

## (statikus) (X)HTML oldalak, stíluslapok

## Áttekintés

- ▶ A HTML története
- ▶ (X)HTML oldal felépítése
- ▶ Egymásba ágyazható stíluslapok – CSS

## Mi a HTML?

### HTML (HyperText Markup Language) - Hiperszöveges jelölőnyelv

- ▶ a World Wide Web által használt egységes nyelv:
  - ▶ online dokumentumok létrehozása, címeikkel, szöveggel, táblázatokkal, fotókkal, stb.,
  - ▶ online tartalom böngészése hiperszöveg hivatkozásokon (hypertext link) keresztül,
  - ▶ form-ok (űrlapok) használata, távoli számítógépek által nyújtott szolgáltatások igénybevételére, mint például információ keresése, termékek megrendelése, stb.,
  - ▶ videók, zene/hang és más multimédia-elemek csatolása a dokumentumokhoz

## A HTML története

- ▶ 1990-ben Tim Berners-Lee, a WEB kezdeményezője alkotta
- ▶ az SGML (Standard Generalized Markup Language, ISO 8879) sablonjára íródott (SGML – jelölő nyelvek definiálását lehetővé tevő rendszer)
- ▶ p, h1–h6, ul, ol – kezdettől fogva benne voltak, a hiperlinkek használata – saját ötlete
- ▶ lelkesedők csapata vitatja a további fejlesztési lehetőségeket
- ▶ a böngésző-fejlesztők saját elképzeléseik szerint implementálnak újabb tag-eket (pl. Mosaic– IMG tag)
- ▶ IETF (Internet Engineering Task Force) – létrehozott egy HTML munkacsoportot (HTML Working Group) → 1995 HTML 2.0 (az első kísérlet a HTML szabványosítása irányában)

## A HTML története

- ▶ 1996-tól a **World Wide Web Consortium** (W3C) vállalja át a felelősséget az újabb HTML specifikációkért (utoljára kiadott szabványuk –1999 HTML 4.01)
- ▶ 1997: HTML 4.0 (ennek apró javításokat tartalmazó változata a HTML 4.01) <http://www.w3.org/TR/html401/>
  - ▶ egyes tag-eket “elavult”-nak minősít
  - ▶ három dokumentum típus:
    - **Strict** (szigorú specifikáció, elavult tag-eket nem enged használni)
    - **Transitional** (átmenet a régebbi használt stílus és az újabb, szigorú között)
    - **Frameset** (frame-eken alapuló oldalak)
- ▶ 2000-ben nemzetközi standarddá válik (ISO HTML, a HTML 4.01 Strict-re alapoz)
- ▶ 2000 – XHTML specifikáció (a HTML 4.01 specifikáció újrafogalmazása XML-re alapozva)– W3C
- ▶ 2008 – HTML 5 – Working Draft (<http://www.w3.org/TR/html5/>)

## Egy HTML dokumentum felépítése

### HTML elemek és tag-ek

- ▶ **elem**: a HTML egy részét leíró struktúra
  - ▶ részei: *kezdő tag*, *törzs (content)*, *befejező tag*
- ▶ **tag**: egy szövegrész “megjelölését” szolgálja
  - ▶ “<” és “>” határolja, befejező tag esetén pedig “< /” és “>”

<em>ez egy kiemelt szövegrész</em>

- ▶ kis- vagy nagybetűk közt nincs különbség (<em>, <eM> vagy <EM> ugyanazt jelenti)
- ▶ a tag-ek nem keresztezhetik egymást (pl. <p><em>szöveg</p></em> - helytelen) ... a böngészők viszont elnézőek
- ▶ egyes tag-ek esetén a befejező tag (elvileg) elhagyható (pl. felsorolásnál li)
- ▶ üres tag-ekhez nincs külön befejező tag: br, img

## Egy HTML dokumentum felépítése

### Attribútumok

- ▶ egy elem attribútumai az illető elem különböző *tulajdonságait* határozzák meg  

- ▶ a kezdő tag-ben adjuk meg őket
- ▶ alakja: attribútum-név="attribútum-érték"
- ▶ az attribútum értéket “,” vagy ‘,’ határolja (nem **kötelező**, amennyiben az érték csak betűket, számjegyeket, illetve “-” vagy “.” karaktereket tartalmaz, de mégis **ajánlott**)
- ▶ az attribútum nevében a kis-/nagybetű nem számít, az értékben viszont számíthat.

