

# Alapfogalmak a Web-fejlesztés téma körben

Web-fejlesztés I. kurzus

Dr. Abonyi-Tóth Andor  
Egyetemi adjunktus  
ELTE Informatikai Kar

# **Szabványok és ajánlások**

# Szabvány vagy ajánlás?

- A Web-es világban szabványok (*standard*) és ajánlások (*recommendation*) alapján fejlesztünk.
- A Web-es jelölőnyelvekre vonatkozó ajánlásokat a W3C konzorcium dolgozta ki.
  - Az ajánlásokra - azok széles körű elterjedése miatt - (de-facto) szabvánnyként kellene, hogy tekintsenek a web-fejlesztők.

# De-facto ás de-jure szabványok

- A **de-facto** szabványok elterjedését egy-egy konkrét megoldás széleskörű használata biztosítja.
- Ahhoz, hogy a de-facto szabványból, hivatalos szabványügyi testület által rögzített **de-jure** szabvány válhasson, azt be kell nyújtani a megfelelő szabványügyi testülethez, amelynek el kell fogadnia azt.
  - Erre a W3C dokumentumok között több precedenst is találhatunk (pl. ECMAScript, WCAG 2.0).

# Kivétel ☺

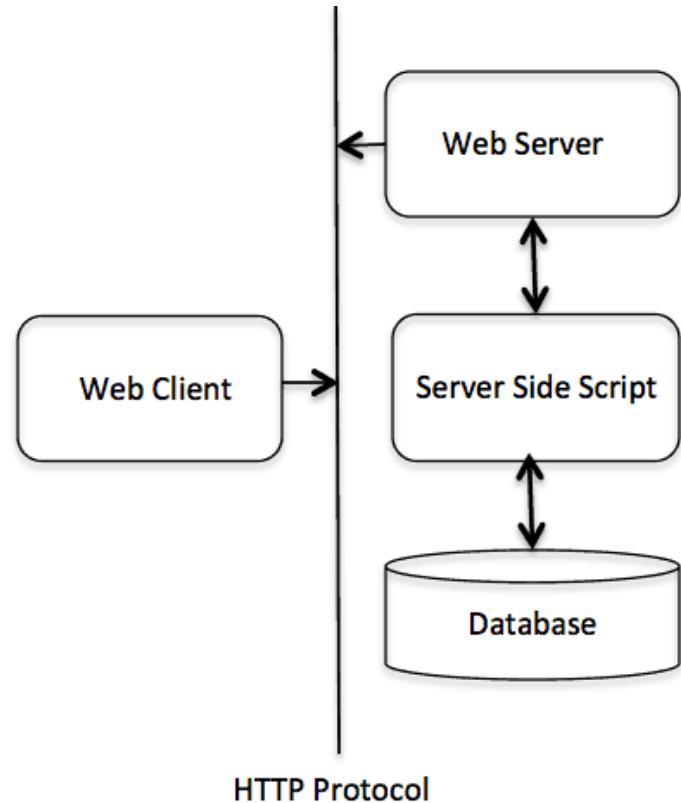
- Csak akkor nem kell a Web-es szabványok tartalmát figyelembe venni, ha Te vagy Chuck Norris.



„Chuck Norris nem veszi figyelembe Web-es szabványokat, mivel a Web igazodik Őhozzá.”

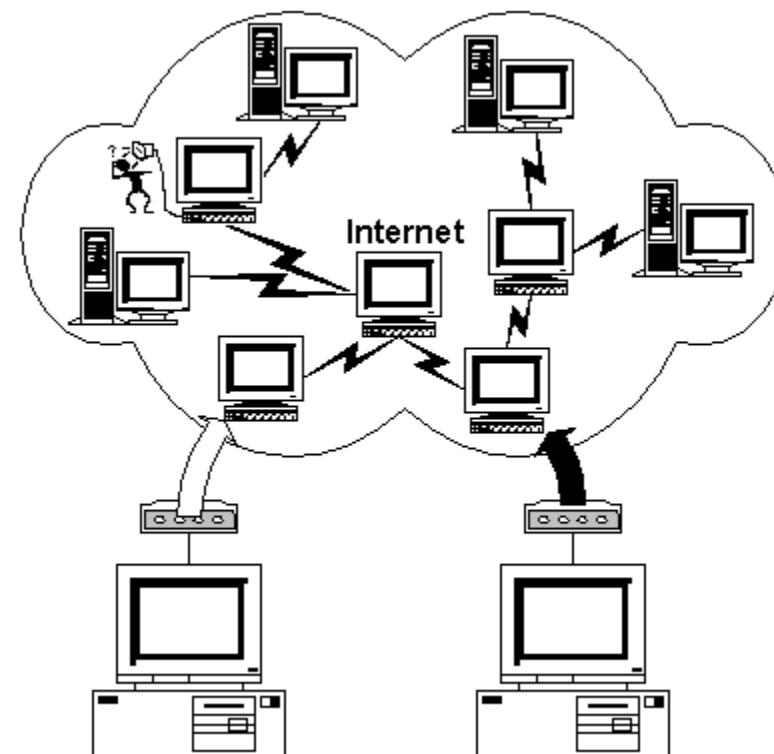
# HTTP protokoll

- Protokoll = szabályrendszer
- HTTP
  - HyperText Transfer Protocol (Hiperszöveg Átviteli Protokoll)
- A webböngésző (kliens) adatokat kérhet le a kiszolgálóról (szerver), illetve adatokat küldhet a kiszolgálóra.
- A kérést a kliens kezdeményezi.



# HTTPS protokoll

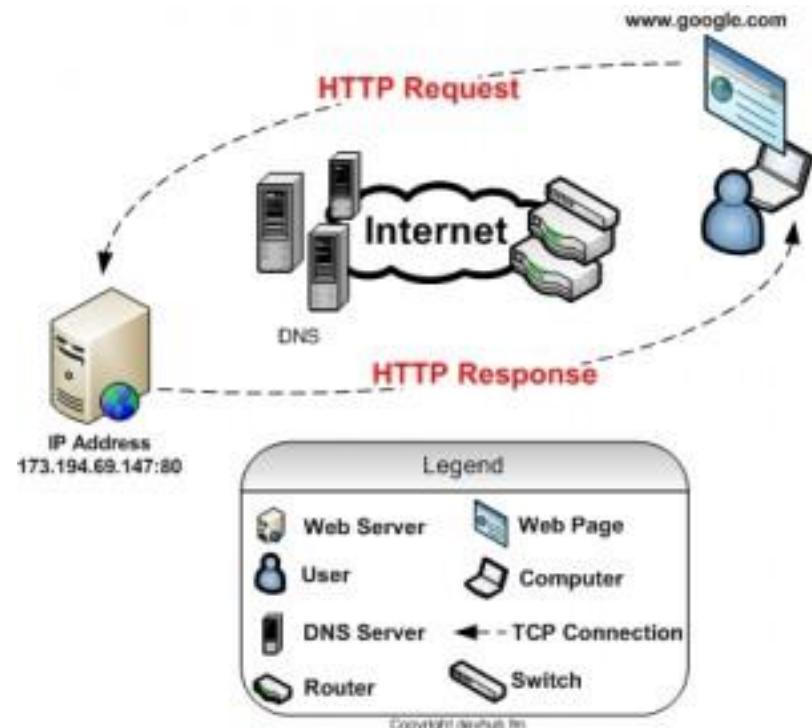
A https séma szintaktikailag megegyezik a http sémával, de jelzi a böngészőnek, hogy használni kell az SSL/TSL titkosító réteget az adatforgalom védelme érdekében



# HTTP kérés és válasz

1. A felhasználó ellátogat egy weboldalra
2. Ez egy kérést indikál (HTTP request)
3. A webszerver megkapja a kérést, és elküldi a választ (HTTP response)

Kérést nem csak webböngészővel küldhetünk, egyéb kliens alkalmazásokkal is (pl. Java, C# kliens)



<http://devhub.fm/http-requestresponse-basics/>

# HTTP kérés (request)

- 1. sor „METÓDUS ERŐFORRÁS VERZIÓ” alakú
  - GET /images/logo.gif HTTP/1.1
- Ezt követik a fejléc (header) sorok ,HEADER: ÉRTÉK alakban
  - Accept: text/plain, text/html
  - Accept-Language: en
- A *header* sorok végét egy üres sor jelzi

# Metódusok I.

HTTP protokoll 8 féle metódust definiál. A metódusok (más szóval *verbek*) a megadott erőforráson végzendő műveletet határozzák meg.

- HEAD
  - Ugyanazt adja vissza, mint a GET, csak magát az üzenettestet hagyja ki a válaszból.
- GET
  - A megadott erőforrás letöltését kezdeményezi. Ez messze a leggyakrabban használt metódus.
- POST
  - Feldolgozandó adatot küld fel a szerverre. Például HTML űrlap tartalmát. Az adatot az üzenettest tartalmazza.
- PUT
  - Feltölti a megadott erőforrást.

# Metódusok II.

HTTP protokoll 8 féle metódust definiál. A metódusok (más szóval *verbek*) a megadott erőforráson végzendő műveletet határozzák meg.

- **DELETE**
  - Törli a megadott erőforrást.
- **TRACE**
  - Visszaküldi a kapott kérést. Ez akkor hasznos, ha a kliens oldal arra kívánCSI, hogy a köztes gépek változtatnak-e illetve mit változtatnak a kérésen.
- **OPTIONS**
  - Visszaadja a szerver által támogatott HTTP metódusok listáját.
- **CONNECT**
  - Átalakítja a kérést transzparens TCP/IP tunnellé. Ezt a metódust jellemzően SSL kommunikáció megvalósításához használják.

# GET metódus

- Ilyenkor a paraméterek az URL-ben adódnak át. Emiatt ez az oldal akár könyvjelzőbe is helyezhető.
  - Pl.
    - <https://www.google.hu/search?q=chuck+norris>
    - Látszik a keresendő kulcsszó a **q** paraméterben.
- Nem szabad jelszavak, vagy személyes információk átadására használni!
- Nem alkalmas nagyobb adatok küldésére!  
(url hossz korlátozás ~2083 karakter IE9-ben)

# POST metódus

- Ilyenkor a paraméterek egy (az url-ben nem látható) üzenetként továbbítódnak.
- Ezt lehet használni például felhasználói név és jelszó küldésénél.

Példa egy kérésre POST metódussal:

POST /login.jsp HTTP/1.1

Host: www.valami.hu

User-Agent: Safari/4.0

Content-Length: 27

Content-Type: application/x-www-form-urlencoded

userid=chuck&password=iamthegod

# Válasz (response)

- A HTTP válasz első sora a státuszszor, amely „VERZIÓ STÁTUSZKÓD INDOKLÁS” alakú.
- A státuszkód (status code) egy három számjegyű szám, az indoklás (reason phrase) egy angol nyelvű üzenet.
  - **HTTP/1.1 200 OK**

# Státuszkódok jelentése

- <http://tools.ietf.org/html/rfc2616>
  - 1xx: Informatív – Kérés megkapva.
  - 2xx: Siker – A kérés megérkezett; értelmezve, elfogadva.
  - 3xx: Átirányítás – A kérés megválaszolásához további műveletre van szükség.
  - 4xx: Kliens hiba – A kérés szintaktikailag hibás vagy nem teljesíthető.
  - 5xx: Szerver hiba – A szerver nem tudta teljesíteni az egyébként helyes kérést.

# Talán a legismertebb státuszkód

# 404

 The page cannot be found

The page you are looking for might have been removed, had its name changed, or is temporarily unavailable.

---

Please try the following:

- If you typed the page address in the Address bar, make sure that it is spelled correctly.
- Open the <http://httpd.apache.org> home page, and then look for links to the information you want.
- Click the  [Back](#) button to try another link.
- Click  [Search](#) to look for information on the Internet.

