a HTML-hez is alkalmazható volt. A CSS-nek azonban volt egy tulajdonsága, ami a többi nyelv fölé emelte: lehetővé tette, hogy a Weben egy dokumentum stílusát a fejlesztő vagy a felhasználó saját maga is meghatározhassa, igényeik kombinálhatók legyenek („lépcsőzetességgel”). Figyelembe vette továbbá a fejlesztő és az olvasó igényein kívül a megjelenítő eszköz és a böngésző képességeit is.

A terveknek megfelelően a kezdeti CSS javaslatot bemutatták a chicago-i webes konferencián 1994 novemberében.

A következő webes konferenciát 1995 áprilisában tartották Németországban. Bos és Wium Lie itt találkozott először személyesen. Bos az Argo-hoz mutatott be stíluslap-támogatást, Wium Lie pedig az Arena böngészőt módosította a stíluslapok támogatásához. Az Arena-t Dave Raggett írta az új ötletek (köztük a stíluslapok) teszteléséhez.



A világ legnagyobb, webes ajánlásokat készítő konzorciuma, a W3C (World Wide Web Consortium) 1995 végére készítette el a HTML Editorial Review Board-ot (HTML ERB) a jövőbeli HTML specifikációk ratifikálásához. Ekkor a stíluslapok már az érdeklődés középpontjában voltak.

1997 februárjában a CSS saját munkacsoportot kapott a W3C-n belül. A munkacsoport elkezdte vizsgálni a CSS1-ből hiányzó képességeket. A csoport vezetője a skót Chris Lilley volt a Manchester-i egyetemről.

A CSS 2. szintje 1998 május 12-én vált ajánlássá.

1999. január 11-én adták ki a CSS1 javított ajánlását.

2001 áprilisában jelent meg a Bevezetés a CSS3-ba című tervezet, melyet azóta is folyamatosan fejlesztenek.

## 1.2.A jelölőnyelvek és a stíluslapok kapcsolata

### 1.2.1. A HTML és a CSS

Az alábbiakban látni fogjuk, milyen egyszerű a stíluslapok tervezése. Nem árt azonban tisztában lennünk néhány alapvető kiadványszerkesztési fogalommal és a HTML-lel.

Nézzünk egy egyszerű HTML dokumentumot:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Vad motorosok</TITLE>
  </HEAD>
  <BODY>
    <H1>Üdvözöljük a vad motorosok honlapján!</H1>
    <P>Ez az oldal a dunántúli vad motorosok programjait tartalmazza.</P>
  </BODY>
</HTML>
```

Ha ezen az oldalon a H1 elemeket zölddel szeretnénk megjeleníteni, az alábbi CSS szabályra lesz szükség:

```
h1 { color: green }
```

Mint a példából is látszik, a CSS szabályok két fő részből állnak. Jelen esetben a h1 az egyik rész, ezt követi egy deklaráció (color: green). (A HTML-ben az elemnevek nem kisbetű-nagybetű érzékenyek, tehát a h1 helyett éppúgy használhatnánk a H1 megadást is.) Bár ebben a példában csak egyetlen tulajdonságot állítottunk a HTML dokumentum megjelenítéséhez, nyilván több lehetőségünk is van. Egy-egy elem több tulajdonságát is meghatározhatjuk, majd ugyanezt megtehetjük egy másik, egy harmadik ... elem tulajdonságaival is. így a sok stílus együttesen határozza meg a stíluslapot használó dokumentum végső megjelenését.

A HTML 4.0 specifikáció definiálja a stílusszabályok HTML dokumentumokban történő alkalmazásának két lehetőségét: a HTML dokumentumon belül (belső stíluslap, helyi stílus), illetve egy (.css kiterjesztésű) külső stíluslap-fájlon keresztül.

### 1.2.1.1. A belső stílus

Helyi stílusokat egy dokumentumban a STYLE elemmel hozhatunk létre. Az előző példánál maradva:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Vad motorosok</TITLE>
    <STYLE type="text/css">
      h1 { color: green }
    </STYLE>
  </HEAD>
  <BODY>
    <H1> Üdvözöljük a vad motorosok honlapján!</H1>
    <P> Ez az oldal a dunántúli vad motorosok programjait tartalmazza.</P>
  </BODY>
</HTML>
```

### 1.2.1.2. A külső stíluslap

A maximális rugalmasság elérése érdekében ma szinte minden honlapon alkalmaznak külső stíluslapo(ka)t. Előnyük, hogy anélkül módosíthatjuk egy honlap számos oldalának megjelenését, hogy magukba a HTML dokumentumok forráskódjába kellene nyúlni. Így ha menet közben gondoljuk meg magunkat, hogy nem tetszik a háttérszín vagy az aktív linkek színük miatt nem jól olvashatók, nem kell egy honlap összes dokumentumában egyesével módosítani, ami fáradságos, időigényes és nagy a hibalehetősége (még a fájlok mérete is nagyobb). A fenti előnyök miatt célszerű és erősen ajánlott a külső stíluslapok használata. Ehhez minden dokumentumban egy hivatkozást kell tenni a külső fájlra a LINK elemmel, az alábbi módon:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Vad motorosok</TITLE>
    <LINK rel="stylesheet" href="motoros.css" type="text/css">
  </HEAD>
  <BODY>
    <H1> Üdvözöljük a vad motorosok honlapján!</H1>
    <P> Ez az oldal a dunántúli vad motorosok programjait tartalmazza.</P>
  </BODY>
</HTML>
```

A LINK elem három dolgot határoz meg:

- a hivatkozás típusát: a "stylesheet" stíluslapot jelent
- a stíluslap elérési helyét a "href" attribútumon keresztül (ha ugyanabban a könyvtárban van a .css fájl, mint a HTML dokumentum, akkor itt csak a fájl neve szerepel, mint a fenti példában is)
- a hivatkozott stíluslap típusát: "text/css"

A következőkben helyi stílusokat fogunk használni (a STYLE elemmel), hogy láthassuk, milyen szoros a kapcsolat egy stíluslap és a strukturált HTML jelölések között. Határozzuk meg több elem színét:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Vad motorosok</TITLE>
    <STYLE type="text/css">
      body { color: black; background: white }
      h1 { color: green; background: white }
    </STYLE>
  </HEAD>
  <BODY>
    <H1> Üdvözljük a vad motorosok honlapján!</H1>
    <P> Ez az oldal a dunántúli vad motorosok programjait tartalmazza.</P>
  </BODY>
</HTML>
```

A stíluslap most négy szabályt tartalmaz. Az első kettő a BODY elem színét és háttérszínét határozza meg, ahol a black a fekete, a white a fehér színt jelöli (célszerű a szövegszínt és a háttérszínt együtt meghatározni). A 3. és 4. szabály a H1 elem színét és háttérszínét adja meg.

Mivel a bekezdéshez (a P elemhez) nem adtunk meg színt, azt a szülőelemétől, a BODY elemtől fogja örökölni. Bár a H1 is gyermekeleme a BODY-nak, a megadott szabály azonban felülbírálja az öröklött értékeket. A CSS-ben gyakran előfordulnak ehhez hasonló ellentmondások a különféle értékek között.

A 2.1-es CSS-nek 90-nél is több tulajdonsága van, a color (=szín) ezek közül csak az egyik. Nézzünk néhány más tulajdonságot is:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Vad motorosok</TITLE>
    <STYLE type="text/css">
      body {
        font-family: "Verdana", "Arial", sans-serif;
        font-size: 10pt;
        margin: 3em;
      }
    </STYLE>
  </HEAD>
  <BODY>
  </BODY>
</HTML>
```