



HTML alapismeretek

A webdesign tanfolyamon nem programozókat képzünk, így nem is kódolunk, programozunk le komplett weboldalakat, azonban a honlapkészítés alapjaival, azaz a **HTML** (HyperText Markup Language = hiperszöveges jelölőnyelv) és a **CSS** (Cascading Style Sheets = rangsorolt stíluslapok vagy egymásba ágyazott stíluslapok) fogalmával és gyakorlatban való működésével meg kell ismerkedned, hiszen ezek lesznek segítségedre akkor is, ha már WordPress weboldalakat készítesz WordPress sablonok átdizájnolásával. A HTML és CSS fájlok szöveges fájlok, a weboldalt megjelenítő eszköz, vagyis a webböngésző ezeket a szöveges fájlokat tölti le az internetről, értelmezi őket, és az eredmény a megjelenő weboldal lesz.

A HTML nyelv írja le a weboldal tartalmát, míg a CSS nyelv a weboldal formázásait.

Megjegyzés: ma már a legtöbb böngészőprogram támogatja ezeket a szabványokat, tehát helyesen jelenik meg a weboldal. Ha esetleg ugyanaz a weboldal máshogyan jelenik meg különböző böngészőprogramokkal megtekintve, akkor ez adódhat abból, hogy az oldal készítője nem tartotta be a megfelelő szabványokat, de sajnos az is előfordulhat, hogy a webböngésző program nem követi a szabványokat.

TUDÁSTÁR

A Web-alapú jelölőnyelvekre vonatkozó ajánlásokat a W3C konzorcium dolgozta ki.

A World Wide Web Consortiumot (W3C) 1994-ben alapította Tim Berners-Lee, a Web szülőatyja. A cél elsősorban az volt, hogy a webtechnológiák fejlesztésével foglalkozó vállalatok és kutatóintézetek a jövőben ne forgácsolják szét feleslegesen tudásukat, elért eredményeiket, hanem egymással vállvetve, egymást segítve tereljék ugyanabba az irányba az informatika újításait. A W3C célja, hogy minél jobban kihasználja az internet (web) nyújtotta lehetőségeket szabványok és iránymutatások kidolgozása révén, melyek egyúttal biztosítják a web hosszú távú fejlődését is.

W3C Magyar Iroda weboldala: <http://w3c.hu/>

HTML

A HTML (HyperText Markup Language – hiperszöveg jelölőnyelv) nyelv használatával tehát tartalmat publikálhatunk a világhálón. Fontos megemlíteni, hogy **leíró nyelv** és nem programozási nyelv, ugyanis **csak a weboldal felépítését hozhatjuk vele létre, a dokumentum tartalmát és struktúráját írhatjuk le vele**. Azaz azt, hogy ez itt egy bekezdés (egy logikailag összetartozó szövegegység), ez egy első szintű címsor, ezt a szöveget ki kell emelni, ez pedig egy rövidítés, stb.), dinamikus vagy intelligensebb viselkedést nem tudunk vele „előállítani”.

A HTML dokumentum **.html** vagy **.htm** kiterjesztésű.

A HTML leíró nyelv legújabb verziója a **HTML5**, amely a korábbi változatokhoz képest egyszerűbben alkalmazható, többféle lehetőséget (pl. interaktív alkalmazásokat) rejt magában. Azonban azt tudni kell, hogy interaktív alkalmazások fejlesztéséhez nem elég önmagában a HTML5 szabvány ismerete, szükséges lehet a JavaScript kliens oldali programozási nyelv és más szerveroldali nyelvek ismeretére is. Azonban ez már komoly programozás, nem képezi a tanfolyam tananyagának részét.

Alap HTML5-struktúra

A HTML három nyelvi elemből építkezik: vannak benne **elemek** (valamilyen tartalmat írnak le, pl. címsort, listát), **címkék** (angolul: tag, ejtése: teg, az elem kezdetét és végét jelölő címke, de utasításokat, jellemzőket is magában foglalhat) és **attribútumok** (elemhez kapcsolódó tulajdonságok, jellemzők). Ezekre hamarosan konkrét példákat is láthatsz.

