

# XHTML és CSS

## XHTML és CSS

Webszerkesztés stílusosan

## A munkamegosztás a weblapon

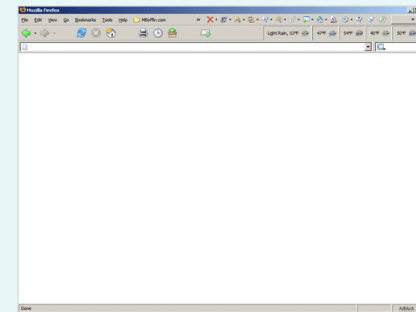
- **Tartalom** → XHTML (Extensible Hypertext Markup Language) a keresők is ezt „olvassák”!
- **Megjelenés** → stíluslapok CSS (Cascading Style Sheets) a design helye
- **Viselkedés** → Javascript használhatóság növelése

## Érvek 1. (Egy HTML-kód - sok CSS-lap)



(Forrás: [www.csszengarden.com](http://www.csszengarden.com))

## Érvek 2. (Egy HTML-kód - sok CSS-lap)



(Forrás: <http://mboffin.com/stuff/designline-openair.gif>)

# XHTML és CSS

5

## Érvek 3. (több HTML-kód - egy CSS-lap)

- Több lap stílusa egy helyen => gyorsan és könnyen frissíthető, módosítható
- Különböző felhasználóhoz különböző stílusokat lehet rendelni: például könnyen olvasható, nyomtatható stílusok
- A dokumentum mérete és komplexitása csökken áttekinthetőbb HTML kód  
kisebb HTML kód
- Stíluslapok a gyorsítótárban => gyorsabb oldalak

6

## XHTML nyelvtana

- Kezdőcímkéket és zárócímkéket elemek megadására használják  
`<p>Ez egy bekezdés</p>`
- jellemzők és értékek megadásával finomíthatjuk az elem megjelenését (kerülendő, CSS-ben formázunk!):  
`<p align="right">Ez a bekezdés.</p>`
- üres elemek pl.:  
`<br />`  
``

7

## XHTML: különleges követelmények

- DOCTYPE meghatározásra van szükség
- minden elemet, jellemzőt és értéket kisbetűvel kell írni
- minden értéket idézőjelben kell megadni
- minden jellemzőnek egyértelműen kell megadni az értékét

8

## XHTML sablon

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="hu" xml:lang="hu">
<head>
  <title>A lap címe</title>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2" />
</head>

<body>
  tartalom
</body>
</html>
```

# XHTML és CSS

9

## XML meghatározás

- A W3C azt javasolja, hogy adjunk meg egy XML meghatározást, mely tudatja a böngészővel, hogy XML változat következik, és hogy UTF-8 karakter kódolást használunk.  
`<?xml version="1.0" encoding="utf-8"?>`
- Az IE6 azonban ennek hatására trükköző üzemmódba vált. Mivel az XML előírások szerint az XML meghatározást nem kötelező használni, ezért, ha azt szeretnénk, hogy az IE szabványos üzemmódban működjön, jobb ha nem használjuk.

10

## DOCTYPE

- DOCTYPE: meg kell adni, melyik dokumentum típus meghatározást (Documentum Type Definition, DTD) kell használni a weblap megjelenítéshez.  
`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">`
- Lehetséges DOCTYPE változatok: *strict* (szigorú), *transitional* (átmeneti), *frameset* (keretkészletes)

11

## XML névtér

- XML meghatározás: a html elemet kombináljuk az XML névtérrel, amely az előírások helyének URL címét adja meg.  
`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">`  
`<html xmlns="http://www.w3.org/1999/xhtml" lang="hu" xml:lang="hu">`

12

## Karakter kódolás

- Ha a kiszolgáló nem állítja be a karakter kódolást, akkor egy meta elem segítségével megtehetjük:  
`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">`  
`<html xmlns="http://www.w3.org/1999/xhtml" lang="hu" xml:lang="hu">`  
`<head>`  
`<title>A lap címe</title>`  
`<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2" />`  
`</head>`
- A magyar nyelvhez az iso-8859-2, az angolhoz az iso-8859-1 kódolást kell használni.

# XHTML és CSS

13

## Hova tegyük a stílusokat?

- **Böngésző stílus:** minden böngésző saját stíluskészlettel rendelkezik. Ezek adják az alapértelmezett megjelenítési tulajdonságokat.
- **Felhasználói stílus:** a felhasználó által a böngészőben beállított saját stílusa elsőbbséggel bírhat a weblapkészítők által beállított bármely stílussal szemben.

14

## Hova tegyük a stílusokat? 2.

- **Külső stílus:** a külső stíluslap egy szöveges dokumentum, .css kiterjesztéssel. A hivatkozás elemet a fejlécben a title elem után helyezük el:

```
<title>oldal címe</title>
<link href="stíluslap.css" rel="stylesheet"
      type="text/css"/>
</head>
```

15

## Hova tegyük a stílusokat? 3.

- **Beágyazott stílus:** az XHTML fejlécébe style elembe kerülnek. Csak az adott dokumentumra érvényesek.

```
<title>weblap címe</title>
<style type="text/css">
<!--
p{
font: 0.9em Georgia;
color: #000;
}
-->
</style>
</head>
```

A régebbi böngészők miatt van a megjegyzésbe csomagolásra szükség.

16

## Hova tegyük a stílusokat? 4.

- **Szövegszint stílus:** az XHTML lapon szerepelnek, mint a formázandó elem jellemzői. Csak az adott elemre van hatással.

```
<p style="color: #00f">Ez a bekezdés egy weblapon
jelenik meg a böngésző által megvalósított
elrendezésben.</p>
```

A stílus a betűszínt kékre (#00f) változtatja, de csak ebben az egyetlen bekezdésben. Az új szabvány szerint kerülendő.

# XHTML és CSS

17

## Hova tegyük a stílusokat? 5.

- másik lehetőség: @import. Főleg a régi böngészők miatt alkalmazzák. Azokat az utasításokat rakják bele amit azok nem ismernek. A Netscape4. a @import utasítást átugorják.

```
<style type="text/css">
@import url(masikstilus.css);
</style>
```

Az IE6-ban ha nincs semmilyen link elem csak a @import, akkor a *formázatlan tartalom bevilanása* (flash of unstyled content, FOUC) hatás jön létre.

18

## Hova tegyük a stílusokat? 6.

- A link elemeket, az @import utasításokat és a beágyazott stílusokat kombinálhatjuk is.