HTTP 404 - File not found  
Internet Explorer

# Státuszkódok macska őrülteknek



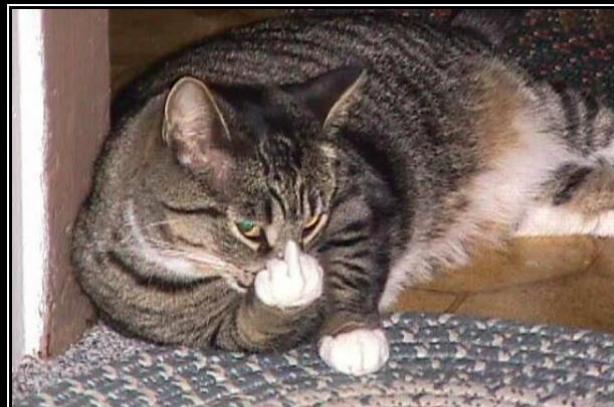
204

No Content



301

Moved Permanently



400

Bad Request



404

Not Found



414

Request-URI Too Long

# Header sorok

- A státuszsor után *header* sorok következhetnek „HEADERNÉV: ÉRTÉK” alakban
  - Date: Mon, 23 May 2009 22:38:34 GMT
  - Server: Apache/1.3.27 (Unix) (Red-Hat/Linux)
  - Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
  - Accept-Ranges: bytes
  - Content-Length: 438
  - Connection: close
  - Content-Type: text/html; charset=UTF-8

# Próbáld ki!

- Ha látni akarod hogyan néz ki egy HTTP kérés/válasz egy adott weboldalnál, látogasd meg a <http://web-sniffer.net/> oldalt!

## HTTP Request Header

Connect to 157.181.152.1 on port 80 ... ok

```
GET /HTTP/1.1[CRLF]
Host: www.elte.hu[CRLF]
Connection: close[CRLF]
User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:23.0) Gecko/20100101 Firefox/23.0[CRLF]
Accept-Charset: ISO-8859-1,UTF-8;q=0.7,*;q=0.7[CRLF]
Cache-Control: no-cache[CRLF]
Accept-Language: de,en;q=0.7,en-us;q=0.3[CRLF]
Referer: http://web-sniffer.net/[CRLF]
[CRLF]
```

## HTTP Response Header

Name	Value	Delim
Status:	HTTP/1.1 200 OK	
Date:	Sun, 08 Sep 2013 19:10:54 GMT	
Server:	Apache/2.2.17 (Fedora)	
Script-Control:	X-stream-mode	
Connection:	close	
Transfer-Encoding:	chunked	
Content-Type:	text/html;charset=utf-8	

## Content (70.02 KiB)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<HTML>
<HEAD>
<TITLE>::: www.elte.hu :: E&ouml;tv&ouml;s Lor&aacute;nd Tudom&aacute;nyegyetem :::</TITLE>
<META http-equiv="Content-Type" content="text/HTML; charset=utf-8">
<META name="Description" content="">
<META name="Keywords" content="">
<META name="Author" content="ELTE IIG WG.IWDG">
```

# Perzisztens kapcsolat

- A HTTP/0.9 és 1.0 verziókban a kapcsolat egy kérés-válasz után lezáródik
- A HTTP/1.1 verzióban vezettek egy mechanizmust a kapcsolat életben tartására, így a kapcsolat újrafelhasználható további kérésekhez. (**perzisztencia**)
  - Gyorsíthatja a kommunikációt, mert a kliensnek nem kell újratalálnia a TCP kapcsolatot minden egyes kérésnél.

# URL, URN, URI, stb..

- Hálózati erőforrásokra (weboldalakra, e-mail címekre, letölthető fájlokra, stb.) az **URL**-ek (**Uniform Resource Locator** - Egységes erőforrás helymeghatározó) segítségével hivatkozunk.
- Tartalmazza:
  - eléréshez szükséges kommunikációs protokoll nevét (pl. HTTP, HTTPS, FTP, MAILTO, NEWS, TELNET)
  - számítógép, vagy tartomány nevét (vagy IP címét), egy hálózati port számot, amelyen az adott szolgáltatás elérhető, és azt az elérési utat, amelyen az erőforrás megtalálható.
- URL = "protokoll:" "//" host [ ":" port ] [ absz\_útvonal [ "?" paraméterek]]

# URL példák

- <http://www.elte.hu/>
- <https://etr.elte.hu/etrweb/login.asp>
- <ftp://ftp.externet.hu/.lib/disk5/pub/win/ssh/putty.exe>
- <http://lib.nyme.hu:8080/corvina/opac/wpac.cgi>
- [https://www.google.hu/?gws\\_rd=ssl#q=webfejleszt%C3%A9s](https://www.google.hu/?gws_rd=ssl#q=webfejleszt%C3%A9s)

# URN

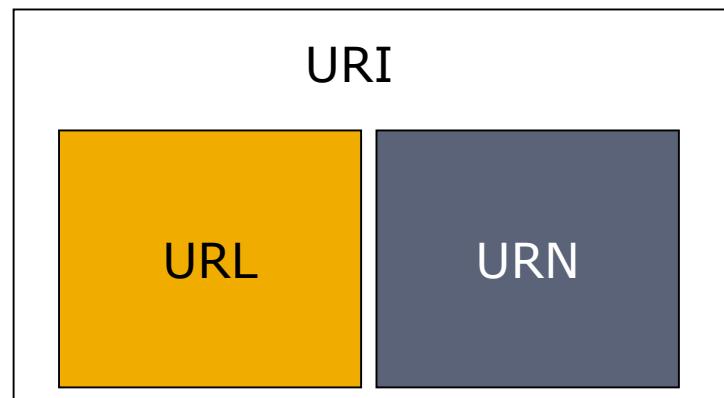
- Ne csak hálózati erőforrásoknak legyen azonosítója, hanem absztrakt erőforrásoknak is (könyv, folyóirat, stb.)
- **URN** (**U**niform **R**esource **N**ame - Egységes erőforrás név)
- Gondok az URL-el
  - Állandóan változhatnak
  - Az URL tartalma is változhat

# URN

- Példa: **urn:isbn:0-520-02356-0**
- A név nem változik
- A név világviszonylatban egyedi
  - Az URN kiadása ellenőrzött folyamat
- Az URL-re továbbra is szükség van

# URI

- URI: Uniform Resource Identifier (Egységes erőforrás azonosító)
- URL+URN=URI



# URL kódolás

- A HTTP URL-ekben ASCII karakterek lehetnek.
- A speciális karakterek átkódolásra kerülnek.

ASCII	Symbol	Replacement
< 32		Encode with %xx where xx is the hexadecimal representation of the character.
32	space	+ or %20
33	!	%21
34	"	%22
35	#	%23
36	\$	%24
37	%	%25
38	&	%26
39	'	%27
40	(	%28
41	)	%29
42	*	*
43	+	%2B
44	,	%2C

# A HTML nyelv

- HyperText Markup Language - Hiperszöveg jelölőnyelv
  - SGML (Standard Generalized Markup Language - szabványos, kiterjesztett jelölő nyelv)-en alapul (1986-os szabvány)
- Szöveges állomány .html vagy .htm kiterjesztéssel elmentve.
- Tageket és objektumhivatkozásokat tartalmaz:
  - Tag (címke): pl. <p>bekezdés</p>
  - Objektumhivatkozás: pl. "../kepek/logo.gif"

# Tagek (címkek)

- Tag-nek (címéknek) a < és > jelek között található jelölőelemet nevezzük.
- A címek többségének van záró párja. Ez ugyanaz, mint a nyitó címke, csak egy / jel vezeti be.
  - Pl. <p>Ez egy bekezdés</p>
- Nem minden címknak van záró párja a szabvány szerint.
  - Helyes:
    - <input type="button" value="Gomb">
  - Helytelen:
    - <input type="button" value="Gomb"></input>

# Tagek (címkék)

- A tag-eket egymásba is ágyazhatjuk, de ekkor fontos, hogy mindenkor azt a tag-et zárjuk le először, amelyet utoljára nyitottunk meg.
  - Hasonlóan ahhoz, ahogy mondjuk a matematikai kifejezéseket zárójelezzük ( $a + [ b * c ]$ ).
- **Helyes:** **< i >** szöveg **< /i >** **< /b >**
- **Helytelen:** **< b >** **< i >** szöveg **< /b >** **< /i >**

# Tag paraméterekkel

- A címkéknek lehetnek paramétereik is.
- A paraméterek (attribútumnak) van neve és értéke:
  - <tag paraméter1="érték1" paraméter2="érték2">
  - Pl. 
- Vannak logikai attribútumok is, ezeknél elég lehet a paraméter nevének szerepeltetése (ez a szabványtól függ!)
  - <p hidden>Ez egy rejtett bekezdés</p>
- Vannak olyan paraméterek, amelyeket minden címkénél lehet használni, ezeket globális paramétereknek nevezzük.
  - Ilyen például az id attribútum, amivel egyedi nevet adhatunk egy elemnek.

# HTML példa

```
<!DOCTYPE html>
<html lang="hu">
<head>
    <meta charset="utf-8">
    <title>Első weblapom</title>
</head>
<body>
<p></p>
<p>Ez az első próbálkozásom.
<b>Remélem sikerül!</b></p>
</body>
</html>
```



Ez az első próbálkozásom. **Remélem sikerül!**

# HTML történet

- 1995. november: HTML 2.0
- HTML 3.0
  - ▣ képek szöveggel történő körbefolyatása, táblázatok, matematikai képletek használata.
  - ▣ nem lett belőle hivatalos szabvány (túl komplex)
- 1997. január: HTML 3.2
- 1997. december: HTML 4.0
  - ▣ kliens oldali scriptnyelvek (pl. Javascript)
  - ▣ Stíluslapok használata
- 1999. december: HTML 4.01
  - ▣ Javításokat tartalmaz
  - ▣ Ez az utolsó (SGML-en alapuló) verzió
- 2014. október: HTML 5 <https://www.w3.org/TR/html5/>
  - ▣ A HTML5.1 (1. revízió) fejlesztés alatt áll. <https://www.w3.org/TR/html51/>

# XHTML

- Igény: webes adatbázisok közti kommunikációt lehetővé tévő, bővíthető nyelv => XML (eXtensible Markup Language - Bővíthető jelölőnyelv)
- **XHTML**
  - A HTML 4.01 XML alapokon történő megvalósítása
  - 2000. január: XHTML 1.0
  - 2001. május: XHTML 1.1
  - XHTML 2.0: leállt a fejlesztés, az erőforrásokat a HTML 5 fejlesztésére fordítják

# HTML és XHTML közti különbségek

- A HTML és XHTML nyelv között néhány szintaktikai különbség van, amit később részletezünk...

# HTML5

- Jelenleg még fejlesztés alatt áll, de az újdonságok egy részét már támogatják a korszerű böngészőprogramok.
- Előnye:
  - Egyszerűbben készíthetünk multimédiás lehetőségekben gazdag, interaktív alkalmazásokat .
  - A szabványban megjelenő új lehetőségeket a web-böngésző programok alap szolgáltatásként (natív módon) támogatják (pl. videó lejátszás).

# Korszerű böngészőprogramok

desktop browsers tablets mobiles other

latest search

## OVERVIEW

	Chrome	Firefox	Internet Explorer	Opera	Safari
Upcoming					9.0 → 400
Current	44 → 526	40 → 467	Edge → 402	31 → 525	8.0 → 396
Older	42 → 523	37 → 449	11 → 336	29 → 519	7.0 → 352
	40 → 511	35 → 449	10 → 297	26 → 497	6.0 → 326
	36 → 486	28 → 416	9 → 113	12.10 → 338	5.1 → 250
	28 → 433	18 → 371	8 → 33		

- <http://html5test.com/results/desktop.html>

# Microsoft Edge

- Az Edge böngésző váltja az Internet Explorer böngészőt (Windows 10)
- Egyelőre kevesen használják..





Forrás: <http://www.forgetfoo.com/images/blog/internet-explorer-098452345.jpg>

# Baj a HTML-el

- A HTML a dokumentum **tartalmának** leírására lenne jó.
- **Ha rosszul használjuk**, összemosódik a tartalom és a megjelenés.
  - Például:  
`<font face="Arial" color="red" size="3">Szöveg</font>`
  - Megoldás: a szabványok szigorúbb (Strict) változatai szerint dolgozzunk, a megjelenést pedig a stíluslapok segítségével írjuk le!

# HTML 4.01 struktúra

Dokumentum  
típus  
(DTD)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01  
Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

Fej

```
<head>  
  <title>Oldalcím</title>  
</head>
```

Törzs

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



# Struktúra HTML5 esetén

Dokumentum  
típus  
(DTD)

```
<!DOCTYPE html>
```

Fej

```
<html lang="hu">
  <head>
    <title>Oldalcím</title>
    <meta charset="utf-8">
  </head>
```

Törzs

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

# Dokumentumtípus

- A kód első sorába a dokumentumtípust írjuk. Ez határozza meg, hogy melyik szabvány, melyik altípusa szerint dolgozunk.
  - Ha ezt elhagyjuk, annak súlyos következményei lesznek, mivel a böngészőprogramok nem szabványos üzemmódba kerülnek, így az oldalak megjelenése, funkcionálitása sérülhet.
  - Altípusok
    - **Strict:** szűkített tag készlet, az elavult, stíluslapokkal kiváltható címkéket nem tartalmazza.
    - **Transitional:** bővebb tag készlet, az elavult tageket is tartalmazza.
    - **Frameset:** a keretdefiníciót tartalmazó oldalak esetén használható (de lehetőleg ne tegyük, elavult módszer!!!)

# Néhány dokumentumtípus

## **HTML 4.01 Transitional**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

## **HTML 4.01 Strict**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

## **HTML 4.01 Frameset**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"  
"http://www.w3.org/TR/html4/frameset.dtd">
```

## **XHTML 1.0 Transitional**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

## **XHTML 1.0 Strict**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

## **XHTML 1.0 Frameset**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

## **HTML 5**

```
<!DOCTYPE html>
```

# Szemantika fontossága

- Nagyon fontos, hogy az oldalakat úgy készítsük el, hogy minden elemet a neki szemantikailag megfelelő címkében helyezzünk el!
- Nem a kinézet a fontos, hanem az adott elem szerepe!
  - Pl. Egy címsor tényleg címsor elem legyen (pl. <h1>), ne pedig egy félkövér, nagybetűs szöveg.

