

HTML

II. KÖNYV: HALADÓ SZINT

I. HTML 4.0 – Miért?

A HTML 3.2 igen rossz volt!

A HTML nyelvnek az eredeti szándék szerint sosem lett volna szaba formázó tageket tartalmaznia. A HTML-tageket az oldal *tartalmának* leírására (pl. `<p>Paragraph</p>` és `<h1>Heading</h1>`), és nem annak formázására találták ki.

Amikor a **** -hoz hasonló tageket és a szín-attribútumokat bekerültek a HTML 3.2 előírásai közé, a weboldal-fejlesztők munkája hirtelen rémálommá változott.

Az új weboldalak fejlesztése és fenntartása, ahol betű-megjelenítési és -szín-adatokat kellett minden egyes HTML-elemhez csatolni, hosszú és drága dologgá vált.

Miért nagyszerű a HTML 4.0?

A HTML 4.0 előírásai szerint az összes formázási információ elkülöníthető a HTML-dokumentumtól, egy külön stílus-lap létrehozásával (=style sheet).

Mivel a HTML 4.0 elválasztja a szövegfomázást a dokumentumtesttől, végre azt nyújtja, amire a fejlesztők régen vártak: tökéletes formázást, a dokumentum-szerkezet megbolygatása nélkül.

Hogyan használjuk?

- Ne használjunk formázó-attribútumokat a HTML-dokumentumokban!
- Használjuk a stílusokat! (A stílus-oldalak készítéséről a CSS-tutorialban olvashat.)
- Ne használjunk tiltott (=deprecated) tageket! (Melyeket [HTML 4.01 Reference](#)-ben foglaltunk össze.)
- Validáljuk a HTML-oldalainkat.

A HTML-fájlok validálása

A HTML-fájlok validálásához ún. „doctype declaration” (=dokumentumtípus-nyilatkozattétel) szükséges. Ez voltaképpen nem egy HTML-tag, hanem csupán eligazítás a böngésző számára, hogy a jelölőnyelv mely verziója szerint íródott az oldal.

A „doctype declaration” egy dokumentumtípus-definícióra (=Document Type Definition, DTD) utal. A DTD tartalmazza a jelölőnyelvre vonatkozó szabályokat, melyek segítségével a böngésző helyesen kezelheti a dokumentumot.

HTML 4.01 DTD-k:

Szigorú:

Ez a DTD minden HTML-elemhez alkalmas, kivéve a formázó- vagy tiltott elemeket (pl. ****). A frameset-ek sem alkalmazhatók;

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

Átmeneti/laza:

Ez a DTD minden HTML-elemet és attribútumot tartalmaz, beleértve a formázó- és tiltott elemeket (pl. ****). A frameset-ek azonban itt is tiltottak;

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

Frameset:

Ez a DTD ugyanaz, mint az átmeneti, de engedélyezi a framesetek használatát is:
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">

Validátor-programok

Az integrált fejlesztőkörnyezetek (=Integrated Development Environment, IDE) általában tartalmazznak hibakereső (=debug) funkciót, melyekkel a kódban előforduló jelölőnyelv-botlásokat megkereshetjük és kijavíthatjuk.

Vannak internetes validáló oldalak is (pl. <http://validator.w3.org/>), amelyek űrlapjába beírva az ellenőrizendő weboldal URL-jét, az oldalba ágyazott validáló/hibakereső program elkészíti és kiírja a hibák listáját.

A HTML DOCTYPE-elem

Tag	Leyrás
<!DOCTYPE>	Meghatározza a dokumentum-típust. A <html> nyitó-tag előtt áll.

Doctype-elemek:

Szigorú (csak szövegetest):

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

Átmeneti/laza (szövegetest + formázó és tiltott tagek):

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

Frameset (minden tag + frameset):

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

II. HTML-stílusok

A HTML 4.0 használatával minden formázásra vonatkozó információt eltávolíthatunk a HTML-dokumentumból, és egy külön „style sheet”-ben tárolhatjuk őket.

Néhány gyors példa:

Első példánkban a HTML-dokumentum **<head>** részében megadott stílusokkal kerülnek megjelenítésre az oldalon lévő **<h1>**, **<h2>** és **<p>** elemek:

```
<html>

<head>
<style type="text/css">
h1 {color:red;}
h2 {color:blue;}
p {color:green;}
</style>
</head>

<body>
<h1>All header 1 elements will be red</h1>
<h2>All header 2 elements will be blue</h2>
<p>All text in paragraphs will be green.</p>
</body>

</html>
```

Az alábbi példában szereplő link szövegét a böngésző nem húzza alá, mert ezt a szövegtestbe ágyazott **style** attribútummal megtiltjuk:

```
<html>
<body>

<a href="http://www.w3schools.com" style="text-decoration:none">
Visit W3Schools.com!</a>

</body>
</html>
```

Harmadik példánkban a szövegtestet egy különálló „style sheet”-tel formázzuk. A „style sheet”-et a weboldal **<head>** részében elhelyezett **<link>** taggel vonatkoztatjuk a **<body>** tartalmára:

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="styles.css" />
</head>

<body>
<h1>I am formatted with an external style sheet</h1>
<p>Me too!</p>
</body>
</html>
```

A style sheet tartalma:

```
h1 {color: green; border: '1pt solid black'}
p {color: red; background-color:#EFE7D6; border: '1pt solid black'}
div {color: #FFFFFF; background-color:#000000;}
```