```
<link rel="stylesheet" type="text/css"
href="egyikstilus.css" media="screen">
<link rel="stylesheet" type="text/css"
href="masikstilus.css" media="print">
<style type="text/css">
<!--
@import url(masstilus.css);
p {font-size: medium;}
-->
</style>
```

19

## Megjelenítési jellemzők

- A *media* megjelenítési jellemző lehetővé teszi, hogy adott eszközhöz szánt stílusokat írjunk. A legtöbb megjelenítési típust a böngészők még nem ismerik.
- A típusok: *all* (minden), *aural* (hangeszközök), *Braille* (Braille írásra alkalmas eszköz), *embossed* (domborított megjelenítés), *hanheld* (kézi eszközök), *print* (nyomtató), *projection* (kivetítő), *screen* (képernyő), *TTY* (szövegkijelzős telefon), *TV* (televízió).

20

## HTML elemek csoportosítása

- A HTML elemek két nagy csoportra oszthatók: **blokk** és **sor** elemekre.
- **Blokk**-elemek: pl. Hn, P, UL, stb. **DIV**  
Leegyszerűsítve: a blokk elemek előtt és után sortörés van
- **Sor**-elemek: pl. B, STRONG, CITE, stb. **SPAN**
- A két típusnál eltérően működik a dobozmodell!

# XHTML és CSS

21

## XHTML szerkezeti kódok

- **div**: a lapok logikai szakaszokra, un. tárolókra oszlanak. Ilyen szakaszok pl. a reklámcsíkok, a navigációs elemek, keresés, tartalom és láblécek.  
`<div id="search">ide jön a keresés tartalma</div>`
- **span**: amikor a formázni kívánt elem a szöveg belsejében helyezkedik el – például egy mondat néhány szava –, a span (kiterjedés) elemmel jelölhetjük ki a formázás határait.  
`<p>Költőink <span class="költő">Arany János </span> és a Lilla versekről ismert <span class="költő">Csokonai Vitéz Mihály</span>`

22

## XHTML szerkezeti kódok 2.

- **id**: azonosítja azt az elemet, amelyhez hozzá van rendelve. Egyedinek kell lenniük, azaz mindegyik csak egyszer szerepelhet a lapon. Betűvel, vagy aláhúzással kell kezdődniük, és nem tartalmazhatnak szóközt. Az *id* jellemző hiperhivatkozás célpontja is lehet.
- **class**: azokhoz az elemekhez használjuk amelyek nem egyediek – azaz többször is felhasználjuk a lapon. A *class* jellemzőhöz hozzárendelhető *div*, *span* vagy *p* elem, vagy *h2* címsor. Nem tartalmazhat szóközt és nem kezdődhet aláhúzással.

23

## Szövegközi formázások

- acronym (mozaik szó)  
`<p>A Weblapok formázása <acronym title="Hípertext Markup Language">HTML</acronym> szabályok szerint történik</p>`
- abbr (rövidítés)  
`<abbr title="Limited">Ltd. </abbr>`  
 csak az első előfordulásakor szoktuk megadni az XHTML2.0 –ból várhatóan az acronym kimarad.

24

## Szövegközi formázások 2.

- **cite**: címhivatkozások megjelölésére – pl. könyvek újságok címének kiemelésére szolgál.
- **blockquote**: idézetblokk – megjelenítéskor minden böngésző behúzza valamennyire az idézetblokkokat jobb és baloldalon, és többszintű elemként kezeli.
- **pre**: előreformázott szöveg – megtartja az eredeti dokumentum szerinti szóközöket, sortöréseket. Versek, hosszú kódblokkok megjelenítésére használják. Blokk szintű elem.

# XHTML és CSS

25

## Szövegek formázások 3.

- **em**: kihangsúlyozás általában dőlt, a hangos felolvasók is kihangsúlyozzák. Csak vizuális párja az `<i></i>`
- **strong**: erős kihangsúlyozás, általában félkövér betűvel jelenik meg. hangos felolvasók kihangsúlyozzák. Csak vizuális párja a `<b></b>`

Némelyik megjelenítő nem a megfelelően jeleníti meg az *em* és a *strong* elemeket ezért a stílus lapon előírhatunk rájuk vonatkozó stílus előírást.

26

## Szövegek formázások 4.

- **tt**: írógép betűk (typewriter)  
`<tt>szöveg</tt>`
- **code**: számítógépes kód (computer code)  
`<code>kód</code>`
- **kbd**: billentyűzet (keygoard)  
`<kbd>szöveg</kbd>`

mind a három elem szövegek formázási elem.

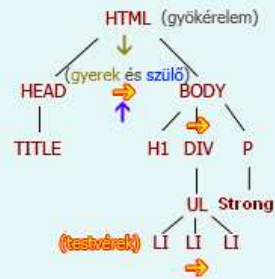
27

## Dokumentum fa - a weblap szerkezete