# Validálás

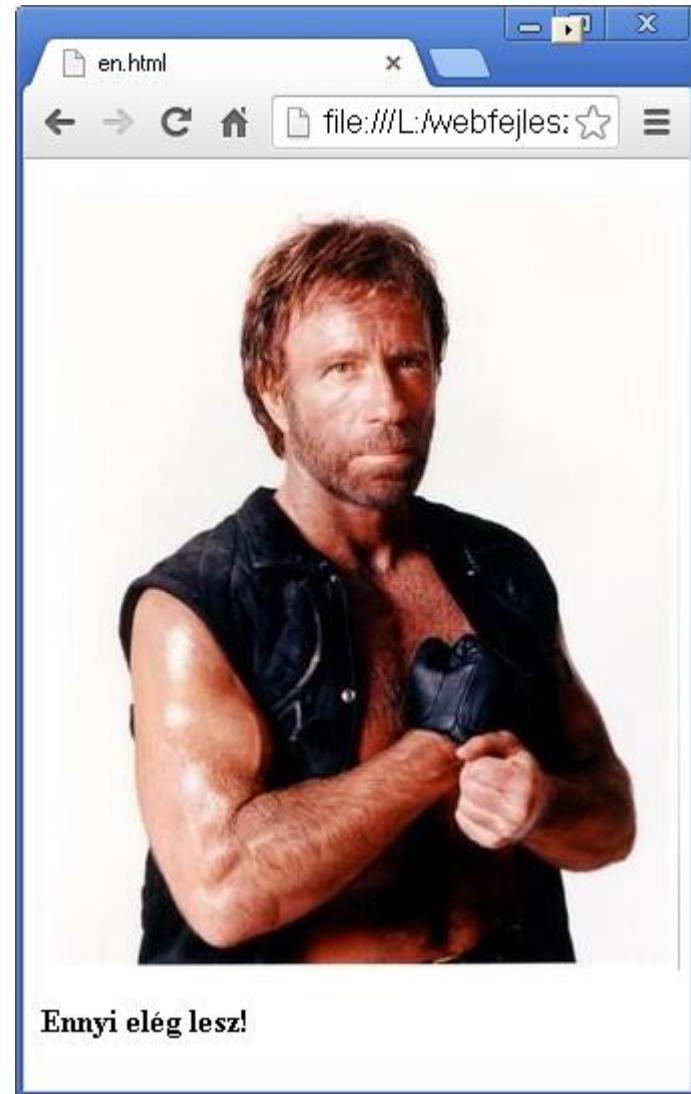
- A szabványosság ellenőrzésének folyamatát **validálásnak** nevezzük.
- **Hivatalos ellenőrző (validáló) eszköz:**
  - <http://validator.w3.org/>

The screenshot shows the W3C Markup Validation Service interface. At the top, there's a blue header bar with the W3C logo and the text "Markup Validation Service". Below it, a sub-header says "Check the markup (HTML, XHTML, ...) of Web documents". There are three tabs at the top: "Validate by URI" (which is selected), "Validate by File Upload", and "Validate by Direct Input". Under the "Validate by URI" tab, there's a section titled "Validate by URI" with the sub-instruction "Validate a document online:". Below this is a "Address:" label followed by a text input field. At the bottom of this section is a link "More Options". At the very bottom is a large, rounded "Check" button.

# Nem szabványos oldal

Hiányzó dokumentumtípus!!!

```
<html>
<head> Hiányzó <title> elem!!!
</head>
<body> Hiányzó kötelező alt paraméter!!!
<p></p>
<p><b>Ennyi elég lesz!</b></p>
</body>
</html> Rossz sorrendben lezárt címkék!!!
```



# Validálás eredménye

Errors found while checking this document as HTML 4.01  
Transitional!

**Result:** 5 Errors, 4 warning(s)

## Validation Output: 5 Errors

Line 1, Column 1: no document type declaration; implying "



"

<html>

The checked page did not contain a document type ("DOCTYPE") declaration. The Validator has tried to validate with a fallback DTD, but this is quite likely to be incorrect and will generate a large number of incorrect error messages. It is highly recommended that you insert the proper DOCTYPE declaration in your document -- instructions for doing this are given above -- and it is necessary to have this declaration before the page can be declared to be valid.

Line 3, Column 7: end tag for "HEAD" which is not finished

</head>

Most likely, you nested tags and closed them in the wrong order. For example <p><em>...</p> is not acceptable, as <em> must be closed before <p>. Acceptable nesting is: <p><em>...</em></p>.

Another possibility is that you used an element which requires a child element that you did not include. Hence the parent element is "not finished", not complete. For instance, in HTML the <head> element must contain a <title> child element, lists require appropriate list items (<ul> and <ol> require <li>; <dl> requires <dt> and <dd>), and so on.

Line 5, Column 27: required attribute "ALT" not specified

<p></p>

The attribute given above is required for an element that you've used, but you have omitted it. For instance, in most HTML and XHTML document types the "type" attribute is required on the "script" element and the "alt" attribute is required for the "img" element.

Typical values for type are type="text/css" for <style> and type="text/javascript" for <script>.

Line 6, Column 26: end tag for "B" omitted, but its declaration does not permit this

<p><b>Ennyi elég lesz!</p></b>

- You forgot to close a tag, or
- you used something inside this tag that was not allowed, and the validator is complaining that the tag should be closed before such content can be allowed.

The next message, "start tag was here" points to the particular instance of the tag in question); the positional indicator points to where the validator expected you to close the tag.

Line 6, Column 4: start tag was here

<p><b>Ennyi elég lesz!</p></b>

Line 6, Column 30: end tag for element "B" which is not open

<p><b>Ennyi elég lesz!</p></b>

The Validator found an end tag for the above element, but that element is not currently open. This is often caused by a leftover end tag from an element that was removed during editing, or by an implicitly closed element (if you have an error related to an element being used where it is not allowed, this is almost certainly the case). In the latter case this error will disappear as soon as you fix the original problem.

If this error occurred in a script section of your document, you should probably read this [FAQ entry](#).

# Miért jelent meg mégis jól az oldal?

- Mert Chuck Norris-szal a böngészőprogramok sem kekeckednek!
- Mert a böngészőprogramok megróbálják kijavítani a web-fejlesztők hibáit!

**Nem garantált, hogy minden hibát kijavítanak, illetve helyesen javítanak ki a böngésző-programok. A jövőbeni technológiák pedig a jelenlegi szabványok alapján nyújtanak visszafele kompatibilitást. Ezért a szabványokban foglaltakat be kell tartani!**

# Validálás a Firefox böngészőprogramban

- Telepítsük a **HTML validator** kiterjesztést a <http://users.skynet.be/mgueury/mozilla/> címről! (rövidített url: <http://bit.ly/qLVIX6>)
- Ezek után a böngészőprogram jobb alsó sarkában látjuk, hogy a megtekintett oldal valid-e.

Nem minden ugyanazt az eredményt adja, mint a W3C validátor! A W3C validátor a hivatalos!

HTML5-s oldalakra sajnos még nem működik az ikon megjelenítése! De az ikonra kattintva megjelenő ablakban látszanak a hibák (on-line ellenőrzés eredményeként)



# Stíluslapok használata

- Tartalom megadása: HTML állományban
- Kinézet (design) megadása: Stíluslap (CSS) állományban
  - Stílusdefiníciókat elhelyezhetünk a HTML állományon belül, de még előnyösebb lehet, ha külső állományokban helyezzük el az arculatra vonatkozó leírást.
  - A külső stíluslap egy egyszerű szöveg (plain text) állomány, .css kiterjesztéssel elmentve.

# Szöveg ismerkedéshez...



**LOLCAPTION.COM**

<http://www.lolcaption.com/win-funny-pictures-with-sayings/you-are-the-css-to-my-html/>

# Lépcsőzetes stíluslapok (CSS)

- CSS: Cascading Style Sheets
- CSS 1. szint: 1996 decemberében jelent meg
- Cél, hogy elkülönítsük a dokumentum megjelenését és tartalmát
  - Növeli a használhatóságot, rugalmasságot, megjelenés kezelhetőségét és csökkenti a dokumentum komplexitását.
- CSS 2.szint: 1998 május
- CSS 2.1.: butított változat, igazodva a böngészők (gyenge) tudásához
- CSS 3.szint:
  - Moduláris
  - Egyes részei már eljutottak a REC (recommendation) szintre.

# CSS Modulok

- <https://www.w3.org/Style/CSS/current-work>

## TABLE OF SPECIFICATIONS

Ordered from most to least stable:

Completed	Current	Upcoming	Notes	DOI
CSS Snapshot 2010	NOTE			<a href="#">DOI</a>
CSS Snapshot 2007	NOTE			<a href="#">DOI</a>
CSS Color Level 3	REC	REC	See <a href="#">Errata</a>	<a href="#">DOI</a>
CSS Namespaces	REC	REC		<a href="#">DOI</a>
Selectors Level 3	REC	REC		<a href="#">DOI</a>
CSS Level 2 Revision 1	REC	REC	See <a href="#">Errata</a>	<a href="#">DOI</a>
CSS Level 1	REC		Unmaintained, see <a href="#">Snapshot</a>	<a href="#">DOI</a>
CSS Print Profile	NOTE			<a href="#">DOI</a>
Media Queries	REC	REC		<a href="#">DOI</a>
CSS Style Attributes	REC	REC		<a href="#">DOI</a>
Stable	Current	Upcoming	Notes	DOI
CSS Backgrounds and Borders Level 3	CR	PR		<a href="#">DOI</a>
CSS Conditional Rules Level 3	CR	CR		<a href="#">DOI</a>
CSS Multi-column Layout	CR	CR		<a href="#">DOI</a>
CSS Values and Units Level 3	CR	PR		<a href="#">DOI</a>
CSS Cascading and Inheritance Level 3	CR	PR		<a href="#">DOI</a>
CSS Snapshot 2015	NOTE		Latest stable CSS	<a href="#">DOI</a>

# Azonos tartalom, különböző arculat

- <http://www.csszengarden.com/>

The screenshot displays the homepage of the CSS Zen Garden. At the top, there's a large, abstract photograph of a modern building at night with warm lights. Below it, the title "css zen garden" and subtitle "Beauty in CSS Design" are visible. A descriptive text block explains the purpose of the site: "A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page." It also provides links to download sample HTML and CSS files.

The main content area features a sidebar on the left with a list of design names and their creators:

- UNDER THE SEA! by Eric Stoltz
- MAKE 'EM PROUD by Michael McAgon and Scotty Reifsnyder
- ORCHID BEAUTY by Kevin Addison
- OCEANSCAPE by Justin Gray
- CSS CO., LTD. by Benjamin Klemm
- SAKURA by Tatsuya Uchida
- KYOTO FOREST by John Poltowski
- A WALK IN THE GARDEN by Simon Van Hauwermeiren

Below this is a section for "Archives" and "Resources". The "Archives" section includes links to "NEXT DESIGNS >" and "VIEW ALL DESIGNS". The "Resources" section includes links to "VIEW THIS DESIGN'S CSS" and "CSS RESOURCES".