```
span {color: #000000; background-color:#FFFFFF;}  
vagyis ugyanaz, mint amit a <head>-be írtunk volna (a link helyére).
```

Hogy használjuk a stílusokat

A style-sheet beolvasása után a böngésző annak megfelelően jeleníti meg az oldalt.

A stílus-elemek bevitelének (a style sheet megalkotásának) három módja van:

- Külső style sheet létrehozása (külön weboldalként).
- Belső style sheet írása (a <head> részbe).
- Szövegtestetn belüli („inline”) formázás (a formázni kívánt elembe írt **style** attribútummal).

Külső style sheet

A külső stílus-oldal ideális megoldás, ha sok weboldalnál ugyanazt a stílust akarjuk használni.

Alkalmazásakor egyetlen fájl átírásával egy egész portál arculatát megváltoztathatjuk.

A külső style sheet-hez a formázandó oldalak mindegyike a <link> taggel csatlakozik, melyet a <head> részben helyezünk el:

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

A <link> tag kapcsolatot létesít a dokumentum és egy külső forrás között. Leginkább külső style-sheet-ek tartalmának beidézésére használatos. Az <a> tag nem a dokumentumok tartalmát, hanem magukat a (teljes) dokumentumokat köti össze, hivatkozásként. A <link> tag tehát weboldalak tartalmát tölt át egymásba, illetve teszi külön szerkeszthetővé, míg az <a> tag a kész weboldalak között állapít meg kapcsolatokat.

Belső style sheet

A belső stílus-oldalt akkor használjuk, hogyha egyetlen dokumentumnak kell egyedi stílust adnunk. A belső stílusok a HTML-oldal <head> részében kerülnek meghatározásra, a <style> tag segítségével, pl.:

```
<head>  
<style type="text/css">  
body {background-color:yellow}  
p {color:blue}  
</style>  
</head>
```

Szövegtesten belüli formázás (inline styles)

A szöveg helyi formázása akkor célszerű, ha a fejrészben vagy style sheet-ben megadottól eltérő tulajdonságokkal akarunk felruházni egy vagy néhány kivételes elemet. Ehhez a **style** attribútumot használjuk, a megfelelő tagben. A **style** attribútum az összes CSS tulajdonságot tartalmazhatja.

A következő példa egy bekezdés betűszínét és bal margóját beállító **style** attribútumot tartalmaz:

```
<p style="color:blue;margin-left:20px">This is a paragraph.</p>
```

HTML stílus-tagek:

Tag	Leírás
<style>	A dokumentum stílus-adatait tartalmazza.
<link />	A dokumentum és egy külső forrás közt teremt kapcsolatot.

Tagek:

Külső style sheet:

Link a fejrészben:

Az URL style-sheet tartalma pl.:

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

```
h1 {color: green; border: '1pt solid black'}
```

```
p {color: red; background-color:#FFFFFF; border: '1pt solid black'}
```

```
div {color: #FFFFFF; background-color:#000000;}  
span {color: #000000; background-color:#FFFFFF;}
```

Belső style sheet:

Style tag a fejrészben pl.:

```
<style type="text/css">  
h1 {color:red;}  
h2 {color:blue;}  
p {color:green;}  
</style>
```

Szövegtesten belüli formázás (két példa):

Egyedileg formázott bekezdés:

```
<p style="color:blue;margin-left:100px"></p>
```

Egyedileg formázott link-szöveg (az automatikus aláhúzás kizárása):

```
<a href="URL" style="text-decoration:none">Szöveg</a>
```

III. HTML fej-elemek

Két gyors példa:

Első példánkban a dokumentumcímet leíró **<title>** taggel találkozunk (melyet a HTML-ALAPOK könyvben ismertünk meg), ami a weboldal **<head>** részében található:

```
<html>
<head>
<title>My first HTML page</title>
</head>

<body>
<p>The content of the body element is displayed in the browser.</p>
<p>The content of the title element is displayed in the browser's
title.</p>
</body>
</html>
```

Amint a két bekezdésben is szerepel, a **<body>** tartalmát a böngésző-ablakban, a **<title>** tag tartalmát pedig az ablak tetején lévő címmezőben jeleníti meg a böngésző.

A **<base>** tag használatával (szintén a **<head>** részben) egy a lapban felhasznált hivatkozásokhoz egy alapértelmezett URL-t vagy **target**-et adhatunk meg:

```
<html>

<head>
<base href="http://www.w3schools.com/images/" />
<base target="_blank" />
</head>

<body>

<p>Notice that we have only specified a relative address for the image.
Since we have specified a base URL in the head section, the browser
will look for the image at
"http://www.w3schools.com/images/stickman.gif"</p>

<a href="http://www.w3schools.com">W3Schools</a>
<p>Notice that the link opens in a new window, even if it has no
target="_blank" attribute. This is because the target attribute of the
base element is set to "_blank".</p>
</body>

</html>
```

Itt tehát a képhez elegendő volt relatív elérést írunk, mert az abszolút alapcímet a **<head>**-be írt **<base>** taggel megadtuk. Ugyanígy, az oldal összes külső hivatkozását „blank” targettel, azaz új ablakba nyitja meg a böngésző.

Megjegyezzük, hogy a **<blank>**-ban megadott előírások az oldal összes linkjére és URL-ére vonatkoznak; tehát ha a linkelt weboldal nem az ott leírt helyen található, akkor a **href** vagy **src** atribútumba abszolút hivatkozást kell írunk, illetve a **target** atribútumot is helyileg kell megadnunk; pl:

```
<html>
<head>
<base href="http://www.webforditas.hu" />
<base target="_blank" />
</head>

<body>
```

```
<a href="/szotar.php">SZÓTÁR</a>
<a href="http://www.mta.hu" target="_top">MTA</a>
</body>
</html>
```

Itt tehát a SZÓTÁR linkkel a <http://www.webforditas.hu/szotar.php> oldal nyílik meg, új ablakban; az MTA-val pedig a <http://www.mta.hu>, azonos ablakban („top” targettel).

Ugyanezt elérhetjük a következő módon is:

```
<html>
<head>
<base href="http://www.webforditas.hu" target="_blank" />
</head>
```