```

<html> - gyökérem
<head> - gyermek, és szülő (HTML gyereke)
<title> </title> - gyermek (HEAD gyereke)
</head>
<body> - gyermek és szülő (HTML gyereke)
<h1> </h1> - gyermek (BODY gyereke)
<div> - gyermek és szülő
  <ul> - a DIV gyereke és a LI szülője
    <li> </li>
    <li> </li>
    <li> </li>
  </ul>
</div>
<p>
  <strong> </strong>
</p>
</body>
</html>

```



28

## Dokumentum fa - fogalmak

- **Gyökérem**: Olyan elem melynek *nincs szülőelem*, de maga *szülőelem*, méghozzá a legfelsőbb. (Lásd: <HTML>)
- **Szülő**: Olyan elem melyeknek *van gyermekük*. Ez nem zárja ki azt, hogy ezek az elemek ne lehessenek maguk is gyermekek. (Lásd: a fenti példa UL-elemét.)
- **Gyermek**: Ha a *szülőket* A elemnek, a *gyermekeket* B-nek nevezzük, akkor a következőképp definiálhatjuk: a B elemet akkor nevezzük az A elem *gyermekének*, ha A elem a B elem *szülője*. Tehát olyan elemek, melyeknek *van szülőjük*.
- **Leszármazott**: a *gyerek* elemet nevezhetjük leszármazottnak is. Viszont leszármazott az az A elem is, mely *gyereke* C-nek, de C *gyereke* B-nek. Tehát B *leszármazotja* az A. Mint jómagad például a nagyszüleidnek.
- **Testvér**: Azok az elemek *testvérek*, melyek *egy szülőn osztoznak*. (Lásd: a fenti példa listájának LI elemei)

# XHTML és CSS

29

## CSS nyelvtana

- a CSS stíluslap lehet közvetlenül a HTML dokumentumban, vagy önálló fájlban kapcsolódhat hozzá
- a stílusokat *kijelölők* (selector) és *meghatározások* (declaration) segítségével lehet megadni:

```
p {
  font-family: Arial, Helvetica;
  font-size: small;
  color: blue;
}
```

30

## Azonosító kijelölők

- Az XHTML lapon:
 

```
<div id="labc" > szöveg </div>
```
- egy azonosítót egy XHTML lapon csak egyszer lehet használni, például: fejléc, lábléc, tartalomblokk vagy menü stb.

```
#labc {
  font-size: x-small; }
```

de környezet függő kijelölő:

```
#labc p {
  font-size: x-small}
```

31

## Osztály kijelölők

- HTML lapon:
 

```
<div class= "definico"> szöveg </div>
```
- osztálykijelölőket annyi példányban használhatjuk egy HTML lapon

```
.definico {
  background-color: silver; }
```

de környezet függő kijelölő:

```
#labc p.definico {
  background-color: gray; }
```

csoportosított kijelölő:

```
p, li, blockquote {
  font-style: gray; }
```

32

## Látszólagos kijelölők

- *:first-child*, amely az elemek első gyermekelemére vonatkozik. A következő stílusmegadás a *megj* elemcsoport (<div id="megj">) első bekezdését barna háttéren kék betűs szöveggé alakítja:

```
#megj p:first-child {color: navy;
  background: #987}
```

- Az utasítás csak akkor lesz értelmezve, ha az első elem ténylegesen P, és nem kerül végrehajtásra az elemek között később, de először előforduló P esetén. Ha a fenti meghatározásból kihagyjuk a P elemet, akkor az a *#megj* bármilyen típusú első gyermekelemére (pl. cím) fog vonatkozni.



# XHTML és CSS

33

## Látszólagos kijelölők 2.

- A `:first-line` pszeudo-elem segítségével a blokk szintű elemek első sorának tulajdonságait tudjuk befolyásolni. A felhasznált utasítás a bekezdések első sorát kék színűvé és nagybetűssé alakítja:
 

```
p:first-line {color: navy; text-transform: uppercase}
```
- Ez nem egy állandó terjedelmű szakaszt jelent: a szöveg első sorának hosszát mindig az aktuális megjelenítési körülmények határozzák meg.

34

## Látszólagos kijelölők 3.

- A `:lang` látszólagos osztály alkalmazásával eltérő formátumot rendelhetünk különböző nyelvű dokumentumokhoz, vagy akár egy dokumentum különböző nyelvű részeihez. Az alábbi stíluslap a megfelelő böngészőkben az oldal magyar nyelvűként meghatározott bekezdéseit (`<p lang=hu>`) zölden, az angolokat (`<p lang=en>`) pedig kék színnel, dőlt betűvel jeleníti meg:
- ```
:lang(hu) {color: teal}
:lang(en) {font-style: italic; color: navy}
```

35

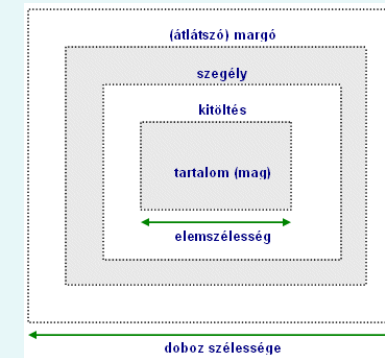
## Látszólagos kijelölők 4.

- A linkek esetében a CSS két pszeudo-osztályt vezetett be, ezek a `:link` és a `:visited`. Az első a még meg nem látogatott, a második pedig a már felkeresett linkeket jelenti. A `:hover` azokra az elemekre vonatkozik, amelyeket a felhasználó kijelölt (pl. fölévitte a kurzort), de nem aktivált. Az `:active` akkor lép életbe, amikor a felhasználó aktívvá tesz egy elemet, pl. az egérgomb lenyomása és felengedése közötti időben.
 