The right side of the page has a sidebar titled "SELECT A DESIGN" which lists the same design names and their creators. It also includes a "ARCHIVES" section with links to "next designs >" and "View All Designs".

At the bottom, three thumbnail previews are shown:

- El Collar de Tomas MARIA STULTZ, Colombia 134
- OrderedZen STEVE SMITH, United States 133
- Bonsai MARTIN PLAZOTTA, Austria 132

# Stíluslap csatolása

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html lang="hu">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Stíluslapok</title>
    <link rel="stylesheet" type="text/css" href="kepek/pelda/stilus1.css" title="sajat">
    <style type="text/css">
      @import url("kepek/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>
```

Beágyazott  
(in-line)  
megadás

Lapon belüli  
definíció

Stíluslap  
importálás

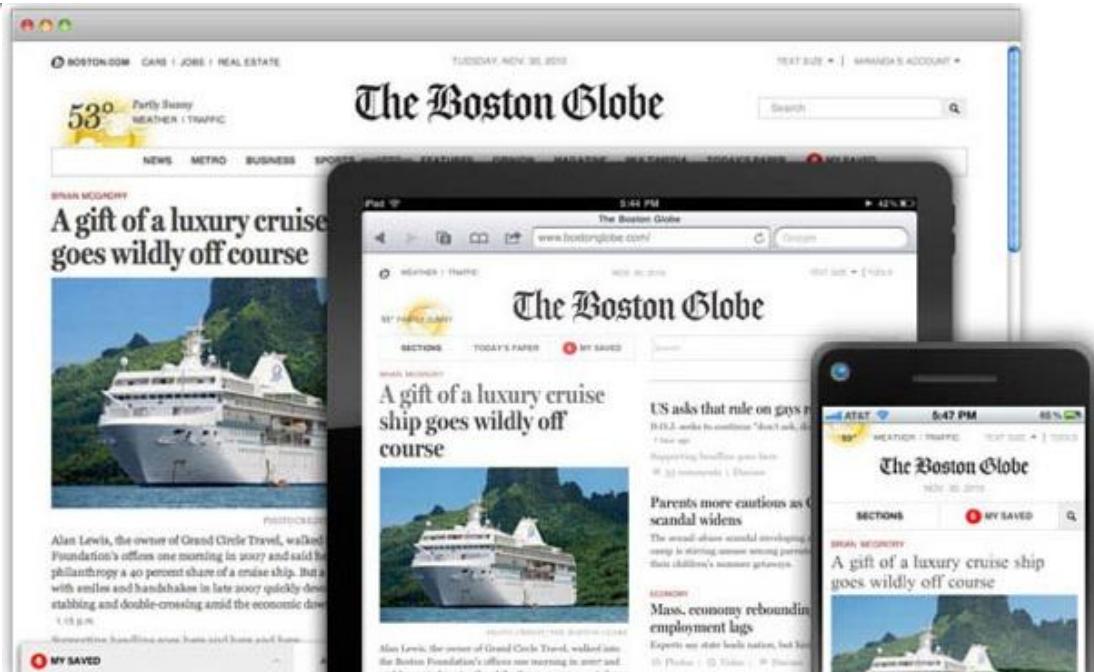
Hivatkozás  
külső  
stíluslapra

Ezt később részletesen  
áttekintjük.

# Médiafüggő stíluslapok

- Különböző média típusokhoz más-más stílusapot tudunk hozzárendelni.
- Optimalizálhatjuk a megjelenést asztali böngészőkre, mobil eszközökre, képernyőolvasókra, nyomtatásra, stb.
- Később bővebben is szót ejtünk erről...

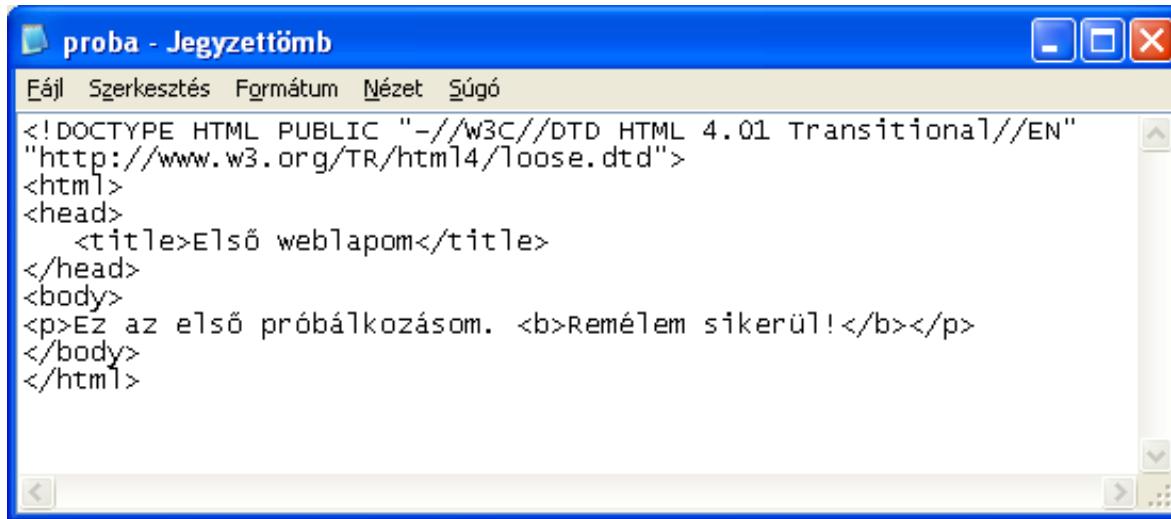
# Reszponzív webdesign



A **Responsive Web Design (RWD)** egy olyan tervezési módszert jelent, amelynek célja, hogy optimális megjelenést biztosítson (egyszerű olvashatóság, könnyű navigálhatóság) a különböző eszközökön (mobil eszközöktől a nagyobb felbontású monitorokig).

# Honlapszerkesztés eszközei

- Szövegszerkesztő program (egyszerű szövegként (txt) képes menteni)
  - Nem a legideálisabb választás...

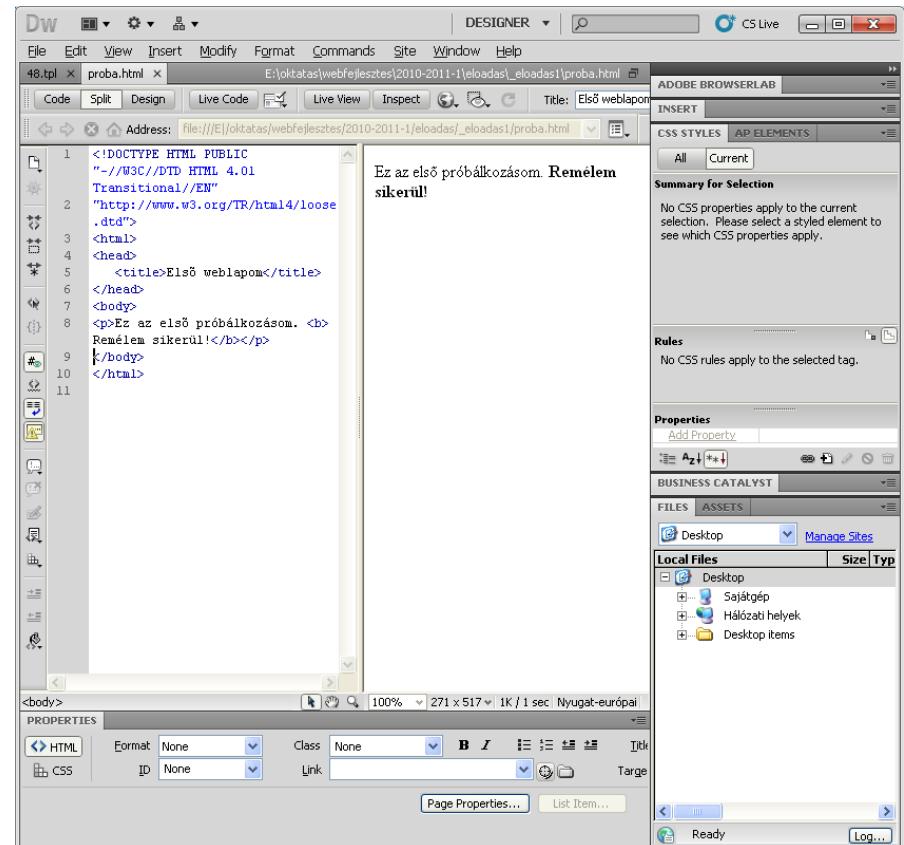


The screenshot shows a Windows Notepad window with a blue title bar. The title bar contains the text "proba - Jegyzettömb". Below the title bar is a menu bar with Hungarian labels: "Fájl" (File), "Szerkesztés" (Edit), "Formátum" (Format), "Nézet" (View), and "Súgó" (Help). The main window area contains the following HTML code:

```
<!DOCTYPE HTML PUBLIC "-//w3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <title>Első weblapom</title>
</head>
<body>
<p>Ez az első próbálkozásom. <b>Remélem sikerül!</b></p>
</body>
</html>
```

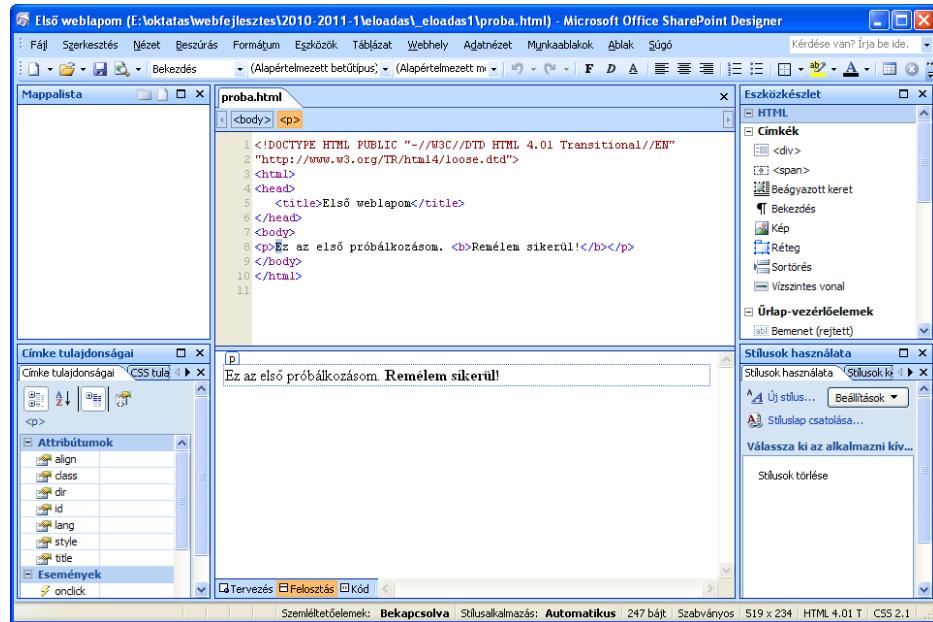
# Honlapszerkesztés eszközei

- Kódszintű szerkesztőprogram
  - Támogatja a szintaxis kiemelést, automatikus kódkiegészítést, stb.



# Honlapszerkesztés eszközei

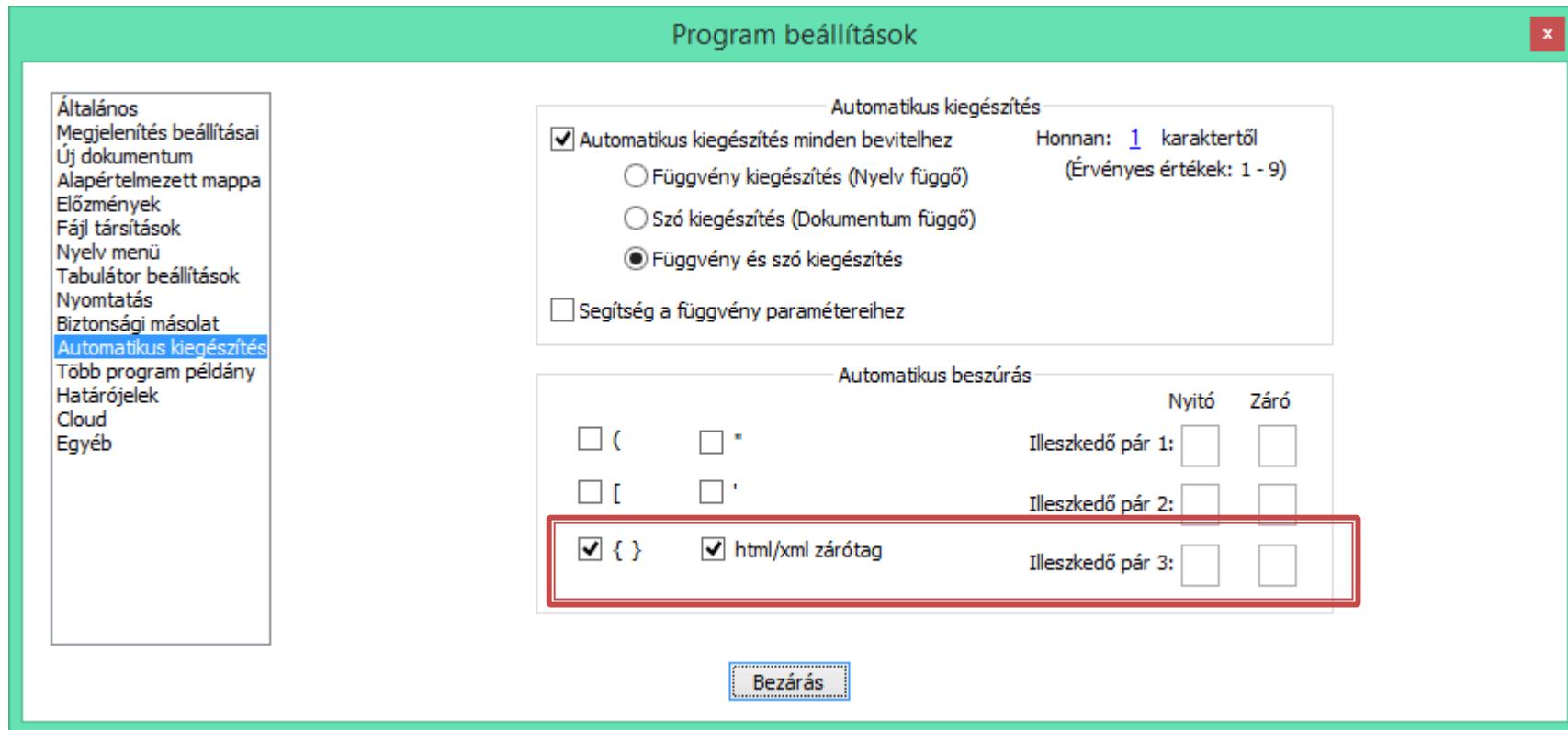
- WYSIWYG (alakhű) programok
  - A honlap szövegszerkesztő alkalmazásoknál megszokott felületen szerkeszthető.
  - A kód nem feltétlenül szabványos és hatékony, ezért nem ajánlott a használata.



# Ajánlott alkalmazások

- **Notepad ++ (Windows)**
  - A <http://notepad-plus-plus.org/> weboldalról letölthető alkalmazás támogatja a HTML és CSS kód kiemelését.
- **Kate editor (Linux)**
  - A <http://kate-editor.org/> oldalról lehet letölteni, támogatja az ablak felosztását, kódkiemelést, automatikus kiegészítést, stb.
- **TextWrangler (Mac OS X)**
  - A [TextWrangler alkalmazás](#) egy általános célú szerkesztőprogram, amely hasznos funkciókat biztosít a HTML kód szerkesztéséhez.

# Notepad++ hasznos beállítások



Ilyenkor lezárja gépelésnél a HTML taget. Vigyázzunk, mert ahol a szabvány szerint nincsen záró pár (pl. <br>), ott is kirak.

# A tananyagok bemutatása

# Tananyagok

- A tananyagok anélkül is elérhetőek, hogy a Moodle-be be kellene jelentkezni.
- A webcím:  
<http://webfejlesztes.elte.hu/tananyagok/>
- Név/jelszó kell az elérésükhez.
  - Név: wf1
  - Jelszó: Wcag2

# A weblapkészítés technikája (HTML5, CSS3) és ergonómiája



A A

## A weblapkészítés technikája (HTML5, CSS3) és ergonómiája



### Tartalomjegyzék

- A weblapkészítés technikája (HTML5, CSS3) és ergonómiája
  - Szerző
  - A modul célja és tartalma
  - A modul leckéi, a megértéshez szükséges előismeretek, tárgyi feltételek
  - A modul irodalomjegyzéke
  - Kulcsszavak



# Ikonok jelentése



Kattintás után visszakerülünk az előzőleg látogatott oldalra. Csak akkor használhatjuk, ha a böngészőnek van JavaScript támogatása.



A tananyag kezdőlapjára kerülünk.



A tananyag előző oldalára kerülünk.



A tananyag következő oldalára kerülünk.



Amennyiben a tananyag fogalom magyarázatokat is tartalmaz, akkor ezen ikonokkal kapcsolhatjuk be/ki a fogalmak megjelenítését. A fogalmak eltérő színnel és pontozott aláhúzással jelennek meg a szövegben. A fogalmak a billentyűzetet használó felhasználók számára elérhetőek, amennyiben a navigálás során a TAB billentyűt használják. A funkció kikapcsolt állapotában a fogalmak nem lesznek megjelölve.

Ez a funkció csak akkor használható, ha a böngészőnek van JavaScript támogatása, illetve a tananyag szerzői megjelölték a fogalmakat a tananyagban.



A lapon szereplő fogalmak listáját tudjuk önálló ablakban megjeleníteni, amennyiben az adott lapon vannak fogalmak.

Ez a funkció csak akkor használható, ha a böngészőnek van JavaScript támogatása, illetve a tananyag szerzői megjelölték a fogalmakat a tananyagban.



Az ikon lenyomásával előhívhatjuk a nyomtatás párbeszédablakot, amelyben a tananyag aktuális oldalát tudjuk kinyomtatni.



A tananyag oldaltérképére kerülünk, ahol fel vannak sorolva az egyes leckék, illetve azok aloldalai.



Ezen ikonokkal az oldalon belüli keresést támogató felületet lehet megjeleníteni, illetve eltüntetni.



A Súgó oldalra kerülünk.

# A weblapkészítés technikája (HTML5, CSS3) és ergonómiája - Leckék

- Bevezető
- HTML bevezető
- A HTML5 legfontosabb szöveg szintű elemei
- Kitekintés a stíluslapok használatába, tipikus szövegformázások
- Csoportosító elemek
- Beágyazott elemek
- Oldalszerkezet elemek
- Táblázatok használata
- Űrlapok használata
- Interaktív elemek
- Bevezetés a stíluslapok használatába
- CSS kiértékelési sorrend (rangsorolás)
- Ismerkedés a CSS dobozmodelljével
- Szín- és háttérbeállítás stíluslappal
- Vizuális formázásmodell
- Transzformációk és átmenetek
- Médiatípusok használata
- Úton a reszponzív arculat felé

# Honlapkészítés XHTML alapokon

## tananyag - Leckék

- XHTML bevezető
- Az XHTML 1.0 szabvány
- XHTML alapok
- Táblázatok
- Stíluslaphasználat alapjai
- Táblázatok formázása
- Képek beillesztése
- Animációk, multimédiás elemek
- Űrlapok
- Kliens oldali térképek (XHTML)
- Metaelemek (XHTML)
- Tipikus oldalfelépítések

# Honlapkészítés XHTML alapokon tananyag

## XHTML BEVEZETŐ

- ◆ Bevezető
- ◆ HTTP
  - ◆ Kérés
  - ◆ Válasz
  - ◆ Kommunikáció
  - ◆ Titkosított

## BEVEZETŐ

Ebben a tananyagban az **XHTML** [Extensible HyperText Markup Language] szabvány szerint fogunk dolgozni.

Az XHTML a **HTML**-hez hasonlóan egy szöveges állomány, ami különböző formázóutasításokat és a megjelenítendő elemekre (kép, videó, hang) mutató hivatkozásokat tartalmazhat. Ilyen állományt nagyon könnyen készíthetünk egy nagyon egyszerű szövegszerkesztő programmal is, csak a szükséges utasításokat kell használnunk.

Nézzünk egy egyszerű példát.

### Példa egy XHTML kódra

#### Kód részlet

Ez egy **vastag betűs** szöveg.

#### Eredmény

Ez egy **vastag betűs** szöveg.

# A HTML5

# **HTML5**



**HTML5** ~= **HTML + CSS + JS**

# Dokumentumtípus (DTD)

<!DOCTYPE html>

# Struktúra

```
<!DOCTYPE html>
<html lang="hu">
  <head>
    <title></title>
    <meta charset="utf-8">
  </head>
  <body>
    </body>
  </html>
```

# HTML 5 és XHTML 5

- A HTML 5 a standard HTML szintaxist követi, de lehet XML alapú változatot is használni.
  - <html>
  - <html xmlns="http://www.w3.org/1999/xhtml">

# Blokkszintű hivatkozások