## Egy HTML dokumentum felépítése

### Speciális karakterek:

- ▶ ha speciális karaktereket vagy a jelölő nyelv számára fenntartott egyes speciális karaktereket szeretnénk megjeleníteni, ezeknek egy sajátos kódját kell használnunk:  
“<” – &lt; ; “>” – &gt; ;  
szóköz – &nbsp; ;  
“&” – &amp; ;  
” – &quot; ;
- ▶ egy másik lehetőség – (Unicode szerinti) numerikus kód használata:  
&#8220; – “  
&#8221; – ”  
&#169; – ©

referencia (pl.): <http://htmlhelp.com/reference/html40/entities/>

### Komment

- ▶ kezdete “<!-- –”, vége “-->” (ne legyen benne “-- –”)  
<!-- ez egy megjegyzés -->

## Egy HTML dokumentum felépítése

### Teljes HTML oldal alkotóelemei:

- ▶ **DOCTYPE** – megadja azt a HTML verziót, amelynek a dokumentum megfelel
- ▶ **html**
  - ▶ **head** – információ a dokumentumról (pl. cím, karakterek kódolása)
  - ▶ **body** – a dokumentum tulajdonképpeni tartalmát foglalja magába

### HTML oldal érvényességének vizsgálata

- ▶ a böngészők “elnézőek” – azaz a *maguk módján* próbálják megjeleníteni a hibás dokumentumot
- ▶ érdemes leellenőrizni a dokumentum helyességét – megfelel-e egy adott specifikációnak  
pl. online ellenőrzés: WDG HTML Validator (<http://www.htmlhelp.com/tools/validator/>),  
W3C ellenőrző oldala (<http://validator.w3.org/>)