```
a {text-decoration: none}
a:active {color: red}
a:hover {font-weight: bold}
a:visited {color: gray; font-size: smaller}
```

36

## Dobozmodell



A CSS egy egyszerű, dobozszerű formázási modellt használ, ahol minden elemformázás eredménye egy, vagy több négyzetleges dobozként képzelhető el. Minden doboznak van egy 'magja', az őt körülvevő 'kitöltéssel' (padding), szegéllyel (border) és margóval (margin).

# XHTML és CSS

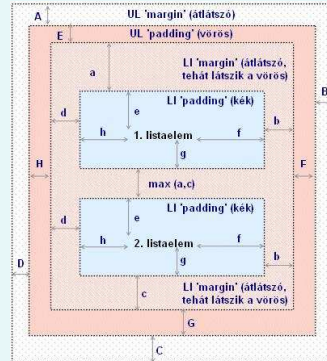
37

## Blokkszintű elemek

- ```

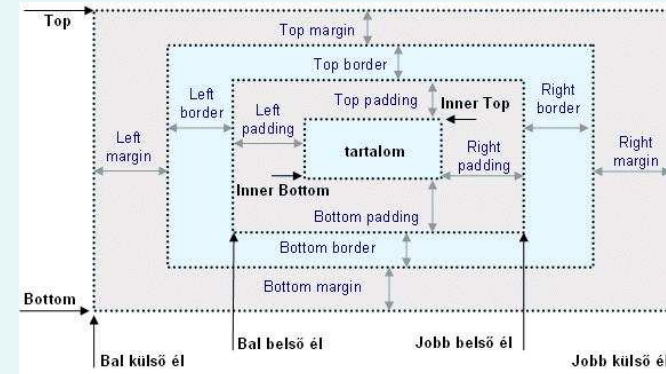
<style type="text/css">
ul {background: red;
margin: A B C D;
padding: E F G H;
}
li {color: white;
background: blue;
margin: a b c d;
padding: e f g h;
}
</style>
...
<ul>
<li>Első elem
<li>Második elem
</ul>

```



38

## Blokkelemek elnevezése



Forrás: <http://htmlspec.web.elte.hu/>

39

## Kódolás html-ben

```

<html>
<head>
<title>CSS példa</title>
</head>
<body>
<h1><font size="4" color="#a00000">Bevezetés</font></h1>
<p><font size="2" color="#00a000">A vers...</font></p>
<h1><font size="4" color="#a00000">Tárgyalás</font></h1>
<p><font size="2" color="#00a000">A költő...</font></p>
<h1><font size="4" color="#a00000">Összefoglalás</font></h1>
<p><font size="2" color="#00a000">Végezetül...</font></p>
</body>
</html>

```

40

## Kód CSS használattal

```

<html>
<head>
<title>CSS példa</title>
<style type="text/css"><!--
h1 { font-size: 20px; color: #a00000; }
p { font-size: 12px; color: #00a000; }
--></style>
</head>
<body>
<h1>Bevezetés</h1>
<p>A vers...</p>
<h1>Tárgyalás</h1>
<p>A költő...</p>
<h1>Összefoglalás</h1>
<p>Végezetül</p>
</body>
</html>

```


# XHTML és CSS

41

## CSS csatolási módok

```
<HTML>
<HEAD>
<TITLE>Stíluslapok - 1.1</TITLE>
<LINK REL=STYLESHEET TYPE="text/css"
HREF="pelda/stilus1.css" TITLE="sajat">
<STYLE TYPE="text/css">
@import url("pelda/stilus2.css");
<!--
H1 {color: blue}
-->
</STYLE>
</HEAD>