Nézzük a HTML5 szabvány szerinti alapstruktúrát, amelyből az oldalak készítése során kell kiindulni!

Forráskód

```
1. <!DOCTYPE html>
2. <html lang="hu">
3.   <head>
4.     <meta charset="UTF-8">
5.     <title></title>
6.   </head>
7.   <body>
8.   </body>
9. </html>
```

1. sor: A HTML-kódot az alkalmazandó dokumentumtípus (DOCTYPE) megadásával kezdjük. Ezzel informáljuk a böngészőprogramot, hogy milyen szabvány szerint készítettük a dokumentumot, hogy az megfelelő módon tudja megjeleníteni.

Megjegyzés: Ha esetleg ezzel találkozunk az első sorban, akkor ez a HTML4 verziónak a dokumentumtípus-definíciója.

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

A dokumentumtípus megadása után már elkezdhettük írni a megfelelő HTML tageket, azaz a címkéket. Tagnek a < és > jelek között található utasítást nevezzük. A dokumentumtípus megadására szolgáló Doctype nem tekinthető HTML-tagnek, a HTML-állomány első tagjének (vagy gyökér elemének) a <html> taget nevezzük (2. sor). Egy tag hatása odáig tart, amíg le nem írjuk a tag záró párját, ami ugyanaz, mint a nyitó tag, csak egy per jel (/) vezet be. Például a <html> tag lezáró párja a </html> tag.

Megjegyzés: Később látni fogjuk, hogy vannak kivételek is, vagyis olyan tagek, amelyeknek nincs záró párjuk. Ilyen például egy kép beszúrására alkalmas tag is.

2. sor: Azonban itt most nem csak a <html> taget nyitjuk meg, hanem az oldal nyelvét is megadjuk, amely a mi esetünkben a magyar. Az oldal nyelvét a <html> tag lang paraméterében kell megadni, és magyar nyelvű tartalom esetén a hu kódot kell szerepeltetnünk.

3. sor: A <html> tag megnyitása és a nyelv megadása után a fejlécelemek megadása következik, melynek kezdetét a <head> címke jelzi.

4. sor: A fejlécelemek között kell megadni a karakterkódolást. Ahhoz, hogy a magyar nyelv ékezetes betűi helyesen jelenjenek meg a különböző operációs rendszerekben található különböző alkalmazásokban, megfelelő karakterkódolást kell beállítani. Ez a mi esetünkben az UTF-8 karakterkódolást jelenti.

Megjegyzés: HTML4 esetén ezzel a hosszabb sorral találkozhatasz:

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

Megjegyzés: Ügyeljünk arra, hogy nem elég csak a kódban szerepeltetni az UTF-8 kódolást, magát a HTML-állományt is ilyen kódolással kell elmenteni, ilyen karakterkódolásra kell beállítani az adott kódszerkesztő programban (pl. Notepad++-ban a Kódolás menüpont alatt)

Figyelmeztetés: Ha már egy meglévő állományt kell átalakítunk UTF-8 kódolásra, akkor a Kódolás/Átalakítás UTF-8 kódolásra BOM nélkül menüpontot válasszuk!



5. sor: A fejlécelemek között adjuk meg a lap/oldal címét, mely címet a <title> és a </title> címkék közé kell zárni. A <title> tagben megadott szöveg fog megjelenni a böngészőprogram címsorában.

Megjegyzés: Ha már rátérünk a weboldal formázásaira, azaz a CSS-re, akkor látni fogod, hogy a CSS fájlra is a fejlécben hivatkozunk.

6. sor: Lezárjuk a fejléct, melynek lezárását a </head> címke jelzi.