## Egy HTML dokumentum felépítése

pl.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>A dokumentum címe</title>
  </head>
  <body>
    <h1>Cím</h1>
    <p>Egyik bekezdés.</p>
    <p>Másik bekezdés.</p>
    <ul>
      <li>Felsorolás I. eleme.</li>
      <li>Felsorolás II. eleme.</li>
    </ul>
  </body>
</html>
```

## Fontosabb HTML tag-ek

### head-en belül:

- ▶ **title** – a dokumentum címe (a szigorú spec. szerint kötelező)
- ▶ **meta** – különböző (meg nem jelenített) információ a böngésző illetve keresőmotorok számára
  - ▶ Példák – lásd: [meta\\_tag\\_uresHTML.html](#)
- ▶ **link** – Jelzi a dokumentum kapcsolatát más dokumentumokkal (pl. stíluslappal)  
attribútumok:
  - ▶ href="url"
  - ▶ media="screen" — "print" — "all" – meghatározza a dokumentum kimenetét

## Fontosabb HTML tag-ek

### body-n belül:

- ▶ **h1** -tól **h6**-ig – főcím, alcím, stb.
- ▶ **p** – bekezdés (paragrafus)
- ▶ **br** – újsor (üres)
- ▶ **pre** – megőrzi a fehér karaktereket
- ▶ **hr** – vízszintes vonal (üres)
- ▶ szövegformázó tag-ek:
  - ▶ **b** – vastagított
  - ▶ **em** – szövegrész kiemelése
  - ▶ **strong** – hangsúlyozottabb szöveghiemelés (vastagított betűvel)
  - ▶ **i** – dőltbetűs szöveg
  - ▶ **code** – állandó betűszélességű karakterkészlet, tipikusan programkód beszúrása
- ▶ **sub**, **sup** – alsó-, felső index
- ▶ **a** – horgony (anchor) – lásd: [anchor.html](#)

## Fontosabb HTML tag-ek

body-n belül:

- ▶ táblázat (**table**, **tr**, **th**, **td**)
- ▶ felsorolás (nem számozott: **ul**, **li**, számozott: **ol**, **li**)
- ▶ **img** – képbeszúrás
- ▶ **div** – elhatárol egy részt a dokumentumon belül
- ▶ **span** – HTML elemeket csoportosít (pl. egységes stílus)

elemek típusa (a megjelenítés szempontjából):

- ▶ **blokk**-típusú elem: kihasználja a teljes rendelkezésre álló szélességet; előtte és utána sortörés. **pl.** p, div, h1-h6, ul, ol, li
- ▶ **inline** elem: az aktuális sorban jelenik meg, csak annyi szélességet foglal el, amennyi szükséges, nincs sortörés. **pl.** a, em, span, i, b

Különböző HTML elemekre vonatkozó leírás illetve példák:

HTML\_elemek.html

## Attribútumok:

Standard attribútumok:

- ▶ *class* – az elemet egy (stílus)osztályba sorolja
  - ▶ *id* – egyedi azonosítót rendel az elemhez
  - ▶ *style* – inline stílusdefiníció
  - ▶ *title* – helyi súgó (tooltip)
- 
- ▶ különböző tag-ek sajátos attribútumai – lásd referenciák

## Referenciák (HTML4.01)

HTML elemek:

- ▶ W3C - index
- ▶ w3schools - HTML 4.01 / XHTML 1.0 referencia

attribútumok

- ▶ W3C - attribútumok indexe

## HTML form(űrlap)-elemek

Form elemek:

- ▶ lehetővé teszik, hogy a felhasználó adatokat vigyen be
- ▶ form adatok küldése: submit segítségével (a *form* tag *action* attribútumába megadott URL-re)
- ▶ ezen adatok feldolgozása (általában) szerver oldalon történik

Példák:

lásd: formelemek.html

## Frame-ek (lapkeretek)

- ▶ a frame-ek használata lehetővé teszi, hogy egyetlen böngésző ablakban több HTML oldalt jelenítsünk meg
- ▶ mindenik HTML oldalt frame-nek hívunk, és ezek függetlenek egymástól

### Frame-k használatának hátrányai:

- ▶ egyszerre több HTML oldalt kell nyilvántartania a fejlesztőnek, ami nehézkes lehet
- ▶ nehéz kinyomtatni a teljes weboldalt

### Példák:

lásd: vizszintes\_framek.htm, vegyes\_framek.htm, navigalo\_pl.htm, iframe-et\_hasznalo\_oldal.html

## XHTML

- ▶ XHTML – EXtensible HyperText Markup Language
- ▶ a cél, hogy átvegye a HTML szerepét
- ▶ szinte azonos a HTML 4.01-val (árnyalatbeli különbségek)
- ▶ a HTML egy tisztább, szigorúbb változata
- ▶ az XHTML gyakorlatilag a HTML-nek érvényes XML-ként való meghatározása
- ▶ a W3C tette közzé

### a legfontosabb különbségek:

- ▶ az XHTML elemeket kötelező módon helyesen kell egymásba ékelni
- ▶ az XHTML elemeket mindig be kell zárni.  
Üres elem esetén: pl. `<br />`
- ▶ az XHTML elemeket és attribútumokat kisbetűvel kell írni
- ▶ az attribútumok értékét kötelező idézőjelbe tenni

## XHTML vagy HTML?

### miért/mikor érdemesebb XHTML-t használni

- ▶ a HTML szintaxisa kevésbé szigorú → laza, felületes szerkezet
- ▶ a HTML-t megjelenítő böngészők még a HTML szintaxisára vonatkozó szabályok betartására sem köteleznek
- ▶ egy XHTML dokumentum szintaktikai helyessége ellenőrizhető

### XHTML verziók

- ▶ XHTML 1.0 (W3C ajánlás, 2000. jan., 2002. aug.)
- ▶ XHTML 1.1 – modularizált XHTML (W3C ajánlás: 2001, 2010)
- ▶ XHTML 1.2 – abbamaradt...
- ▶ XHTML 2.0 – nem vált W3C ajánlássá (W3C jegyzetek, 2009)
- ▶ XHTML5 – dolgoznak rajta

## XHTML

### egy minimális XHTML dokumentum

```
<?xml version = "1.0" encoding = "utf-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
<title>Cím</title>
</head>

<body>
</body>

</html>
```

# (X)HTML5

## irányvonalak

- ▶ az újabb elemek alapja: HTML, CSS, DOM és JavaScript
- ▶ külső plug-in-ok használatának szükségességét csökkenteni
- ▶ jobb hibakezelés
- ▶ új elemek (ahelyett, ami addig csak szkripttel volt megoldható)

## néhány újdonság

- ▶ *canvas* elem (rajz készítésére)
- ▶ A *video* és *audio* elemek
- ▶ helyi (a kliens gépen történő) adattárolás
- ▶ új tartalom-specifikus elemek (*article*, *footer*, *header*, *nav*, *section*)
- ▶ új form-elemek (*calendar*, *date*, *time*, *email*, *url*, *search*)

## Böngésző támogatottság - néhány HTML5 nyújtotta lehetőség esetén

- ▶ Safari, Chrome, Firefox, Opera és IE legújabb verziói



21 / 1

# CSS - Cascading Style Sheets

- ▶ leírnyelv, melynek segítségével különböző stíluslapokat hozhatunk létre és ágyazhatunk be HTML oldalakba
- ▶ az egyes HTML tag-ek megjelenítési stílusát határozzák meg (méret, szín, stb.)
- ▶ HTML 4.0-hoz – a HTML standadizálásáért felelős World Wide Web Consortium (W3C) hozta létre
- ▶ a legtöbb mai böngésző támogatja a CSS-t
- ▶ verziók: CSS1 (1996), CSS2 ('98), CSS3 (fejlesztés alatt)

## előnyei:

- ▶ egységes stílust biztosít a HTML dokumentumoknak (pl. ugyanazon honlapon, web-alkalmazáson belül)
- ▶ rövidebbé, átláthatóbbá teszi a HTML oldalakat
- ▶ megjelenítési stílus és tartalom szétválasztása
- ▶ rugalmasság (elég a stílusállományt módosítani)



22 / 1

## CSS - stílus megadása különböző szinteken

### 1. Stílus megadása inline módon

- ▶ a stílus a tag-en belül van megadva