<BODY>
<H1>A címsor1 kék</H1>
<H2>A címsor2 piros</H2>
<P STYLE="color: green">Az egész bekezdés
zöld</P>
</BODY>
</HTML>
```



A fenti (egyszerű) példában használt stílus1.css állomány felel azért, hogy az oldal háttérszíne sárga legyen, a stílus2.css pedig a címsor szövegét állítja pirosra.

42

## Stíluslapok méretének csökkentése

- A szelektorok csoportosíthatók:
  - H1 {font-family: verdana}
  - H2 {font-family: verdana}
  - H3 {font-family: verdana}
- A csoportosítás után:
  - H1, H2, H3 {font-family: verdana}

43

## Stíluslapok méretének csökkentése 2.


- a deklarációk is csoportosíthatók:
  - H1 {font-family: helvetica}
  - H1 {font-size: 12pt}
  - H1 {font-style: normal}
- Most a tulajdonságokat csoportosítjuk:
  - H1 {
    - font-family: helvetica;
    - font-size: 12pt;
    - font-style: normal;
- Néhány tulajdonság eltérő csoportosítási szintaktikát is megenged:
  - H1 {font: bold 12pt helvetica}

44

## Példa

```
<HTML>
<HEAD>
<TITLE>Stíluslapok</TITLE>
<STYLE TYPE="text/css">
<!--
H1, H2, H3 {color: blue}
H1 {font: italic 22pt helvetica}
-->
</STYLE>
</HEAD>

<BODY>
<H1>Ez egy 1-es címsor</H1>
<H2>Ez egy 2-es címsor</H2>
<H3>Ez egy 3-as címsor</H3>
</BODY>
</HTML>
```



A fenti példában látható, hogy a csoportosítás segítségével beállítottuk, hogy a H1,H2,H3 címsor is kék legyen. A H1 címsor esetén a betűtípusra vonatkozó tulajdonságokat csoportosítottuk

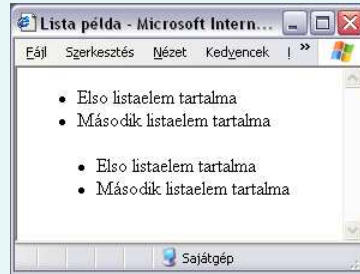
# XHTML és CSS

45

## Lista elemek

```
<HTML>
<HEAD>
<TITLE>Stíluslap példa</TITLE>
<STYLE TYPE="text/css">
<!--
  UL {list-style: outside}
  UL.compact {list-style: inside}
-->
</STYLE>
</HEAD>
<BODY>
<UL>
<LI>Első listaelem tartalma
<LI>Második listaelem tartalma
</UL>

<UL CLASS="compact">
<LI>Első listaelem tartalma
<LI>Második listaelem tartalma
</UL>
</BODY>
</HTML>
```



46

## Lebegő elemek



```
<html>
<head>
<title>Stíluslap példa</title>
<style type="text/css">
<!--
  img {float: left}
  body, p, img {margin: 25px}
-->
</style>
</head>
<body>
<p>