```
<li>
  <a href="page.html">
    
    <h3>Title</h3>
    <p>Text</p>
  </a>
</li>
```



Home About Contact



Search

## Italics Kittehs

Posted 29 May 2008



Hungry kittens

[42 Comments](#) | [Permalink](#)

Rating:

© 2008 Cat Lovr. "Astrophy" cat logo created by and stolen from Hixie

**Cat Lovr** BETA

Home About Contact

**Italics Kittehs**

Posted 29 May 2008

Hungry kittens

[42 Comments](#) | [Permalink](#)

Rating:

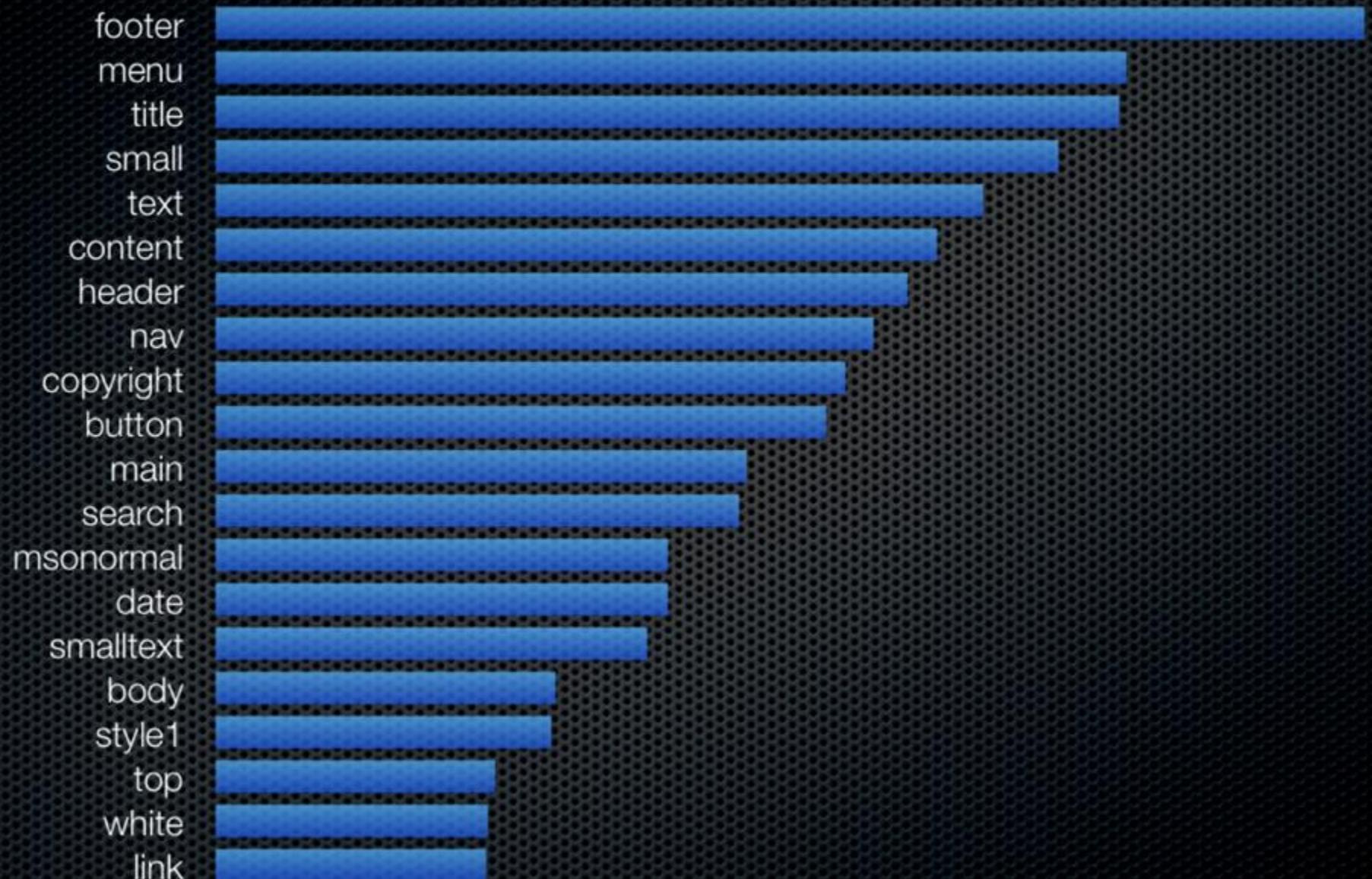
© 2008 Cat Lovr. "Astrophy" cat logo created by and stolen from Hixie

Search

Archives

[May 2008](#)  
[April 2008](#)  
[March 2008](#)  
[February 2008](#)  
[January 2008](#)  
[December 2007](#)  
[November 2007](#)  
[October 2007](#)  
[September 2007](#)  
[August 2007](#)

## Top 20 Class Names



Forrás: Lachlan Hunt:HTML 5 Overview, PubCon, Las Vegas 2008-11-13

**Cat Lovr** BETA

Home About Contact

## Italics Kittehs

Posted 29 May 2008

Hungry kittens

42 Comments | [Permalink](#)

Rating: 

© 2008 Cat Lovr. "Astrophy" cat logo created by and stolen from Hixie

**<header>**

**<nav>**

**Archives**

[May 2008](#)  
[April 2008](#)  
[March 2008](#)  
[February 2008](#)  
[January 2008](#)  
[December 2007](#)  
[November 2007](#)  
[October 2007](#)  
[September 2007](#)  
[August 2007](#)

**<aside>**

**<footer>**

# Whack A LOL

Posted 29 May 2008 <time>

<figure>



<figcaption>

I gotz me a hi score!

<meter>

[26 Comments](#) | [Permalink](#)

Rating:



# Customer Service <h1>

<section>

Rapidiously e-enable compelling customer service whereas progressive catalysts for change. Globally formulate cross-unit best practices for client-focused intellectual capital. Objectively target optimal total linkage with plug-and-play expertise.

Quickly reinvent long-term high-impact strategic theme areas via out-of-the-box paradigms. Dynamically leverage existing compelling innovation for focused architectures. Enthusiastically myocardiate magnetic internal or "organic" sources after enabled schemas.

# Development Strategies <h1>

Autoritatively exploit extensible e-tailers and just in time benefits. Competently communicate unique markets through backward-compatible channels. Synergistically strategize interactive action items for exceptional architectures.

Conveniently leverage other's extensible strategic theme areas for superior vortals. Synergistically iterate intuitive channels and frictionless data. Efficiently transition functionalized interfaces and corporate mindshare.

# Investments in Infrastructure <h1>

Distinctively transform clicks-and-mortar paradigms whereas ubiquitous infrastructures. Distinctively grow quality meta-services whereas sustainable architectures. Interactively initiate bricks-and-clicks ideas after installed base ROI.

Dynamically engage optimal quality vectors before bricks-and-clicks vortals. Energistically disseminate innovative methods of empowerment with fully tested e-tailers. Synergistically extend plug-and-play platforms rather than professional web services.

# Strukturális elemek (példa)

```
<body>
  <header>
    <hgroup>
      <h1>Page title</h1>
      <h2>Page subtitle</h2>
    </hgroup>
  </header>

  <nav>
    <ul>
      Navigation...
    </ul>
  </nav>

  <section>
    <article>
      <header>
        <h1>Title</h1>
      </header>
      <section>
        Content...
      </section>
    </article>
  </section>

  <aside>
    Top links...
  </aside>

  <figure>
    
    <figcaption>Chart 1.1</figcaption>
  </figure>

  <footer>
    Copyright ©
    <time datetime="2010-11-08">2010</time>.
  </footer>
</body>
```

# Ábra (példa)



*Australian Birds. From left to right, Kookaburra, Pelican and Rainbow Lorikeet*

*Originals by [Richard Clark](#)*

```
<figure>
  
  
  
  <figcaption>Australian Birds. From left to right, Kookaburra, Pelican and Rainbow Lorikeet.
  Originals by <a href="http://www.flickr.com/photos/rclark/">Richard Clark</a></figcaption>
</figure>
```

# HTML 5 tagek

## HTML Tags Ordered Alphabetically

 = New in HTML5.

Tag	Description
<u>&lt;!--...--&gt;</u>	Defines a comment
<u>&lt;!DOCTYPE&gt;</u>	Defines the document type
<u>&lt;a&gt;</u>	Defines a hyperlink
<u>&lt;abbr&gt;</u>	Defines an abbreviation
<u>&lt;acronym&gt;</u>	<b>Not supported in HTML5. Use &lt;abbr&gt; instead.</b> Defines an acronym
<u>&lt;address&gt;</u>	Defines contact information for the author/owner of a document
<u>&lt;applet&gt;</u>	<b>Not supported in HTML5. Use &lt;object&gt; instead.</b> Defines an embedded applet
<u>&lt;area&gt;</u>	Defines an area inside an image-map
<u>&lt;article&gt;</u> 	Defines an article
<u>&lt;aside&gt;</u> 	Defines content aside from the page content
<u>&lt;audio&gt;</u> 	Defines sound content
<u>&lt;b&gt;</u>	Defines bold text
<u>&lt;base&gt;</u>	Specifies the base URL/target for all relative URLs in a document
<u>&lt;basefont&gt;</u>	<b>Not supported in HTML5. Use CSS instead.</b> Specifies a default color, size, and font for all text in a document
<u>&lt;bdi&gt;</u> 	Isolates a part of text that might be formatted in a different direction from other text outside it

Teljes lista: <http://www.w3schools.com/tags/default.asp>

# Globális attribútumok

## HTML Global Attributes

ⓘ = Attribute added in HTML5.

Attribute	Description
<u>accesskey</u>	Specifies a shortcut key to activate/focus an element
<u>class</u>	Specifies one or more classnames for an element (refers to a class in a style sheet)
<u>contenteditable</u> ⓘ	Specifies whether the content of an element is editable or not
<u>contextmenu</u> ⓘ	Specifies a context menu for an element. The context menu appears when a user right-clicks on the element
<u>data-*</u> ⓘ	Used to store custom data private to the page or application
<u>dir</u>	Specifies the text direction for the content in an element
<u>draggable</u> ⓘ	Specifies whether an element is draggable or not
<u>dropzone</u> ⓘ	Specifies whether the dragged data is copied, moved, or linked, when dropped
<u>hidden</u> ⓘ	Specifies that an element is not yet, or is no longer, relevant
<u>id</u>	Specifies a unique id for an element
<u>lang</u>	Specifies the language of the element's content
<u>spellcheck</u> ⓘ	Specifies whether the element is to have its spelling and grammar checked or not
<u>style</u>	Specifies an inline CSS style for an element
<u>tabindex</u>	Specifies the tabbing order of an element
<u>title</u>	Specifies extra information about an element
<u>translate</u> ⓘ	Specifies whether the content of an element should be translated or not

# Ablak események

## Window Event Attributes

Events triggered for the window object (applies to the <body> tag):

Attribute	Value	Description
<u>onafterprint</u>	<span style="color: orange;">5</span> <code>script</code>	Script to be run after the document is printed
<u>onbeforeprint</u>	<span style="color: orange;">5</span> <code>script</code>	Script to be run before the document is printed
onbeforeunload	<span style="color: orange;">5</span> <code>script</code>	Script to be run before the document is unloaded
onerror	<span style="color: orange;">5</span> <code>script</code>	Script to be run when an error occur
onhaschange	<span style="color: orange;">5</span> <code>script</code>	Script to be run when the document has changed
<u>onload</u>	<code>script</code>	Fires after the page is finished loading
onmessage	<span style="color: orange;">5</span> <code>script</code>	Script to be run when the message is triggered
onoffline	<span style="color: orange;">5</span> <code>script</code>	Script to be run when the document goes offline
ononline	<span style="color: orange;">5</span> <code>script</code>	Script to be run when the document comes online
onpagehide	<span style="color: orange;">5</span> <code>script</code>	Script to be run when the window is hidden
onpageshow	<span style="color: orange;">5</span> <code>script</code>	Script to be run when the window becomes visible
onpopstate	<span style="color: orange;">5</span> <code>script</code>	Script to be run when the window's history changes
onredo	<span style="color: orange;">5</span> <code>script</code>	Script to be run when the document performs a redo
<u>onresize</u>	<span style="color: orange;">5</span> <code>script</code>	Fires when the browser window is resized
onstorage	<span style="color: orange;">5</span> <code>script</code>	Script to be run when a Web Storage area is updated
onundo	<span style="color: orange;">5</span> <code>script</code>	Script to be run when the document performs an undo
<u>onunload</u>	<code>script</code>	Fires once a page has unloaded (or the browser window has been closed)

# Űrlap és billentyű események

## Form Events

Events triggered by actions inside a HTML form (applies to almost all HTML elements, but is most used in form elements):

Attribute	Value	Description
<u>onblur</u>	script	Fires the moment that the element loses focus
<u>onchange</u>	script	Fires the moment when the value of the element is changed
oncontextmenu	script	Script to be run when a context menu is triggered
<u>onfocus</u>	script	Fires the moment when the element gets focus
onformchange	script	Script to be run when a form changes
onforminput	script	Script to be run when a form gets user input
oninput	script	Script to be run when an element gets user input
oninvalid	script	Script to be run when an element is invalid
<u>onreset</u>	script	Fires when the Reset button in a form is clicked <b>Not supported in HTML5</b>
<u>onselect</u>	script	Fires after some text has been selected in an element
<u>onsubmit</u>	script	Fires when a form is submitted

## Keyboard Events

Attribute	Value	Description
<u>onkeydown</u>	script	Fires when a user is pressing a key
<u>onkeypress</u>	script	Fires when a user presses a key
<u>onkeyup</u>	script	Fires when a user releases a key

# Egér események

## Mouse Events

Events triggered by a mouse, or similar user actions:

Attribute	Value	Description
<u>onclick</u>	script	Fires on a mouse click on the element
<u>ondblclick</u>	script	Fires on a mouse double-click on the element
ondrag	⌚ script	Script to be run when an element is dragged
ondragend	⌚ script	Script to be run at the end of a drag operation
ondragenter	⌚ script	Script to be run when an element has been dragged to a valid drop target
ondragleave	⌚ script	Script to be run when an element leaves a valid drop target
ondragover	⌚ script	Script to be run when an element is being dragged over a valid drop target
ondragstart	⌚ script	Script to be run at the start of a drag operation
ondrop	⌚ script	Script to be run when dragged element is being dropped
<u>onmousedown</u>	script	Fires when a mouse button is pressed down on an element
<u>onmousemove</u>	script	Fires when the mouse pointer moves over an element
<u>onmouseout</u>	script	Fires when the mouse pointer moves out of an element
<u>onmouseover</u>	script	Fires when the mouse pointer moves over an element
<u>onmouseup</u>	script	Fires when a mouse button is released over an element
onmousewheel	⌚ script	Script to be run when the mouse wheel is being rotated
onscroll	⌚ script	Script to be run when an element's scrollbar is being scrolled

# Média események

Attribute	Value	Description
onabort	script	Script to be run on abort
oncanplay	script	Script to be run when a file is ready to start playing (when it has buffered enough to begin)
oncanplaythrough	script	Script to be run when a file can be played all the way to the end without pausing for buffering
oncuechange	script	Script to be run when the cue changes in a <track> element
ondurationchange	script	Script to be run when the length of the media changes
onemptied	script	Script to be run when something bad happens and the file is suddenly unavailable (like unexpectedly disconnects)
onended	script	Script to be run when the media has reached the end (a useful event for messages like "thanks for listening")
onerror	script	Script to be run when an error occurs when the file is being loaded
onloadededata	script	Script to be run when media data is loaded
onloadedmetadata	script	Script to be run when meta data (like dimensions and duration) are loaded
onloadstart	script	Script to be run just as the file begins to load before anything is actually loaded
onpause	script	Script to be run when the media is paused either by the user or programmatically
onplay	script	Script to be run when the media is ready to start playing
onplaying	script	Script to be run when the media actually has started playing
onprogress	script	Script to be run when the browser is in the process of getting the media data
onratechange	script	Script to be run each time the playback rate changes (like when a user switches to a slow motion or fast forward mode)
onseeked	script	Script to be run when the seeking attribute is set to false indicating that seeking has ended
onseeking	script	Script to be run when the seeking attribute is set to true indicating that seeking is active
onstalled	script	Script to be run when the browser is unable to fetch the media data for whatever reason
onsuspend	script	Script to be run when fetching the media data is stopped before it is completely loaded for whatever reason
ontimeupdate	script	Script to be run when the playing position has changed (like when the user fast forwards to a different point in the media)
onvolumechange	script	Script to be run each time the volume is changed which (includes setting the volume to "mute")
onwaiting	script	Script to be run when the media has paused but is expected to resume (like when the media pauses to buffer more data)

# Új (beszédes) link kapcsolatok