7. sortól: A fejléc lezárása után következik a dokumentumtörzs, melynek a <body> és </body> címkék közötti részt nevezzük. Ezen elemek között kell elhelyeznünk az oldal tényleges tartalmát, a szövegeket, hivatkozásokat, képeket, táblázatokat, stb. (Például az egyik leggyakrabban használt tag a <p>, amellyel bekezdéseket hozhatunk létre. Az oldal tartalma állhat akár csak egy mondatból, amelyet egy bekezdésben (paragrafusban) helyezhetünk el a <p> tag segítségével.)

Végül lezárjuk a <html> taget, láthatjuk, hogy a <html> tagen belül lett megnyitva a <body> tag is, de a <html> elemet csak a <body> elem lezárása után zárhatjuk le a </html> taggel.

Próbáljuk ki mindezt a gyakorlatban is! A kódszerkesztő programokat bemutató leckében már láhattad, hogy milyen egyszerűen ki lehet írni akár csak egy sort is a böngészőben. Ha nem emlékszel rá, lapozz vissza a leckéhez, és tekintsd meg az oktató videókat!

Nyisd meg a kódszerkesztőt, nyiss meg egy új fájlt, add meg, hogy a karakterkódolás UTF-8 legyen, a Nyelv pedig a HTML. Gépeld be az alább látható sorokat, mentsd el a fájlt a számítógépedre, a mentés során nevezd el pl. *index.html* -nek. A lementett fájlt nyisd meg a böngészőben – duplaklikk a fájlra, és annak tartalma a böngészőben jelenik meg.

```
<!DOCTYPE html>
  <html lang="hu">
    <head>
      <meta charset="UTF-8">
      <title>Első weboldalam</title>
    </head>
    <body>
      <p>Köszöntelek az első weboldalamon!</p>
    </body>
  </html>
```



Alap HTML tagekre már jó néhány példát láthattál az imént, most ezek listáját folytatjuk – a lista nem teljes, azokat mutatjuk be, melyeket a soron következő gyakorló feladatok során is alkalmazunk, illetve ezekkel általában gyakrabban is találkozhatasz:

Szövegformázáshoz kapcsolódó, szöveg szintű elemek

<i> – dőlt

`<p>Gondolj egy nagyot,<i>válts karriert</i>, webdesigner leszel!</p>`

** – hangsúlyos kiemelés**

`<p>Webdesigner szeretnék lenni.</p>`

** – félkövér betű**

`<p>Kérd azINGYENESleckéket, és vágj bele most!</p>`

** – erős kiemelés**

`<p>Figyelem!Már csak 2 nap maradt a jelentkezésig!</p>`

<u> – aláhúzás

`<p>A webdesigner a jövő szakmája: Vágj bele <u>Te</u> is!</p>`

<mark> – szövegkiemelés

`<p><mark>100% online</mark>oktatás és tanulás.</p>`

**
 – sortörés**

Manuális sortörést jelent. A tagnek nincs záró párja. Akkor használhatjuk, ha az egyébként tartalmilag összetartozó elemeket akarjuk külön sorokba tördelni. Ne használjuk a `
` elemet tétközként – Az elemek közti üres tétköz megadására a stíluslapok adnak lehetőséget a margók, belső kitöltések stb. definiálásával. Szintén nem helyes a `
` használata bekezdések helyett – az összetartozó gondolatokat bekezdésekbe (`<p>`) érdemes szervezni, nem pedig egy hosszabb bekezdés tartalmát `
` tagekkel széttörni.

`<p>Kiss Izsó
webdesigner
1111 Webdesignsuli, Tanulás útja 1.</p>`

<a> - hiperhivatkozás

Ezzel tudunk hiperhivatkozásokat (linkeket) létrehozni az egyes objektumok között, szövegekre, képekre tudunk hivatkozást tenni. A böngészők a linkeket alapesetben aláhúzással, illetve eltérő színnel jelölik.

