

(statikus) HTML (XHTML) oldalak, stíluslapok

# Áttekintés

- A HTML története
- HTML oldal felépítése
- Egymásba ágyazható stíluslapok – CSS

# Áttekintés

- A HTML története
- [HTML oldal felépítése](#)
- Egymásba ágyazható stíluslapok – CSS

# Áttekintés

- A HTML története
- 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, címekkel, 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
- Dan Connolly írta meg hozzá 93-ban a formális specifikációt (DTD)
- az SGML (Standard Generalized Markup Language, ISO 8879) sablonjára  
(SGML – jelölő nyelvek definiálását lehetővé tevő rendszer)
- p, h1–h6, ul, ol – már kezdetől fogva benne voltak, a hiperlinkek használata – saját ötlete
- lelkesedők csapata vitatja a további fejlesztési lehetőségeket
- Dave Raggett (Bristol-i Hewlett-Packard Lab.) – HTML+ (az eredeti HTML egy kibővített változata)
- 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 régebb 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
- **meta** – különböző (meg nem jelenített) információ a böngésző illetve keresőmotrok 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** -től **h6**-ig – főcím, alcím, stb.
- **p** – bekezdés (paragrafus)
- **br** – újsor (üres)
- **hr** – vízszintes vonal (üres)
- szövegformázó tag-ek:
  - **b** – vastagított
  - **em** – szövegrész kiemelése
  - **i** – dőltbetűs szöveg
- **a** – horgony (anchor) – lásd: [anchor.html](#)
- 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)

# 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:

- <http://www.w3.org/TR/html401/index/elements.html>
- <http://www.w3schools.com/tags/default.asp>

## attribútumok

- <http://www.w3.org/TR/html401/index/attributes.html>



# 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: [4\\_vizszintes\\_frameek.htm](#), [4\\_vegyes\\_frameek.htm](#), [4\\_navigalo\\_pl.htm](#)

# 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 kisbetűvel kell írni
- az attribútumok értékét kötelező idézőjelbe tenni

# XHTML

## egy minimális XHTML dokumentum

```
<!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>
```

# 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 és Opera legújabb verziói, IE9 (közeljövőben)

# CSS - Cascading Style Sheets

- leírónyelv, 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

## 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)

# CSS

## Stílus megadása inline módon

- a stílus a tag-en belül van megadva

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

## Belső stíluslap

- a stílust a dokumentum *head* elemében adjuk meg

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

## Külső stíluslap

- a stílust külön .css kiterjesztésű állományban adjuk meg, melyet a dokumentum head elemébe "csatolunk"

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

# CSS

- egy HTML elem stílusa többféleképpen adható meg, ezek végül egy egységes “virtuális” stílussá alakulnak (innen a “cascading” elnevezés)

a különbözőképpen megadott stílusok egységesítése a köv. prioritási sorrendben történik:

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

Szintaxis:

```
szelektor {tulajdonság: érték;}  
pl. body {color: black;}  
p {font-family: "sans serif";}  
p {text-align:center; color:red;}
```



## a szelektorok csoportosíthatók:

```
h1,h2,h3,h4,h5,h6  
{  
color: green;  
}
```

## 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>`
- 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;}
```

## Az ID szelektor #

- adott ID attribútumú elem stílusa  
`#unicum {color: green;} → <p id="unicum">ez zöld lesz</p>`

## Pseudo/ szelektorok – a:link a:visited a:hover a:active

```
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>

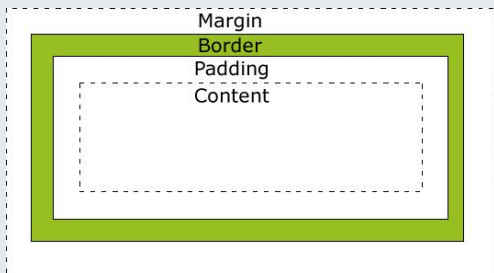
## megjegyzés:

```
/* ez itt CSS megjegyzés */
```

## háttér, 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)

## 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

# Megjelenítés

## blokk-típusú, illetve inline elemek

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

## megjelenítés módosítása:

```
display: inline;  
display: block;
```

## láthatóság:

- display: none;
- visibility: hidden;

# 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

## CSS referencia:

[http://www.w3schools.com/CSS/css\\_reference.asp](http://www.w3schools.com/CSS/css_reference.asp)