```
<p style="itt van megadva a stílus">
```

### 2. Belső stíluslap

- ▶ a stílust a dokumentum *head* elemében adjuk meg

```
<style type="text/css">  
    stílus elemek  
</style>
```



23 / 1

## CSS - stílus megadása különböző szinteken

### 3. Külső stíluslap

- ▶ a stílust külön .css kiterjesztésű állományban adjuk meg, melyet a dokumentum head elemébe "csatolunk" (MIME típus: text/css)

```
<link href="stílusfilenev.css" rel="stylesheet"  
type="text/css">
```

- ▶ másik alternatíva:

```
@import url('allomanynev');
```

(a style elem legelején kell szerepeljen, régi böngészők nem ismerik fel)

### a külső stíluslap helyessége ellenőrizhető:

<http://jigsaw.w3.org/css-validator/>



24 / 1

- ▶ egy HTML elem stílusa különbözőképpen (a leírt 3 szinten) adható meg, ezek végül egy egységes "virtuális" stílussá alakulnak (innen a "cascading" elnevezés)

### prioritási sorrend:

- ▶ inline módon megadott stílus (legnagyobb prioritású)
- ▶ belső stíluslap (a head elemben megadott)
- ▶ külső stíluslap
- ▶ böngésző alapértelmezés szerint meghatározott stílusa

### Szintaxis:

```
szelektor { tulajdonság1:érték1;
            tulajdonság2:érték2; ... }
pl. body {color: black;}
p {font-family: "sans serif";}
```

### Az id szelektor #

- ▶ adott id attribútumú elem stílusa  
#13fejezet {color: green;} → <h2 id="13fejezet">ez zöld lesz </h2>

### Pseudo-szelektorok – :link :visited :hover :active :focus

```
a:link { color: red; background:#000000; }
a:visited { color: blue; background:#000000; }
a:hover { color: green; background:#000000; }
a:active { color: lime; background:#000000; }
```

- ▶ többféle hivatkozás-stílus: a.elseo:link, a.masik:link, stb.

### További (részletes) információ a szelektorokról

- ▶ CSS2 <http://www.w3.org/TR/CSS2/selector.html#selector-syntax>
- ▶ CSS3 <http://www.w3.org/TR/2009/PR-css3-selectors-20091215/#selectors>

### egyszerű szelektorok:

- ▶ **elemnev:** pl. p {text-align:center; color:red;}
- ▶ a szelektorok csoportosíthatók: pl. h1,h2,h3 {color: green;}
- ▶ kontextussal megadott szelektor: pl. ol ol li

### A class szelektor– stílusosztály

- ▶ ugyanannak a HTML elemnek különböző stílusokat adhatunk meg  
p.jobb {text-align:right;} → <p class="jobb" >szov</p>  
p.bal {text-align:left;} → <p class="bal" >szov</p>
- ▶ általános stílusosztály: több különböző elemnek adhatjuk ugyanazt a stílust  
.piros {color:#AA0000;} → <p class="piros" >szov</p>  
<em class="piros" >szov</em>

### bizonyos attribútummal rendelkező elem stílusa

```
input[type="text"] {background-color: blue;}
```

## CSS tulajdonságok

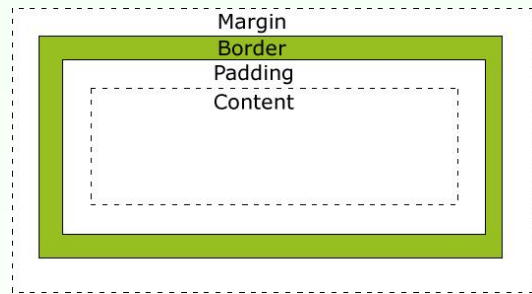
### megjegyzés:

```
/* ez itt CSS megjegyzés */
```

### hátter, háttérszín, szövegszín

- ▶ háttérszín és szövegszín: body { **background**:#00ff00; **color**:#ffffff; }
- ▶ háttérkép: body { **background-image**: url("kep.gif"); }
  - ▶ fix háttérkép:  
body { background-image: url("kep.gif"); **background-attachment**: fixed; }
  - ▶ szöveggel gördülő háttérkép (alapértelmezett)
  - ▶ kép ismétlése x, y, x-y tengely mentén:  
**background-repeat**: repeat-x, repeat-y, (alapértelmezett)
- ▶ szöveg színe megadható numerikusan (color : #AABB00), szimbolikusan (color: red) vagy RGB komponensek segítségével (color:rgb(255,0,0))

Szél (**margin**), behúzás (**padding**), szegély (**border**) (W3C specifikációja):



használható mértékegységek:

**em** (a releváns betűtípus mérete), **ex** (a releváns betűtípus x-magassága), **px** (pixel), **in** (inch), **cm** (centiméter), **mm** (milliméter), **pt** (pont), **pc** (= 12 pont)

## Megjelenítés

megjelenítés (**display**) módosítása:

display: inline;  
display: block;

láthatóság (**visibility**):

- ▶ display: none;
- ▶ visibility: hidden;

## Betűtípus

- ▶ betűtípus család (font-family)  
p { **font-family**: Verdana, Arial, 'Times New Roman', serif }  
általános típusok: serif, sans-serif, cursive, fantasy, vagy monospace
- ▶ betű stílus (**font-style**): *italic*, *normal*
- ▶ betűvastagság (**font-weight**): *normal*, *bold*, *bolder*, *lighter*, *100*, ..., *900*
- ▶ betűméret (**font-size**): abszolút érték (pl. x-small, small, medium, large), relatív (pl. larger, smaller ), hossz, százalék

szövegigazítás (**text-align**)

- ▶ lehetséges értékek: left, right, center, justify

táblázatok, listák

- ▶ lásd: 2\_tabla\_CSS-el.html
- ▶ lásd: 3\_lista\_CSS-el.html

## Pozíció megadása

pozíció megadásának különböző módjai (**position** tulajdonság):

- ▶ statikus (static) – alapértelmezett
  - ▶ fix (fixed) – elhelyezés a böngészőablakhoz képest (a görgetés sem befolyásolja az elhelyezést)
  - ▶ relatív (relative) – a normál elhelyezéshez viszonyított relatív elhelyezés
  - ▶ abszolút (absolute) – az első nem statikus pozíciójú szülőelemhez viszonyított elhelyezés
- ▶ pozíció megadása: top, right, bottom, left tulajdonságok segítségével

**z-index**

- ▶ egymást átfedő elemek esetén megadható a megjelenítés sorrendje
- ▶ nagyobb index-értékű elemek a kisebb index-értékűek fölött helyezkednek el



## float/clear tulajdonságok

### a **float** tulajdonság segítségével

- ▶ jobb vagy baloldalra tolhatunk egy elemet
- ▶ az utána következő elemek körülötte fognak elhelyezkedni

### tipikus használat

- ▶ szöveg, illetve kép(ek) egymás mellé rendezése
- ▶ weboldal tartalmának részekre bontása, és ezek elhelyezése (táblázat használata nélkül)

### Példa:

lásd: 4\_float\_pl.htm

## Konfliktushelyzetek kezelése

### Stílusdefiníció forrásai:

- ▶ szerző
- ▶ böngésző
- ▶ felhasználó

### Stílusjegy fontossága

- ▶ normál
- ▶ fontos (`!important`)

### Prioritási sorrend konfliktus esetén:

1. a felhasználó fontos beállításai (legnagyobb prioritás)
2. a szerző fontos beállításai
3. a szerző normál beállításai
4. felhasználó normál-beállításai
5. böngészőből származó beállítások

### Specifikusság szerinti sorrend (továbbra is fennálló konfliktus esetén):

(a prioritás csökkenő sorrendjében)

1. id szelektor
2. stílusosztály illetve pseudo-szelektorok
3. kontextussal megadott szelektorok (minél több elemtípus szerepel, annál specifikusabbnak számít)
4. univerzális szelektor (\* - minden elemre vonatkozik)

### amennyiben továbbra is konfliktus van:

- ▶ a későbbi felülírja a korábban definiált stílusjegyet

### Hasznos hivatkozások:

- ▶ CSS referencia: [http://www.w3schools.com/CSS/css\\_reference.asp](http://www.w3schools.com/CSS/css_reference.asp)
- ▶ CSS specifikációk:  
<http://www.hungarian-webstyles.com/Style/CSS/#specs>