href: értékként a hivatkozott oldal URL-jét adhatjuk meg, pl.:

```
<a href="http://webdesignsuli.hu/">Webdesignsuli honlapja</a>
```

azonban a href: értékkel hivatkozhatunk egy fotóra is, amely például egy *kepek* nevű mappában található, és ez a mappa egy közös mappában van a HTML fájlunkkal

```
<a href="kepek/webdesign.jpg"></a>
```

target: ezzel befolyásoljuk, hogy a link hova töltődjön be, pl.:

```
<a href="kepek/webdesign.jpg" target="_self"> - a fotó a jelenlegi böngészőben töltődik be
```

```
<a href="kepek/webdesign.jpg" target="_blank"> - a fotó új ablakban megnyílván töltődik be
```

Oldalszerkezet elemek

<body> - dokumentum törzse

Már a bevezetőben is láthattad, hogy a HTML oldalak két fő részre tagolódnak, a <head> és a <body> részre. A <body> és </body> címkék közötti részt nevezzük a dokumentum törzsének, ebben kell elhelyeznünk az oldal tényleges tartalmát, a szövegeket, hivatkozásokat, képeket, táblázatokat, stb.

<head> - fejléc

A head taggel fejléc adható meg, ezt már a bevezetőben is láthattad. Ez a rész tartalmazza a bevezető információkat a weboldalról, mint például az oldal címét és a karakterkódolást.

<header> - fejrész

A header taggel az a fejrész adható meg, melyben általában már a logó, a szlogen, a menüsor, stb. szerepel.

```
<header>
  <h1>A webdesigner a jövő szakmája.</h1>
</header>
```

<footer> - lábléc

A lábléc jelölésére alkalmas tag. A lábléc általában a weboldallal kapcsolatos információkat tartalmazza, mint például a készítő neve, kapcsolódó dokumentumok linkjei, szerzői jogi információk, stb.

```
<footer>
  <p>Webdesignsuli - Minden jog fenntartva.</p>
</footer>
```

<nav> - navigációs elem

Ez az elem az oldal azon részének van fenntartva, amely más oldalakra vagy az adott oldal egy bizonyos területére mutató linkeket (pl. a menüsört, menüpontokat) tartalmazza, és általában a <header> és </header> tagek közé kerül.

```
<nav>
  <h1>Menü</h1>
  <ul>
    <li><a href="#">Menüpont1.</a></li>
    <li><a href="#">Menüpont2.</a></li>
    <li><a href="#">Menüpont3.</a></li>
    <li><a href="#">Menüpont4.</a></li>
  </ul>
</nav>
```

("#" - az oldalon belül található menüpontokra ugrik)

<h1>, <h2>, <h3>, <h4>, <h5>, <h6> - címsorok

A címsorok (heading) segítségével tagolhatjuk a HTML oldal tartalmát fejezetekre és alfejezetekre, és a navigációt is segítik.

```
<header>
  <h1>A webdesigner a jövő szakmája.</h1>
</header>

<footer>
  <h3>Itt találsz meg minket!</h3>
</footer>
```

Csoportosító elemek

<p> - bekezdés

A <p> taggel már a bevezetőben megismerkedtünk. Segítségével a szöveget bekezdésekre oszthatjuk. Az adott bekezdés tartalmát a <p> és </p> elemek közé kell tenni.

<div> - blokk definiálása

A <div> elemnek nincs önálló szemantikai jelentése, azaz szerepe, nem fejt ki hatást a benne elhelyezett tartalomra. Arra használjuk, hogy struktúrákat hozzunk létre, csoportba foglaljunk nagyobb blokkokat, és ezeket majd könnyen meg tudjuk formázni a CSS-ben.

```
<body>
  <div class="proba1">
    <p>Lorem ipsum dolor sit amet.</p>
    <p>Phasellus suscipit arcu lacus, at volutpat nibh porttitor.</p>
  </div>
  <div class="proba2">
    <p>Lorem ipsum dolor sit amet.</p>
    <p>Phasellus suscipit arcu lacus, at volutpat nibh porttitor.</p>
  </div>
</body>
```