```
<body>
<a href="/szotar.php">SZÓTÁR</a>
<a href="http://www.mta.hu" target="_top">MTA</a>
</body>
</html>
```

A **target** attribútumot tehát a **href**-fel együtt, egyetlen **<base>** elemben is megadhatjuk. Ha azonban egymás után, egymást felülíró módon adjuk meg őket, az eredmények nehezen megjósolhatók – tehát célszerűen csak egyféle **href** és **target <blank>** elemet adjunk meg. (Egyébiránt a böngésző mindig az sorban legelől álló **href** és **target** beállítást veszi figyelembe.)

A HTML **<head>** elem

A **<head>** elem tartalmazza a dokumentum fej-részébe tartozó elemeket, azaz különféle parancsállományokat (=scripts), slítus-adatokat (pl. a style sheet elérési címét), meta-adatokat stb..

A fejrész a következő tageket tartalmazhatja: **<title>**, **<base>**, **<link>**, **<meta>**, **<script>** és **<style>**.

A HTML **<title>** elem

A **<title>** tag a dokumentum címét adja meg; ami minden HTML/XHTML dokumentum alapvető kelléke.

A **<title>** elembe írt cím:

- megjelenik a böngésző címsávjában,
- automatikus fejlévként szolgál az oldal könyvjelzőzésekor (Kedvencek közé vételekor),
- valamint a keresőmotorok is ezen a címen tüntetik fel a találati listán.

Az alábbi példa egy egyszerű HTML-dokumentumot mutat, melyben minden kötelező elem megtalálható:

```
<html>
<head>
<title>Title of the document</title>
</head>
```

```
<body>
The content of the document.....
</body>
```

```
</html>
```

A HTML **<base>** elem

A **<base>** tag egy alapértelmezett címet illetve megnyitási helyet (**target**) jelöl ki, az oldal összes linkjére nézve:

```
<head>
<base href="http://www.w3schools.com/images/" />
```

```
<base target="_blank" />
</head>
```

A két attribútumot (**href** és **target**) elegánsabb egyetlen **<base>** tagba foglalni.

A HTML **<link>** elem

A **<link>** tag egy külső forrással köti össze a dokumentumot. Leginkább style sheet-ek tartalmának betöltésére használatos:

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

A HTML **<style>** elem

A **<style>** tag a dokumentumra vonatkozó formázási adatokat tartalmazza. Meghatározza, hogy a böngésző hogy jelenítse meg a (**<body>**-ba tartozó) HTML-elemeket:

```
<head>
<style type="text/css">
body {background-color:yellow}
p {color:blue}
</style>
</head>
```

A HTML **<meta>** elem

A **<meta>** tag a HTML-dokumentumra vonatkozó metaadatokat (=metadata) tartalmazza. Részletesebb ismertetését lásd a következő fejezetben!

A HTML **<script>** elem

A **<script>** tag felhasználó-oldali (=client-side) parancsfájlokat (script-eket) tartalmazhat, pl. JavaScript-et. Részletesebben a következő fejezetben lesz róla szó.

HTML fej-elemek

Tag	Leírás
<u><head></u>	A dokumentumra vonatkozó általános információkat tartalmaz.
<u><title></u>	Dokumentumcím
<u><base /></u>	Alapértelmezett (az oldal összes linkjére vonatkozó) URL-t és targetet határoz meg
<u><link /></u>	A dokumentum és egy külső forrás között létre hoz kapcsolatot.
<u><meta /></u>	A HTML-dokumentumra vonatkozó metaadatokat tartalmazza
<u><script></u>	Felhasználó-oldali parancsfájl
<u><style></u>	Formázási adatokat tartalmaz.

Tagok:

Fej-elem: `<head></head>`

A következő elemeket tartalmazhatja: `<title> <base> <link> <meta> <script> <style>`

Dokumentumcím:

```
<title>Cím</title>
```

Alapértelmezett URL és target:

```
<base href="URL" target="_top/_blank/_parent/_self" />
```

Link (külső forrás adataival):

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

Belső style sheet:

```
<style type="text/css">
body {background-color:yellow}
p {color:blue}
</style>
```

Metaadat:

```
<meta></meta>
```

Felhasználó-oldali parancsfájl:

```
<script></script>
```


IV. A HTML <meta> elem

Három gyors példa:

Az első példában a dokumentum rövid leírására, „felcímkezésére” való <meta> elemet a szerző és a frissítési dátum megjelölésére alkalmazzuk:

```
<html>
<head>
<meta name="author" content="Hege Refsnes" />
<meta name="revised" content="2010/06/20" />
</head>

<body>
<p>The meta elements on this page identifies the author and the revise
date.</p>
</body>
</html>
```

Második példánkban a <meta> elemet az oldal tartalmának összefoglalására és a kulcsszavak kiemelésére alkalmazzuk:

```
<html>
<head>
<meta name="description" content="Free Web tutorials on HTML, CSS,
XML">
<meta name="keywords" content="HTML, CSS, XML">
</head>

<body>
<p>The meta elements on this page defines a description of the page,
and its keywords.</p>
</body>
</html>
```

A harmadik példában egy átirányító <meta> elemet látunk, mely öt másodperc után automatikusan átirányítja a felhasználót a **content** attribútumban leírt webhelyre:

```
<html>
<head>
<meta http-equiv="Refresh" content="5;url=http://www.w3schools.com" />
</head>

<body>
<h1>Sorry! We have moved!</h1>
<h2>The new URL is: <a
href="http://www.w3schools.com">http://www.w3schools.com</a></h2>
<p>You will be redirected to the new address in five seconds.</p>
<p>If you see this message for more than 5 seconds, please click on the
link above!</p>
</body>
</html>
```

A HTML <meta> elem

A „metadata” (=metaadat) nem más, mint adatra vonatkozó információ (adat).

A <meta> tag a HTML-dokumentumra vonatkozó metaadatokat tartalmazza. Mindig a <head> elemben található. A metaadatok nem jelennek meg az oldalon, de segítik a gépi feldolgozást.

A metaelemeket (<meta> tagbe írt HTML-elemeket) többnyire az oldal tartalmának rövid összefoglalására, kulcsszavak, szerzői és frissítési adatok és egyéb metaadatok megadására használjuk. Ezen adatokat böngészőprogramok (tartalom-megjelenítésre és oldal-újrátöltésre), keresőmotorok (a kulcsszavakat) és egyéb webes szolgáltatások hasznosíthatják.

Kulcsszavak a keresőmotorok számára

Egyes kereső,otorok a **name** és **content** attribútumokat használják az oldal-tartalmak felismeréséhez. A következő <meta> elem egy ilyen oldal-jellemzést tartalmaz:

```
<meta name="description" content="Free Web tutorials on HTML, CSS, XML" />
```

A következő <meta> elem pedig kulcsszavakat ad meg:

```
<meta name="keywords" content="HTML, CSS, XML" />
```

Megjegyezzük, hogy számos webmester használt <meta> tageket spameléshez, pl. úgy, hogy sokszor ismételte vagy hibásan adta meg a kulcsszavakat, hogy a keresőmotor előbbre sorolja az oldalt. Emiatt a legtöbb keresőprogram beszüntette a <meta> tagek alkalmazását az oldalak rangsorolásában.

Metaadat-tagek:

Szerzői adatok: <meta name="author" content="Hege Refsnes" />

Frissítési adat: <meta name="revised" content="2010/06/20" />

Rövid összefoglaló: <meta name="description" content="Free Web tutorials on HTML, CSS, XML">

Kulcsszavak: <meta name="keywords" content="HTML, CSS, XML">

Átirányítás: <meta http-equiv="Refresh" content="5;url=http://www.w3schools.com" />

[A **content** attriútumban álló szám az átirányítási művelet kezdetét határozza meg, az oldal megnyitásától számított másodpercekben, a célt pedig az **url** adja meg.]

V. HTML scriptek

Két gyors példa:

Első példánkban egy egyszerű scriptet (=parancsfájlt) mutatunk be:

```
<html>
<body>

<script type="text/javascript">
document.write("Hello World!")
</script>

</body>
</html>
```

Az oldal szövegtestében ekkor egy „Hello World!” felirat jelenik meg. Vagyis arra utasítottuk a browsert, hogy írja ki ezt a feliratot a dokumentum adott helyén.

Arra az esetre, hogyha a böngésző nem támogatja a parancsfájlok alkalmazását, vagy ez a funkció valamiért nem működik, a **<noscript>** taggel hibaüzenetet helyezhetünk el:

```
<html>
<body>

<script type="text/javascript">
document.write("Hello World!")
</script>

<noscript>Sorry, your browser does not support JavaScript!</noscript>

<p>A browser without support for JavaScript will show the text in the
noscript element.</p>

</body>
</html>
```

A HTML <script> elem

A **<script>** taget felhasználó-oldali (client-side) parancsfájlok (pl. JavaScript) beágyazására használjuk. A **<script>** elem vagy programszerű, szöveges utasításokat tartalmaz, vagy pedig egy illye tartalmú, külső fájlra (script fájlra) hivatkozik, az **src** attribútum segítségével.

A **type** attribútum használata kötelező a **<script>** tagben; ez a parancsfájl MIME-típusának kijelölésére szolgál. A MIME (=Multipurpose Internet Mail Extensions) egy internetes kommunikációs protokoll-gyűjtemény. A megfelelő előírást a **type** attribútummal adjuk meg, melynek értéke a MIME médiatípus és -altípus perjellel elválasztott jeléből áll, pl.:

```
type="text/javascript"
```

A JavaScriptet leginkább képek kezelésére, űrlapok hitelesítésére és dinamikus tartalomkezelésre használjuk.

A fenti scripttel

```
(<script type="text/javascript">
document.write("Hello World!")
</script>)
```

a „Hello World!” szöveget írattuk bele a HTML-kimenetbe.

A JavaScriptről a [JavaScript tutorial](#)-ból tudhatunk meg többet.

A HTML <noscript> elem

A **<noscript>** taget azért írjuk a dokumentumba, hogy a felhasználó-oldali parancsfájlokat nem támogató vagy hibásan működő böngészőprogramok használói számára helyettesítő tartalmat biztosítsunk.

A **<noscript>** elemben minden, a normál HTML-oldal **<body>** részébe írható elem elhelyezhető. Mindez azonban csak akkor kerül megjelenítésre, hogyha a fenti rendellenességek valamelyike fennáll.

Fenti példánkban szerepelt egy egyszerű **<noscript>** tag:

```
<noscript>Sorry, your browser does not support JavaScript!</noscript>
```

HTML script-tagek:

Tag	Leírás
<code><script></code>	Felhasználói parancsfájl (script)
<code><noscript></code>	Helyettesítő tartalom a <script> működési hibája esetére.

Script-tagek:

Szöveget kiírató JavaScript:

```
<script type="text/javascript">  
document.write("Szöveg")  
</script>
```

Script-helyettesítő (noscript) elem:

```
<noscript>Szöveg/Egyéb szövegtest-elemek</noscript>
```

VI. HTML-entitások

Az ún. foglalt karaktereket a HTML-nyelvben karakter-entitásokkal kell helyettesíteni.

HTML-entitások

A HTML-ben egynémely karakterek „foglaltak” (=reserved). Pl. nem használhatunk < (=„less than”) illetve > (=„greater than”) jeleket a szabad szövegben, mivel a közjük írt adatokat a böngésző tagnek tekinti.

A foglalt karakterek megjelenítéséhez tehát ún. entitásokat írunk a HTML-kódba.

A karakter-entitás kétféle képpen írható:

`&entity_name;`

vagy

`&#entity_number;`

Azaz az entitásokat „et” szimbólummal és az azután írt névvel vagy számkóddal azonosítjuk, és pontosvesszővel zárjuk le. Pl. a < jel kódjai:

`<`

vagy

`<`

Az entitás-nevek használata azért előnyösebb, mert könnyebb őket megjegyezni (mivel logikusak, pl. „less than” = „<”). Ugyanakkor nem minden böngésző támogatja az összes vagy ugyanazon entitás-neveket, míg ezzel szemben az entitás-számok általánosan elfogadottak.

„Tör(öl)hetetlen” szóközök

A törölhetetlen szóköz () gyakori HTML-entitás.

A böngészők ugyanis minig elhagyják a HTML-kódban szereplő többszörös szóközöket. Tehát pl. két betű közé írt 10 szóközből kilencet a megjelenítéskor eltávolít, s csak egyet mutat közülük. A (=„non-breaking space”) karakter-entitás használatával törölhetetlen szóközöket írhatunk a szövegbe.

Egy egyszerű példa:

```
<html>
```

```
<body>
```

```
<p>Character entities</p>
```

```
<p>&X;</p>
```

```
<p>
```

```
Substitute the "X" with an entity number like "#174" or an entity name like "pound" to see the result.
```

```
</p>
```

```
</body>
```

```
</html>
```

Mint látjuk, az első bekezdésben lévő egyetlen karakter-entitás értékét változtatva a böngésző más-más karaktereket jelenít meg; pl. **®** esetén trademark-jelet (®), **£** esetén pedig font-jelet (£).

Hasznos HTML karakter-entitások

Az alábbi táblázatban a gyakrabban előforduló karakter-entitásokat foglaltuk össze. Fontos ismét megjegyeznünk, hogy a karakter-entitás – nevek böngésző-kompatibilitása esetfüggő.

Karakter	Karakter-név (angol)	Entitás-név	Entitás-szám
	non-breaking space	 	
<	less than	<	<

>	greater than	>	>
&	Ampersand	&	&
¢	Cent	¢	¢
£	Pound	£	£
¥	Yen	¥	¥
€	Euro	€	€
§	Section	§	§
©	Copyright	©	©
®	registered trademark	®	®
™	Trademark	™	™

Az összes karakter-entitás kódja megtalálható [HTML Entities Reference](#) oldalunkon.

HTML karakter-entítások:

Entitás-név: &Név;

Entitás-szám: &#Szám;

Fontosabb karakter-entítások:

Törölhetetlen szóköz: () sp;

Less than: (<) < <

Egyenlő (=) =

Greater than: (>) > >

VII. HTML URL-ek (Uniform Resource Locators)

Az URL a webcím szó megfelelője. Állhat szavakból (pl. w3schools.com) vagy egy ún. Internet Protokolból (=Internet Protocol, IP; pl. 192.68.20.50). A legtöbben a webhely nevét írják be a böngészőbe, mert könnyebben megjegyezhetők, mint a számok.

URL – Uniform Resource Locator

A HTML-oldalakon szereplő linkeket alkotó <a> tagek webcímekekre hivatkoznak URL-ek segítségével.

Az URL a dokumentum azonosítására szolgál a weben. A webcímekek (pl.:

http://www.w3schools.com/html/default.asp) a következő nyelvtani elemekből állnak:

scheme://host.domain:port/path/filename

Jelentésük:

- **scheme:** az Internet-szolgáltatás **típusára** utal. Ez leggyakrabban **http** (=HyperText Transfer Protocol).
- **host:** megmutatja, melyik **webes tartományban** (=domain host) érhető el a kiszolgáló számítógép. A http alapértelmezett tartománya a www (=world wide web).
- **domain:** a **kiszolgáló számítógép neve** (=domain name), pl. w3schools.com.
- **port:** a kiszolgáló számítógép **csatorna-azonosítója** [=port number (at the host)]. http esetén az alapértelmezett csatorna-azonosító a 80.
- **path:** a kért oldal **elérési útvonala** a kiszolgáló számítógépen belül [=path (at the server)]. Nem kell megadni akkor, ha a megnyitni kívánt dokumentum a kiszolgáló gyökér-könyvtárában (=root), azaz legalapvetőbb mappájában van.
- **filename:** a megnyitni kívánt **dokumentum** vagy forrás **neve**.

A legtöbb webhely címe http://-rel kezdődik. Az ilyen oldalak nincsenek titkosítva, vagyis az oldal és a lekérdező számítógép közti adatcsere nyilvános, bárki hozzáférhet (hackerek is).

A biztosított/titkosított webcímekek https://-rel kezdődnek. Az „s” „biztosat” (=secure) jelent. Az itt folyó adatcsere titkosítva zajlik, ami értelmetlenné teszi annak megfigyelését (hacsak nem tudja valaki megfejteni a titkosított üzeneteket).

Gyakori URL típusok (schemes):

Az alábbi táblázatból megismerhetünk párat a legfontosabb internetes kommunikációs protokollak közül:

Rövidítés	Pontos név	Which pages will the scheme be used for...
http	HyperText Transfer Protocol	A közönséges weboldalak címe http://-rel kezdődik. Titkosítatlan kommunikációt eredményez, ezért nem célszerű személyes adatokat továbbítani ilyen oldalakra.
https	Secure HyperText Transfer Protocol	Az adatbiztos, titkosított weboldalak cím-kezdeté. Minden adatcsere titkosítva folyik, így a hackerek számára felhasználhatatlan.
ftp	File Transfer Protocol	Fájloknak webhelyekre való fel- és letöltésére szolgál. Internetes portálok üzemeltetésében hasznos.
file		Saját számítógépen lévő fájl.
gopher		Gopher dokumentum vagy menü. A Gopher www elődjeként egy fájlokból és menükből álló hálózat volt. Az alapjául szolgáló protokollt ma már sok böngésző nem támogatja, és csupán kb. 150 webhely használja (2010-ben).
news		Egy (többnyire Usenet protokoll alatt futó) hírcsoport címe. A hírcsoport sok különálló webhelyről beküldött adatok (hírek, vita-hozzászólások) halmaza (=repository).
WAIS	Wide Area Information Search	Egy WAIS adatbázis vagy dokumentum azonosítására szolgál. A WAIS (=Wide Area Information Servers) egy kliens- szerver kapcsolati protokoll, amely távoli számítógépek index-adatbázisainak (azaz címszavainak) keresését teszi lehetővé a felhasználóknak. Ezzel a

HTML-URL (általános alak):

típus://tartomány.kiszolgáló:csatorna/útvonal/dokumentum.fájlkiterjesztés

ahol:

- a **típus** a kommunikációs protokolt (pl.: **http/https/ftp**) jelöli;
- a **tartomány** az Internet azon részét, ahol a dokumentum van (pl.: **www**);
- a **kiszolgáló** a szerver-számítógép neve (pl.: **w3schools.com**);
- a **csatorna** a megnevezett kiszolgáló-számítógép elérni kívánt kommunikációs csatornáját jelöli, ahol az adat van (alapértelmezett értéke: http=**80**; https=**443**);