```
<link rel="alternate" type="application/rss+xml" href="http://myblog.com/feed"/>
<link rel="icon" href="/favicon.ico"/>
<link rel="pingback" href="http://myblog.com/xmlrpc.php"/>
<link rel="prefetch" href="http://myblog.com/main.php"/>
...
<a rel="archives" href="http://myblog.com/archives">old posts</a>
<a rel="external" href="http://notmysite.com">tutorial</a>
<a rel="license" href="http://www.apache.org/licenses/LICENSE-2.0">license</a>
<a rel="nofollow" href="http://notmysite.com/sample">wannabe</a>
<a rel="tag" href="http://myblog.com/category/games">games posts</a>
...
```

Magyarázat:

<http://blog.whatwg.org/the-road-to-html-5-link-relations>

# Mikroadat (szematikus web)

```
<div itemscope itemtype="http://example.org/band">
  <p>My name is <span itemprop="name">Neil</span>.</p>
  <p>My band is called <span itemprop="band">Four Parts Water</span>.</p>
  <p>I am <span itemprop="nationality">British</span>.</p>
</div>
```



webmaster tools

Rich Snippets

Help with:

[Documentation](#)

Tips & Tricks

## Rich Snippets Testing Tool Beta

Rich Snippets allows you to enhance your Google search results by marking up web pages with Microformats, RDFa or Microdata.

Test your website

Enter a web page URL to see how it may appear in search results:

[Preview](#)

Examples: [Urbanspoon](#) [LinkedIn](#)

Google search preview

[Pizza My Heart - Santa Cruz | Urbanspoon](#)

★★★★★ 10 reviews · Price range: Under \$10 per entree

Excerpt from the page will show up here. Excerpt from the page will show up here.

Excerpt from the page will show up here. Excerpt from the page will show up here.

[www.urbanspoon.com/r8/765421/restaurant/Pizza-My-Heart-Santa-Cruz](#) · [Cached](#) · [Similar pages](#)

Note that there is no guarantee that a Rich Snippet will be shown for this page on actual search results. For more details, see the [E](#)

Extracted Rich Snippet data from the page

review-aggregate

item hood

In = Pizza My Heart

org

# Aria paraméter

```
<ul id="tree1"
    role="tree"
    tabindex="0"
    aria-labelledby="label_1">
  <li role="treeitem" tabindex="-1" aria-expanded="true">Fruits</li>
  <li role="group">
    <ul>
      <li role="treeitem" tabindex="-1">Oranges</li>
      <li role="treeitem" tabindex="-1">Pineapples</li>
      ...
    </ul>
  </li>
</ul>
```

Ezzel még hozzáférhetőbbé tehetők az internetes tartalmak a fogyatékossággal élő felhasználók számára.

# Új Úrlapmező típusok

**DATETIME**

*Allows input of a date and a time.*

**DATETIME-LOCAL**

*Allows input of a date and a time, in local time.*

**NUMBER**

*Allows input of a number.*

**RANGE**

*Input is verified to be within a range.*

**EMAIL**

*Confirms the input to be a valid email.*

**URL**

*Ensures input is a valid URL.*

**COLOR**

*Provides a mechanism for the user to input an RGB color.*

# Új Űrlapmező típusok

```
<style>
  [required] {
    border-color: #88a;
    -webkit-box-shadow: 0 0 3px rgba(0, 0, 255, .5);
  }
  :invalid {
    border-color: #e88;
    -webkit-box-shadow: 0 0 5px rgba(255, 0, 0, .8);
  }
</style>
```

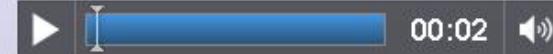
<input type="text" <b>required</b> />	<input type="text"/>
<input type="email" value="some@email.com" />	some@email.com
<input type="date" <b>min</b> ="2010-08-14" <b>max</b> ="2011-08-14" value="2010-08-14"/>	2010-08-17
<input type="range" <b>min</b> ="0" <b>max</b> = "50" value="10" />	<input type="range" value="10"/>
<input type="search" <b>results</b> ="10" <b>placeholder</b> ="Search..." />	<input type="search"/>
<input type="tel" <b>placeholder</b> ="(555) 555-5555" <b>pattern</b> ="^\\(\\d{3}\\)\\?[-\\s]\\d{3}[-\\s]\\d{4}.*?\$" />	(555) 555-5555
<input type="color" <b>placeholder</b> ="e.g. #bbbbbb" />	<input type="color"/>
<input type="number" <b>step</b> ="1" <b>min</b> ="-5" <b>max</b> ="10" value="0" />	0

# Audió, video

```
<audio id="audio" src="sound.mp3" controls></audio>  
document.getElementById("audio").muted = false;
```

```
<video id="video" src="movie.webm" autoplay controls></video>  
document.getElementById("video").play();
```

Add CSS reflection to video



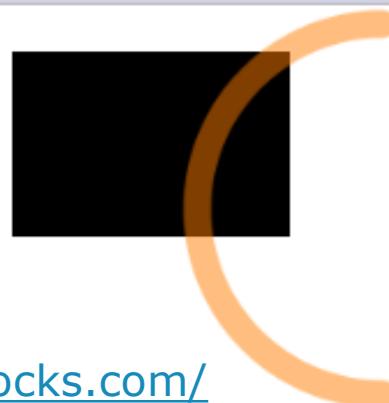
Play    Pause    Mute

# Vászon

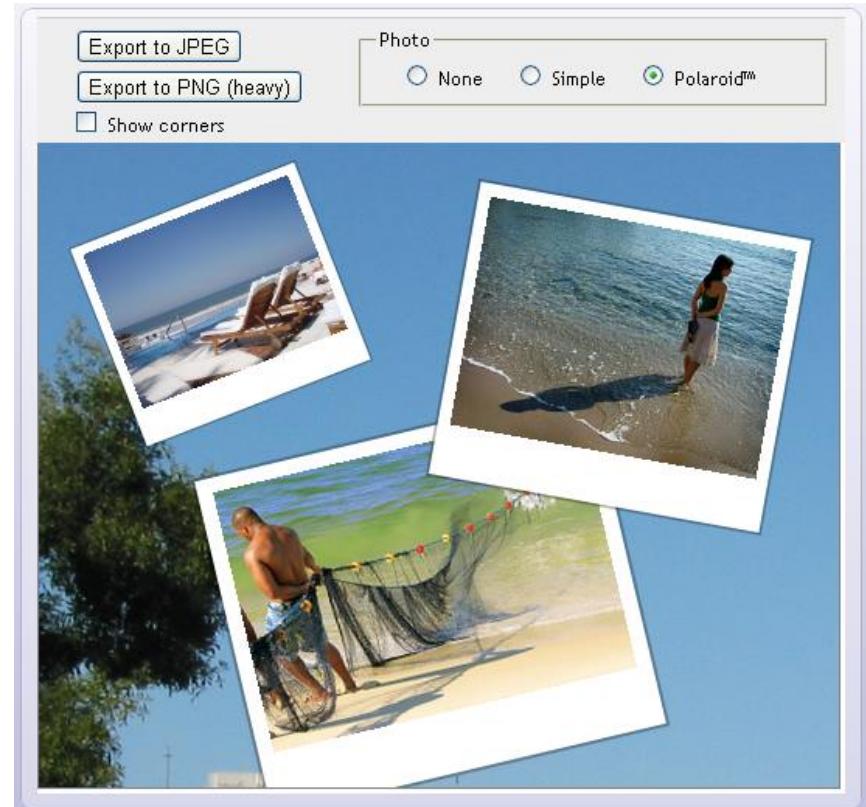
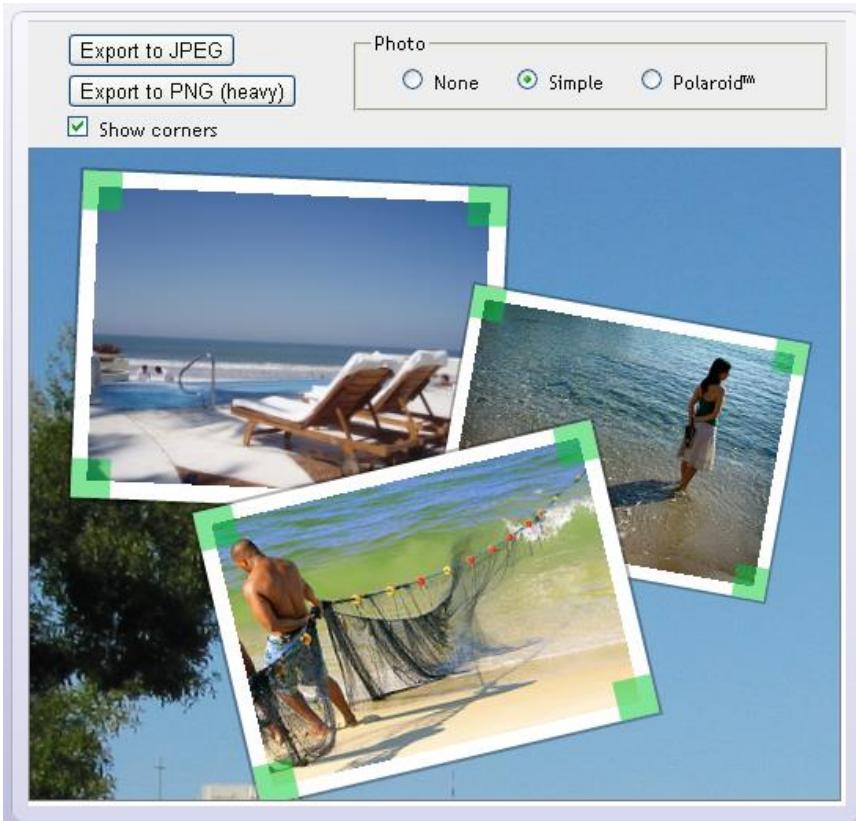
```
<canvas id="canvas" width="838" height="220"></canvas>

<script>
  var canvasContext = document.getElementById("canvas").getContext("2d");
  canvasContext.fillRect(250, 25, 150, 100);

  canvasContext.beginPath();
  canvasContext.arc(450, 110, 100, Math.PI * 1/2, Math.PI * 3/2);
  canvasContext.lineWidth = 15;
  canvasContext.lineCap = 'round';
  canvasContext.strokeStyle = 'rgba(255, 127, 0, 0.5)';
  canvasContext.stroke();
</script>
```



# Vászon példa



<http://slides.html5rocks.com/#canvas-2d-example>

# Vászon példa

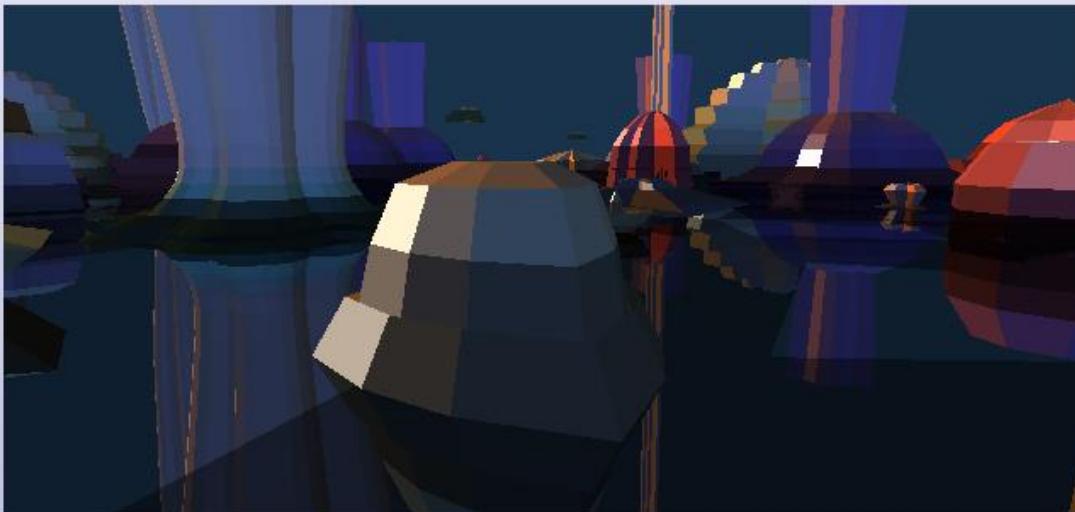


Exportálás

# Vászon 3D (WebGL)