<hr> - vízszintes elválasztó

A <hr> tag elhelyezésével egy elválasztó vonalat jeleníthetünk meg. A <hr> tagnek nincs záró párja. Általában weboldalon belüli tagoló elemként szoktuk használni, például egymástól logikailag is elkülöníthető szövegrészek között.

```
<h1>Ez az oldal fő címsora</h1>
<p>Ez egy bekezdés</p>
<hr>
<p>Ez egy újabb bekezdés</p>
```

, - sorszámozott lista

Sorszámozott lista (1., 2., stb) esetén az és tagek közé kell elhelyeznünk a listaelemeket, amelyeket a tagek (lista) vezetnek be.

```
<ol>
  <li>első listaelem</li>
  <li>második listaelem</li>
  <li>harmadik listaelem</li>
</ol>
```

, – felsorolás lista

Felsorolás lista (szimbólummal, pl. pontokkal ellátott lista) esetén az és tagek közé kell elhelyeznünk a listaelemeket, amelyeket a tagek (lista) vezetnek be.

```
<ul>
  <li>felsoroláselem</li>
  <li>felsoroláselem</li>
  <li>felsoroláselem</li>
</ul>
```

<dl> <dt> <dd> – definíciós lista

Ezen elemekkel úgynevezett definíciós, vagy meghatározás listát készíthetünk (fogalmak meghatározásához). A <dl> és </dl> tagek közé kell elhelyeznünk a listaelemeket. A fogalmat <dt> és </dt>, a magyarázatát <dd> és </dd> tagek közé kell tennünk.

```
<dl>
  <dt>Fogalom1</dt>
  <dd>MagyarázatMagyarázatMagyarázat</dd>
  <dt>Fogalom2</dt>
  <dd>MagyarázatMagyarázatMagyarázat</dd>
</dl>
```

Táblázatok

Táblázatokat gyakran használunk a szövegszerkesztési feladatok során is, hogy az adatokat átlátható módon, sorokba és oszlopokba rendezve prezentáljuk. A táblázat megadása a <table> tag használatával lehetséges. A <table> és </table> tageken belül a <tr> és </tr> elemek között adhatjuk meg a táblázat sorait. A <td> elem a sorokon belüli cellák létrehozására szolgál.

```
<table>
  <tr>
    <td>1. sor 1. elem</td>
    <td>1. sor 2. elem</td>
    <td>1. sor 3. elem</td>
  </tr>
  <tr>
    <td>2. sor 1. elem</td>
    <td>2. sor 2. elem</td>
    <td>2. sor 3. elem</td>
  </tr>
</table>
```

Lehetőségünk van továbbá a táblázat celláinak egyesítésére is. Az oszlop celláinak egyesítéséhez a `rowspan=""` attribútumot használjuk, míg a sorokéhoz a `colspan=""` -t. Értékük mindig annyi, ahány cellát egyesíteni szeretnénk.

```
<table>
  <tr>
    <td rowspan="2">Első oszlop, egy közös cella</td>
    <td colspan="2">Második oszlop, két egyesített cella</td>
  </tr>
  <tr>
    <td>Második sor, első cella</td>
    <td>Második sor, második cella</td>
  </tr>
</table>
```

Beágyazott elemek

** – kép beszúrása**

A képek oldalunkra történő beágyazására használhatjuk. Külön lezáró taggel nem rendelkezik, szóközzel és / jellel végezzük el a lezárását. A legfőbb, hogy a kép elérési útvonalát adjuk meg – pl. ez a kép a *kepek* nevű mappában található, innen hívja be a HTML fájl, amely alapesetben a számítógépünkön egy közös mappában van ezzel a *kepek* nevű mappával:

```

```

Továbbá meg kell adni a helyettesítő szöveget, a kép leírását, azaz az `alt` -ot, amely akkor jelenik meg, ha a böngésző valamiért nem tudja megjeleníteni a képet, és keresőoptimalizálás (erről lesz szó a későbbiekben) szempontjából is hasznos:

```

```

És olyan attribútumokat is megadhatunk, hogy milyen méretben jelenjen meg a kép, mekkora legyen a szélessége és a magassága:

```

```