- az **útvonal** a dokumentum elérési útvonala a kiszolgálón (és csatornán) belül (pl.: **documents**);
- a **dokumentum** a megjelenítendő dokumentum neve (pl.: **mainpage**);
- a **fájlkiterjesztés** az előbbi fájl típusának rövidítése (pl.: **html**).

VIII. HTML URL-kódolás

Az URL-kódolás a szavakat biztonságosan továbbítható formátumúvá alakítja.

URL – Uniform Resource Locator

A webböngészők URL-ek segítségével érik el a letöltendő tartalom szolgáltatóját.

Az URL a letöltendő dokumentum címe, pl.: <http://www.w3schools.com>.

URL-kódolás

Az Interneten csak az ASCII betűkészlettel írt URL-ek továbbíthatók. Az ASCII (=American Standard Code for Information Exchange) egy 1960-as évekből eredő standard betűkészlet, melyeket számítógépek és perifériák (azaz más számítógépek, nyomtatók, mágnesszalag-táruk) közti kommunikációra találtak ki.

Az ASCII egy 7 bites, 128 karakterből álló készlet. Tartalmazza a 0-tól 9-ig terjedő arab számokat, az angol ABC nagy- és kisbetűit és néhány speciális jelet.

E betűkészletet használják a modern számítógépek, a HTML-nyelv és az Internet is. A következő táblázatban megadtuk a 128 ASCII karaktert, és a nekik megfelelő HTML entitás-kódokat is:

ASCII Character	HTML Entity Code	URL-kódolás	Description
(space)	 	%20	space
!	!	%21	exclamation mark
"	"	%22	quotation mark
#	#	%23	number sign
\$	$	%24	dollar sign
%	%	%25	percent sign
&	&	%26	ampersand
'	'	%27	apostrophe
((%28	left parenthesis
))	%29	right parenthesis
*	*	%2A	asterisk
+	+	%2B	plus sign
,	,	%2C	comma
-	-	%2D	hyphen
.	.	%2E	period
/	/	%2F	slash
0	0	%30	digit 0
1	1	%31	digit 1
2	2	%32	digit 2
3	3	%33	digit 3
4	4	%34	digit 4
5	5	%35	digit 5
6	6	%36	digit 6
7	7	%37	digit 7
8	8	%38	digit 8
9	9	%39	digit 9
:	:	%3A	colon
;	;	%3B	semicolon
<	<	%3C	less-than
=	=	%3D	equals-to

>	>	%3E	greater-than
?	?	%3F	question mark
@	@	%40	at sign
A	A	%41	uppercase A
B	B	%42	uppercase B
C	C	%43	uppercase C
D	D	%44	uppercase D
E	E	%45	uppercase E
F	F	%46	uppercase F
G	G	%47	uppercase G
H	H	%48	uppercase H
I	I	%49	uppercase I
J	J	%4A	uppercase J
K	K	%4B	uppercase K
L	L	%4C	uppercase L
M	M	%4D	uppercase M
N	N	%4E	uppercase N
O	O	%4F	uppercase O
P	P	%50	uppercase P
Q	Q	%51	uppercase Q
R	R	%52	uppercase R
S	S	%53	uppercase S
T	T	%54	uppercase T
U	U	%55	uppercase U
V	V	%56	uppercase V
W	W	%57	uppercase W
X	X	%58	uppercase X
Y	Y	%59	uppercase Y
Z	Z	%5A	uppercase Z
[[%5B	left square bracket
\	\	%5C	backslash
]]	%5D	right square bracket
^	^	%5E	caret
_	_	%5F	underscore
`	`	%60	grave accent
a	a	%61	lowercase a
b	b	%62	lowercase b
c	c	%63	lowercase c
d	d	%64	lowercase d
e	e	%65	lowercase e
f	f	%66	lowercase f
g	g	%67	lowercase g
h	h	%68	lowercase h
i	i	%69	lowercase i
j	j	%6A	lowercase j
k	k	%6B	lowercase k

l	l	%6C	lowercase l
m	m	%6D	lowercase m
n	n	%6E	lowercase n
o	o	%6F	lowercase o
p	p	%70	lowercase p
q	q	%71	lowercase q
r	r	%72	lowercase r
s	s	%73	lowercase s
t	t	%74	lowercase t
u	u	%75	lowercase u
v	v	%76	lowercase v
w	w	%77	lowercase w
x	x	%78	lowercase x
y	y	%79	lowercase y
z	z	%7A	lowercase z
{	{	%7B	left curly brace
	|	%7C	vertical bar
}	}	%7D	right curly brace
~	~	%7E	tilde

Mivel az URL gyakran az ASCII-n kívüli betűket is tartalmaz, át kell alakítani ASCII formátumúvá.

Az URL-kódoláskor a nem-ASCII karaktereket az ISO-8859-1 betűkészlettel írjuk le. Megkeressük a kódolni kívánt jelhez tartozó HTML-entitás-számot az utóbbi betűkészletben, majd azt hexadecimális alakban egy %-jel után írjuk. Pl. az Á betű ISO-8859-1 szerinti HTML-entitása: Á , tehát az ASCII-kompatibilis alak decimálisan %193 volna, ami hexadecimálisan %C1 (mivel $196/16=12=C$ és a maradék 1).

Az URL-ek szóközt sem tartalmazhatnak; az URL-kódolásban helyette leggyakrabban + jelet használunk.

Néhány egyszerű példa

Az alábbiakban néhány szó URL-kódolására látunk példát:

Hello Günter = Hello+G%FCnter

Benke Tamás = Benke+Tam%E1s

Magyarország = Magyarország%E1g

Mit sütsz, kis szücs? = Mit+s%FCtsz%2C+kis+sz%26%23369%3Bcs%3F

Talán sós húst sütsz, kis szücs!?! =

Tal%E1n+s%F3s+h%FAst+s%FCtsz%2C+kis+sz%26%23369%3Bcs%21%3F

URL-kód – példák:

A fenti és az alábbi táblázatban megadtuk néhány karakter URL-kódolását:

Character	URL-encoding
€	%80
£	%A3
©	%A9
®	%AE
À	%C0
Á	%C1

Â	%C2
Ã	%C3
Ä	%C4
Å	%C5

Az összes karakter URL-kódját megtaláljuk az [URL Encoding Reference](#) oldalon.

Nem-ASCII karakter URL-kódolása (az ISO-8859-1 HTML-entitás-száma alapján):

A karakter HTML-entitás-szám hexadecimális alakja (az ISO-8859-1 szerint): ÿ

A karakter URL-kódja: %FF

Szóköz URL-kódja: +

IX. HTML-webszerverek

Hogyan alakítsuk webszerverré (=internetes kiszolgálóvá) Windows-os számítógépünket?

1. Hogyha szeretnénk, hogy mások is megnézhessék weboldalainkat, publikálnunk kell őket.
2. Dokumentumaink publikálásához fel kell töltenünk őket egy webszerverre.
3. Saját számítógépünk is működhet webszerverként, hogyha installáljuk az IIS vagy PWS programokat. A Microsoft IIS és PWS programok ingyenes szerver-komponensek.

IIS – Internet Information Server