  Példaszöveg, melynek nincs más szerepe, minthogy bemutassa, hogy a lebegő elem a szülő elem szélére igazodik, miközben meg tartja annak margóját, szegélyét és kitöltését. Megjegyzendő, hogy a szomszédos függőleges margók összevonódnak a nem-lebegő blokkszintű elemek között
</p>
</body>
</html>
```

47

## Stílusok formátuma

- Egy-egy meghatározás két részből áll, egy kiválasztó és egy tulajdonság részből. A kiválasztó rész azt határozza meg, hogy mely HTML elemekre vonatkozzon a definíció, míg a tulajdonság rész a megjelenést befolyásolja.
  - `kiválasztó { tulajdonság }`
  - `kiválasztó { tulajdonság }`
  - `kiválasztó { tulajdonság }`

48

## Stílusok formátuma

- Az egyes definícióknak, de még a kiválasztónak és a tulajdonságnak sem szükséges új sorban lenniük, gyakorlatilag szabadon ránk van bízva, hogy a fenti tartalmat milyen elrendezésben valósítjuk meg. A következő formátumok mind helyesek:
  - `kiválasztó { tulajdonság } kiválasztó { tulajdonság }`
  - `kiválasztó { tulajdonság }`
- Több kiválasztót is meghatározhatunk, a kiválasztókat vesszővel kell elválasztani:
  - `kiválasztó, kiválasztó { tulajdonság; tulajdonság; }`

# XHTML és CSS

49

## Háttér formázás

background-color (háttérképszín)	<RGB>, <színnév>
background-image (háttérkép)	url(képnév.gif)
background-repeat (háttérkép ismétlődés)	repeat (ismétlődés) no-repeat (nem ismétlődjön) repeat-x (vizzintes ismétlődés) repeat-y (függőleges ismétlődés)
background-position (háttérképhelyzet)	top (felülre), center (középre), bottom (alulra), right (jobbra), left (balra), <százalék érték>, <hossz>
background-attachment (háttérképcsatolás)	scroll (gördülő), fixed (rögzített)

50

## Margó formázás

margin-top (felső margó)	<százalékérték>, <hossz>
margin-right (jobb margó)	<százalékérték>, <hossz>
margin-bottom (alsó margó)	<százalékérték>, <hossz>
margin-left (bal margó)	<százalékérték>, <hossz>
margin (margó)	<százalékérték>, <hossz>

összevont méret megadás:

```
margin: 1em 2em 3em 1em; /* top right bottom left */
margin: 1em 2em 3em; /* ha right=left */
margin: 1em 2em; /* ha top=bottom és right=left */
margin: 2em; /* ha top=bottom=right=left */
```

51

## Kitöltés formázás

padding-top (felső margó)	<százalékérték>, <hossz>
padding-right (jobb margó)	<százalékérték>, <hossz>
padding-bottom (alsó margó)	<százalékérték>, <hossz>
padding-left (bal margó)	<százalékérték>, <hossz>
padding (margó)	<százalékérték>, <hossz>

összevont méret megadás:

```
padding: 1em 2em 3em 1em; /* top right bottom left */
padding: 1em 2em 3em; /* ha right=left */
padding: 1em 2em; /* ha top=bottom és right=left */
padding: 2em; /* ha top=bottom=right=left */
```

52

## Keret formázás

border-width (szegélyvastagság)	thin (vékony), medium (közepes), thick (vastag), inherit (öröklés), <százalékérték>, <hossz>
border-style (szegélystílus)	none (nincs), hidden (rejtett), dotted (pontozott), dashed (szaggatott), solid (folytonos), double (dupla), groove bemélyített), ridge (kidomborodó), inset (süllyesztett), outset (kiemelkedő), inherit (öröklés)
border-color	<szín>, transparent (átlátszó), inherit (öröklés)
border	A vastagság, a stílus és a szín rövidítve is megadható

# XHTML és CSS

53

## Szöveg formázás

font-style (betű stílus)	italic (dőlt), oblique (ferde), normal (normál), bold (félkövér), inherit (öröklés)
font-variant (betűváltozat)	small-caps (kis kapilláris), normal, inherit
font-size	<százalékérték>, <hossz>
line-height (sormagasság)	<méret>
letter-spacing (betűköz)	<hossz>, normal, inherit
white-space	normal (sorvégeken tördel), nowrap (nem enged sortörést), pre (megtartja a sortöréseket, szóközöket, tabulátorokat)
word-spacing (szóköz)	<méret>

54

## Szöveg formázás

text-align (szövegigazítás)	left (balra), right (jobbra), center (középre), justify (sorkizárt), inherit
text-indent (szövegbehúzás)	<százalékérték>, <hossz>, normal, inherit
text-decoration	none (díszítetlen), underline (aláhúzott), overline (felülhúzott), line-through (áthúzott), blink (villogó), inherit
text-transform	none (módosíthatlan), capitalize (első betű nagy), uppercase (összes betű nagy), lowercase (összes betű kicsi)
text-shadow	5px 5px 3px #000000; lefele való távolság, jobbra való távolság, elmosottság sugara, szín.