```
<canvas id="canvas" width="838" height="220"></canvas>

<script>
  var gl = document.getElementById("canvas").getContext("experimental-webgl");
  gl.viewport(0, 0, canvas.width, canvas.height);
  ...
</script>
```



[Original demo](#) by Jetro Lauha WebGL port by Kenneth Waters



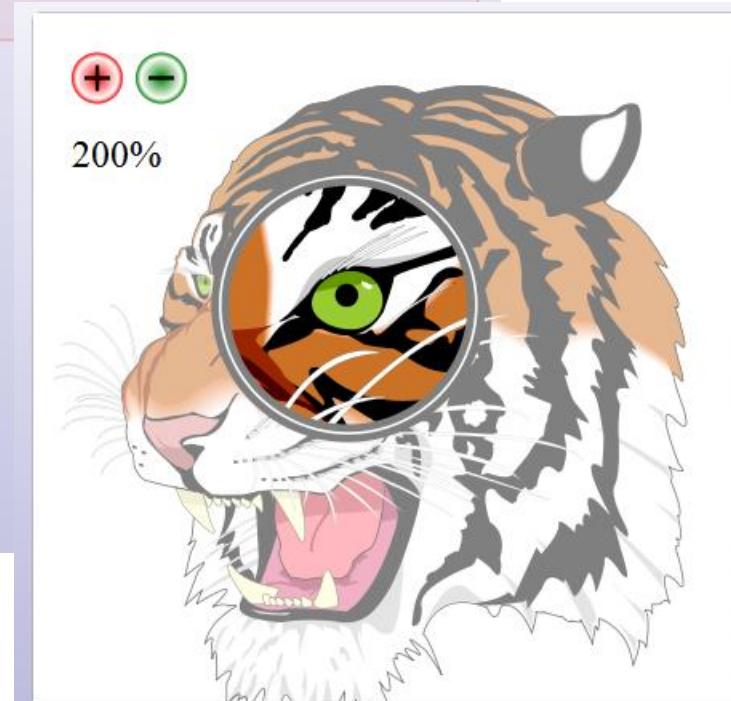
# Pesti György: A WebGL és WebSocket technológiák lehetőségeinek bemutatása egy 3D-s játék fejlesztése során



<http://racer.nomo.hu/>

# Vászon (beágyazott SVG)

```
<html>
  <svg>
    <circle id="myCircle" class="important" cx="50%" cy="50%" r="100"
      fill="url(#myGradient)"
      onmousedown="alert('hello');"/>
  </svg>
</html>
```



# Drag & Drop natív támogatása

JS

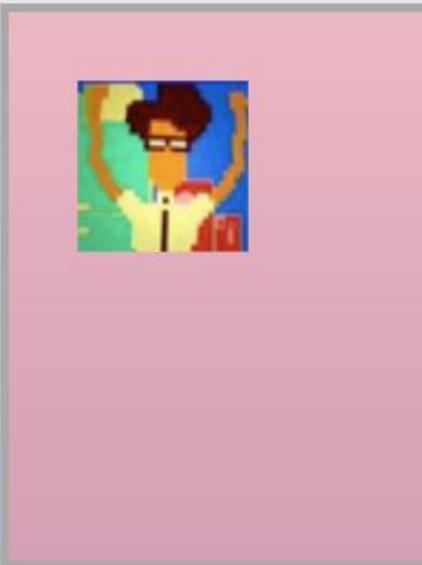
## Native Drag & Drop

```
document.addEventListener('dragstart', function(event) {  
  event.dataTransfer.setData('text', 'Customized text');  
  event.dataTransfer.effectAllowed = 'copy';  
}, false);
```



Select text and drag  
(original text will be  
dropped)

Select text and drag  
(dragged text data  
will be altered from  
original)



# Böngészők tesztelése HTML5 támogatás szerint

<http://html5accessibility.com/>

HTML5 Accessibility Support Score



HTML5 feature accessibility support in Major Browsers

HTML5 elements (links to test files)	Safari 8 - OSX	Chrome 43 - windows Opera 0	Firefox 38 - Windows	IE 11 - windows	Edge	Notes
<u>article</u>	✓	✓	✓	✗	?	IE: provides no semantic information via accessibility APIs ( <a href="#">related bug</a> ) Firefox: exposes element with a document role in MSAA/A2
<u>aside</u>	✓	✓	✓	✗	?	IE: provides no semantic information via accessibility APIs ( <a href="#">related bug</a> ) Firefox: exposes as ARIA role="complementary" via IA2 object attribute and IA2 note role.
<u>audio</u>	~~✓	✓	✓	✓	?	Safari: requires VoiceOver to be enabled to be usable with the keyboard. Firefox: (funky) keyboard access is provided via <a href="#">Firefox specific shortcuts</a> Chrome: Full support! IE 11: Full Support
<u>canvas</u>	~~✓	✓	✓	~~✓	?	IE, Chrome, Firefox: supports the accessible canvas element sub-DOM. Firefox, Chrome: support drawfocus ring. Safari: supports the accessible canvas element sub-DOM. Note: VoiceOver does not appear to announce text in canvas sub-DOM.
<u>datalist</u>	?	~~✓	✓	~~✓	?	Chrome: keyboard accessible. Does not expose datalist as list. ( <a href="#">related bug</a> )
<u>details</u>	✗	✓	?	?	?	Safari: refer to <a href="#">bug131111</a>

# **HTML és XHTML különbségei**

# XHTML

- XHTML (XML+HTML „házasság”)
  - Mindkét szabvány erősségeit magában foglalja, bővíthető, az XML miatti megkötések miatt jobban kell figyelnie a web-fejlesztőknek
  - A tartalmat bármilyen XML kompatibilis program képes megjeleníteni, feldolgozni

# XHTML

- XHTML: eXtensible HTML
- Előnyök
  - Az **XHTML** dokumentumok **XML** alapúak, könnyen áttekinthetők, szerkeszthetők, érvényesíthetők (validálhatók) standard **XML** eszközökkel.
  - Az **XHTML** oldalak a **HTML 4.0** szabványt támogató szerkesztőprogramokkal is szerkeszthetők.
  - Az **XHTML** dokumentumban használhatunk olyan alkalmazásokat (appletek, szkriptek), amelyek futtatásához szükséges a **HTML Document Object Model (DOM)** vagy az **XML DOM**.
  - Az **XHTML** "család" fejlődésével az **XHTML 1.0** kritériumainak megfelelő dokumentumok egyre inkább együtt tudnak működni egymással a különböző **XHTML** környezetekben.

# HTML, XHTML különbségek

- Az elemek egymásbaágyazásánál ügyelnünk kell a sorrendre

`<b><u>Vastag, aláhúzott szöveg</b></u>`



`<b><u>Vastag, aláhúzott szöveg</u></b>`



# HTML, XHTML különbségek

Jól formázott (well formed) dokumentumot kell létrehozni

- minden elemet a `<html>` elemen belül kell elhelyezni.
- minden elemnek lehetnek további beágyazott elemei.
- Ezek az elemek páronként kerülnek megadásra, és ügyelni kell arra, hogy a szülő elembe szabályosan kerüljön beágyazásra

# HTML, XHTML különbségek

A tag neveket kis betűkkel kell írnunk.

- az **XML** szabvány megkülönbözteti a kis- és nagybetűket a **<br>** és **<BR>** tag két különböző dolgot jelölhet

# HTML, XHTML különbségek

Minden XHTML elemet kötelező lezárni.

**<p>Első bekezdés**

**<p>Második bekezdés**



**<p>Első bekezdés</p>**

**<p>Második bekezdés</p>**



Azon elemek végére, amelyeknek a HTML szabványban nincs záró párjuk (**area, base, bgsound, br, col, frame, hr, img, input, isindex, keygen, link, meta, option, param**), / karaktert kell tennünk.

**<br>**

****

**<br />**

****

Ahhoz, hogy a dokumentum kompatibilis legyen a jelenlegi böngészőkkel, space karakter kell a / jel elő

# HTML, XHTML különbségek

A paramétereket kis betűvel kell írni.

**<table WIDTH="250">**



**<table width="250">**



# HTML, XHTML különbségek

A paraméterek értékeit "" jelek közé kell zárnunk

**<table WIDTH=250>**



**<table width="250">**



# HTML, XHTML különbségek

Az attribútumok minimalizálása, rövidítése tiltott

**<input checked>**



**<input checked="checked" />**



## *HTML és XHTML attribútumok*

HTML	XHTML	HTML	XHTML
compact	compact="compact"	ismap	ismap="ismap"
checked	checked="checked"	nohref	nohref="nohref"
declare	declare="declare"	noshade	noshade="noshade"
readonly	readonly="readonly"	nowrap	nowrap="nowrap"
disabled	disabled="disabled"	multiple	multiple="multiple"
selected	selected="selected"	noresize	noresize="noresize"

# HTML, XHTML különbségek

A *name* attribútumok helyett *id* attribútumot kell használni

```

```



```

```



A régebbi böngészők miatt érdemes egy ideig mindkét attribútumot szerepeltetni, ugyanazon értékkel.

```

```

# HTML, XHTML különbségek

Az xml:lang elem szerepe

**<div lang="hu" >Hajrás Magyarok</div>**



**<div lang="hu" xml:lang="hu">Hajrás  
Magyarok</div>**



# HTML, XHTML különbségek

## Kötelező XHTML elemek

- minden XHTML dokumentumnak rendelkezni kell DOCTYPE deklarációval. A html, head és body elemeknek szerepelniük kell, a title elemnek pedig a head elemen belül kell elhelyezkednie.

```
<!DOCTYPE ide jön a dokumentumtípus>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Oldalcím</title>
</head>
<body>
  Ide jön a tartalom.
</body>
</html>
```

# DOCTYPE

## 1. XHTML 1.0 Strict

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

## 2. XHTML 1.0 Transitional

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

## 3. XHTML 1.0 Frameset

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

# HTML -> XHTML konverzió?

## ■ HTML Tidy

- <http://tidy.sourceforge.net/#binaries>
- Parancsmódban használható program
- **tidy -help**

Paraméter	Magyarázat
-out <i>file</i>	a kimeneti állomány megadása
-modify	az eredeti állományt módosítja
-asxhtml	a HTML állományt jól formázott XHTML állománnyá konvertálja
-f <i>file</i>	a feldolgozás során jelentkező hibák kiírása a megadott állományba
-latin1	ISO-8859-1 karakterkódolás használata
-utf8	UTF8 karakterkódolás használata
-utf16	UTF16 karakterkódolás használata

# HTML -> XHTML konverzió?

```
tidy -f hiba.txt -asxhtml -utf8 -o pelda2.html pelda.html
```

## Eredeti kód

```
<html>

<head>
<meta http-equiv="Content-Language"
content="hu">
<meta http-equiv="Content-Type"
content="text/html; charset=windows-1250">
</head>

<body>

<p>Csak az teszteljük,<br>
hogy mi történik az átalakítás során....</p>

</body>

</html>
```

## Konvertált kód

```
<!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>
<meta name="generator" content=
"HTML Tidy for Windows (vers 1st December 2004),
see www.w3.org" />
<meta http-equiv="Content-Language" content="hu" />
<meta http-equiv="Content-Type" content=
"text/html; charset=utf-8" />
<title></title>
</head>
<body>
<p>Csak az tesztelj&#273;k,<br />
hogy mi t&#273;rtd&#273;nik az d&#273;talakd&#273;td&#273;s
sord&#273;n....</p>
</body>
</html>
```

VÉGE