Az IIS a Microsoft által nyújtott Internet-alapú szolgáltatási csomag, Microsoft Windows operációs rendszerrel működő webszerverek számára. A Windows 2000, XP, Vista és NT rendszerekkel egyaránt hozzáférhető. Egyszerűen installálható és igen alkalmas webes alkalmazások fejlesztésére és tesztelésére.

Az IIS tartalmazza az ún. ASP-t (=Active Server Pages), ami egy szerver-oldali parancsfájl-norma (=server-side scripting standard); dinamikus és interaktív webes alkalmazások készíthetők vele. Az ASP-ről bővebben lásd az [ASP Tutorial](#)-t!

PWS – Personal Web Server

A PWS a régebbi (95, 98, NT) Windows-verziókhoz készült, az IIS-hez hasonló céllal. Könnyen installálható és weboldalak fejlesztésére és tesztelésére használható (ASP-vel is).

Használatát csak gyakorlásképpen javasoljuk, ugyanis idejétmúlt és biztonsági kockázatot jelent.

Windows webszerver-program – verziók

- Windows Vista Business, Enterprise és Ultimate: IIS 7
- Windows Vista Home Premium: IIS 7
- Windows Vista Home Edition: - (nem támogatja a PWS-t és IIS-t)
- Windows XP Professional: IIS 5.1
- Windows XP Home Edition: -
- Windows 2000 Professional: IIS 5.0
- Windows NT Professional: IIS 3 (az IIS 4-et is támogatja)
- Windows NT Workstation: PWS vagy IIS 3 telepíthető
- Windows ME: -
- Windows 98: PWS
- Windows 95: PWS

Az IIS installálása Windows Vistára

1. Nyissuk meg a „Vezérlőpultot” (=Control Panel) a Start menüből!
2. Kattintsunk a „Programok telepítése és törlése” (=Programs and Features) ikonra!
3. Kattintsunk a baloldali menüoszlopban a „Windows-összetevők hozzáadása vagy eltávolítása” (=Turn Windows features on or off) gombra!
4. Válasszuk ki az „Internet Information Services (IIS)” checkboxot, és nyomjuk meg az OK-t!

Az IIS installálása után a Windows Update futtatásával győződjünk meg róla, hogy az összes (biztonsági) programfrissítés is megtalálható-e a gépünkön!

Az IIS installálása Windows XP-re és 2000-re

csak annyiban különbözik az előzőtől, hogy a 3. lépés után „Windows-összetevők varázsló” IIS listaelemét ki kell jelölni a checkbox kipipálása előtt; majd alul a Részletek (=Details) gombra kattintva, a megnyíló listában kipipáljuk a „World Wide Web szolgáltatás” (=World Wide Web Service) checkboxot, és OK-t nyomunk. Ezután, a fenti 4. lépésnek megfelelően, kijelöljük az IIS-hez tartozó checkboxot és OK-t nyomunk! A számítógép naprakész voltát ugyanúgy ellenőrizzük!

Teszteljük webhelyünket!

Az IIS vagy PWS installálása után tegyük a következőket:

1. Keressük meg az **Inetpub** mappát a számítógépünk merevlemezén!
2. A mappát megnyitva egy **wwwroot** mappát találunk (=gyökérmappa).
3. Nyissunk egy új mappát (pl. **MyWeb**) a **wwwroot** mappában!
4. Írjunk egy ASP (vagy HTML) oldalt, és mentjük el ebbe a mappába **test1.asp** vagy **test1.html** néven!
5. Bizonyosodjunk meg arról, hogy webszerverünk működik! (Lásd alább!)
6. Nyissuk meg a böngésző-programot, és írjuk be a **http://localhost/MyWeb/test1.asp** vagy **http://localhost/MyWeb/test1.html** címet, és nyissuk meg weboldalunkat!

A webkiszolgálónk működésével kapcsolatban érdemes megkeresnünk az IIS (vagy PWS) pontot a Start menüben! Az IIS program segítségével elindíthatjuk ill. lekapcsolhatjuk szerverünket, ki- illetve bekapcsolhatjuk az AAS-szolgáltatásokat stb..

Tovább lépés: egy professzionális webhely/webszerver létrehozása

- Hogyha nem akarjuk használni a PWS-t vagy IIS-t, egy publikus szerverre kell feltöltenünk az adatokat.
- A legtöbb internetes szolgáltatónál (=Internet Service Provider, ISP) lehet webhelyet üzemeltetni.
- Ha a munkahelyünknek, iskolánknak vagy ismerősünknek van internetes szervere, megfelelő egyeztetés utánm oda is feltölthetjük webhelyünket.
- Ha igazán komoly webhelyben gondolkozunk, saját webszervert is létrehozhatunk.

A megfelelő megoldás illetve internet-szolgáltató vagy -tárhely kiválasztásához tanulmányozzuk át a [Web Hosting Tutorial](#)-t!

A számítógépünk webes gyökérmappájában lévő dokumentumok URL-je:

http://localhost/elérési_út/dokumentumnév.fájlkiterjesztés

X. HTML zárszó

A HTML-ről szóló I. és II. könyvből megismerkedtünk a webhely-készítés alapfogásaival.

A HTML egy univerzális jelölő-nyelv, számítógép-hálózatok számára. Szöveget formázhatunk, grafikai elemeket kezelhetünk, linkeket, beviteli elemeket, keretcsoportokat, táblázatokat stb. készíthetünk vele; s mindezt egy szöveges (=text) fájl formájában elmentve, minden böngésző számára olvashatóvá és megjeleníthetővé tehetjük.

A HTML legfontosabb elemei a tagek, melyek a soron következő tartalom jellegét jelölik.

HTML-ÖSSZEFOGLALÁS

II. KÖNYV: HALADÓ SZINT

I. Doctype-elemek

Szigorú (csak szövegtest):

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

Átmeneti/laza (szövegtest + formázó és tiltott tagek):

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

Frameset (minden tag + frameset):

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

II. HTML-stílusok

Külső style sheet:

Link a fejrészben:

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

Az URL style-sheet tartalma pl.:

```
h1 {color: green; border: '1pt solid black'}
p {color: red; background-color:#FFFFFF; border: '1pt solid black'}
div {color: #FFFFFF; background-color:#000000;}
span {color: #000000; background-color:#FFFFFF;}
```

Belső style sheet:

Style tag a fejrészben pl.:

```
<style type="text/css">
h1 {color:red;}
h2 {color:blue;}
p {color:green;}
</style>
```

Szövegtesten belüli formázás (két példa):

Egyedileg formázott bekezdés:

```
<p style="color:blue;margin-left:100px"></p>
```

Egyedileg formázott link-szöveg (az automatikus aláhúzás kizárása):

```
<a href="URL" style="text-decoration:none">Szöveg</a>
```

III. HTML fej-elemek

Fej-elem: <head></head>

A következő elemeket tartalmazhatja: <title> <base> <link> <meta> <script> <style>

Dokumentumcím:

```
<title>Cím</title>
```

Alapértelmezett URL és target:

```
<base href="URL" target="_top/_blank/_parent/_self" />
```

Link (külső forrás adataival):

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

Belső style sheet:

```
<style type="text/css">
body {background-color:yellow}
p {color:blue}
</style>
```

Metaadat:

```
<meta></meta>
```

Felhasználó-oldali parancsfájl:

```
<script></script>
```

IV. A HTML <meta> elem (metaadatok)

Szerzői adatok: <meta name="author" content="Hege Refsnes" />

Frissítési adat: <meta name="revised" content="2010/06/20" />

Rövid összefoglaló: <meta name="description" content="Free Web tutorials on HTML, CSS, XML">

Kulcsszavak: <meta name="keywords" content="HTML, CSS, XML">
Átirányítás: <meta http-equiv="Refresh" content="5;url=http://www.w3schools.com" />
[A **content** attribútumban álló szám az átirányítási művelet kezdetét határozza meg, az oldal megnyitásától számított másodpercekben, a célt pedig az **url** adja meg.]

V. HTML scriptek

Szöveget kiírató JavaScript:

```
<script type="text/javascript">  
document.write("Szöveg")  
</script>
```

Script-helyettesítő (noscript) elem:

```
<noscript>Szöveg/Egyéb szövegtest-elemek</noscript>
```

VI. HTML (karakter)-entitások

Entitás-név: &Név;
Entitás-szám: &#Szám;

Fontosabb karakter-entitások:

Törölhetetlen szóköz:	()	 	
Less than:	(<)	<	<
Egyenlő	(=)		=
Greater than:	(>)	>	>

VII. HTML URL-ek (Uniform Resource Locators)

HTML-URL (általános alak):

típus://tartomány.kiszolgáló:csatorna/útvonal/dokumentum.fájlkiterjesztés

ahol:

- a **típus** a kommunikációs protokolt (pl.: **http/https/ftp**) jelöli;
- a **tartomány** az Internet azon részét, ahol a dokumentum van (pl.: **www**);
- a **kiszolgáló** a szerver-számítógép neve (pl.: **w3schools.com**);
- a **csatorna** a megnevezett kiszolgáló-számítógép elérni kívánt kommunikációs csatornáját jelöli, ahol az adat van (alapértelmezett értéke: **http=80**; **https=443**);
- az **útvonal** a dokumentum elérési útvonala a kiszolgálón (és csatornán) belül (pl.: **documents**);
- a **dokumentum** a megjelenítendő dokumentum neve (pl.: **mainpage**);
- a **fájlkiterjesztés** az előbbi fájl típusának rövidítése (pl.: **html**).

VIII. HTML URL-kódolás

Nem-ASCII karakter URL-kódolása (az ISO-8859-1 HTML-entitás-száma alapján):

A karakter HTML-entitás-szám hexadecimális alakja (az ISO-8859-1 szerint): &#FF;

A karakter URL-kódja: %FF

Szóköz URL-kódja: +

IX. HTML-webszerverek

A számítógépünk webes gyökérmappájában lévő dokumentumok URL-je:

http://localhost/elérési_út/dokumentumnév.fájlkiterjesztés