

HTML 5 alapismeretek

Bevezetés a HTML 5-be

Mi is az a HTML?

A HTML (HyperText Markup Language = hiperszöveges jelölőnyelv) egy leíró nyelv, amelyet direkt a weblapok létrehozására fejlesztettek ki, valamint szabványosítottak aztán.

Fontos megemlíteni, hogy leíró és nem pedig programozási nyelv. Sokszor említik programozási nyelvként, holott ez téves, hiszen csak a weboldal felépítését (layout- ját) hozhatjuk vele létre, intelligensebb viselkedése nem tudjuk használni.

És miért is HTML 5?

Az 5 csupán annyit jelent, hogy ez a nyelv 5. verziója. A korábbi változatokhoz képest itt már próbáltak a készítők az egyszerűsége és jobb alkalmazhatóságra törekedni. Ezáltal több lehetőséget ad a kezünkbe a nyelv és a tanulhatósága is egyszerűbb a korábbi verziókhöz képest. A tananyagban mi is pont ezért fogjuk ezt a verziót használni, mert könnyű elsajátítani és jelenleg ez a legmodernebb HTML nyelv is, mellyel dolgozhatunk.

Fájl kiterjesztés, nyelvi szintaxis

A HTML dokumentumuk általában *.html*, vagy *.htm* kiterjesztéssel rendelkeznek. Néhány speciális esetben el lehet ettől térni, de erről majd kicsit később.

Tartalmuk értelmezhető szöveg, mind a böngésző, mind az ember számára. Ezért a weblapok könnyen létrehozhatóak, bármilyen szöveg mentésre alkalmas programmal. (A mi esetünkben az előző részben említett PSPad-el)

Felépítését úgynevezett tagekre (ejtsd: teg) bontjuk, amelyeknek összessége fogja adni a DOM-ot (Document Object Model), azaz egy logikusan felépített dokumentum struktúrát.

Lássuk, mik azok a szempontok, amelyeket be kell tartanunk egy HTML kód írásánál:

- Rendelkeznie kell a dokumentumnak a `<!DOCTYPE>`, `<html>`, `<head>`, `<body>` tegekkel
- Minden teget le kell zárni. Önálló elemeket egy szóközzel és egy `/` jellel, például: `
`
- A jellemzőiket időjelek között kell megadni, például: ``
- Nem kötelező, de nagyban megkönnyíti munkákat, és azokét is akik utánunk szerkesztik a dokumentumot, ha egy jól átlátható formában írjuk meg a kódunkat. Azaz használunk behúzásokat, szellős sorokat. A későbbi példákban erre látni is fogsz mintákat.

Nyitó és záró tagek

Lássuk is egy egyszerű HTML dokumentumot példaként:

```
<!DOCTYPE HTML>
<html lang="hu">
  <head>
    <meta charset="UTF8" />
    <title>Az oldal címe</title>
  </head>
  <body>
    <h1>Köszöntelek a weboldalamon!</h1>
  </body>
</html>
```

Nézd meg a kapcsolódó oktató videót!

A `<p></p>` tagek kivételével amit látunk, az mind kötelező eleme egy HTML fájlnek. Minden HTML dokumentum kell, hogy tartalmazza ezeket, hogy a böngészők megfelelően tudják megjeleníteni a `<body>` tegeken belüli részeket.

Vegyük górcső alá az egyes tegek jelentését:

- A `<!DOCTYPE HTML>` fogja megadni a böngészőnek, hogy neki most egy HTML fájl kell megjelenítenie.
- Maga a HTML dokumentumunk értelmezendő része kerül a `<html>` `</html>` elemek közé. Ennél az elemnél megadhatunk egy olyan jellemzőt is, amivel a szöveges tartalmunk nyelvét definiálhatjuk, az országkód megadásával. A mi esetünkben ez *hu* lesz.
- A `<head>` `</head>` rész a dokumentumunk fejléce. Ebbe kerülhet számos olyan dolog, amit weboldalunk megjelenéséért és kiegészítő adataiért felelős.
 - A `<meta charset="UTF8" />` -al utasítjuk a böngészőt, hogy milyen karakterkódolással jelenítse majd meg a szöveges tartalmakat. Az UTF8 egy olyan kódolást takar, amely szinte minden nyelvhez megfelelő, mivel biztosítja az olyan speciális karakterek megjelenítését is, mint a magyar ékezetes karakterek.
 - Oldalunk címét a `<title>` `</title>` tagek között adhatjuk meg. Ez fog megjelenni a böngésző címsorában, vagy lent a tálcán, illetve ha valaki egy keresővel talál az oldalunkra, akkor a kereső is ezt fogja címként megjeleníteni.
- Weboldalunk törzsét a `<body>` `</body>` elemek adják. Ide kerül a tényleges tartalmi rész, ami már a böngészőben fog megjelenni.
- A példában látható `<p>Köszöntelek a weboldalamon!</p>` szöveg is már a böngészőnkben fog feltűnni.

Alap HTML tagek

Lássuk mik azok a HTML tagek, amelyeket a leggyakrabban fogunk használni egy weboldal létrehozásánál:

Dokumentum: <html>

Megadja a böngészőnek, hogy a közötté lévő tartalmat HTML-ként kell értelmeznie, és annak szabványa szerint megjelenítenie. Minden esetben kell, hogy tartalmazza a dokumentumunk ezt az elemet. Lezárása mindig kötelező a </html> taggel.

Fejléc: <head>

A dokumentum fejléce. Számos információt helyezhetünk el benne, vagy külső fájlokat hívhatunk meg, például CSS, vagy JavaScriptet. Ugyan nem kötelező a használata, viszont erősen ajánlott. Lezárása mindig kötelező a </head> taggel.

Cím: <title>

A weboldalunk címét adjuk meg vele. Szintén nem kötelező elem, de ajánlatos az oldalunk címét beállítani vele. Lezárása mindig kötelező a </title> taggel.

Példa a tag használatára:

```
<title>A saját weboldalam</title>
```

Törzs: <body>

A weboldalunk tartalmi része. Ez ebben foglaltakat fogja a böngésző megjeleníteni. Minden esetben kell, hogy tartalmazza az oldalunk ezt az elemet! Lezárása mindig kötelező a </body> taggel.

Paragrafus: <p>

Bekezdés, vagy paragrafus. Egy összefüggő szövegblokkot definiálunk vele. Alap esetben minden bekezdés új sorban kezdődik, amit ezzel a taggel fogunk közre. Segítségével egyszerűen tudunk nagyobb szöveget tördelni, bekezdésbe foglalni. Lezárása mindig kötelező a </p> taggel.

Példa a tag használatára:

```
<p>Szöveges tartalom első bekezdése</p>  
<p>Szöveges tartalom második bekezdése</p>
```

Címsorok: <h1>, <h2>, <h3>, <h4>, <h5>, <h6>

Alkalmazásukkal különböző betűméretű címeket készíthetünk, ahol a <h1> lesz a legnagyobb, még a <h6> a legkisebb betűméretű. Például, ha egy témának szeretnénk főcímet adni, akkor használjuk a <h1> taget, még az egyes fejezetek alcíméhez a <h2> taget. Lezárása mindig kötelező a </h1>, </h2> ... taggel.

Példa a tag használatára:

```
<h1>Tartalmi rész főcíme</h1>  
<h2>Tartalmi rész alcíme</h2>
```

Képek:

Ezzel a tagel képet szúrhatunk be arra a helyre, ahova elhelyeztük az elemet. A kép elérési útját az `src=""` paraméterrel kell megadnunk. Külön lezáró taggel nem rendelkezik, lezárását szóköz / jel használatával kell elvégeznünk.

Példa a tag használatára:

```

```

Az taghez különféle attribútumokat is megadhatunk. A leggyakoribbak a következők, melyekre szükségünk lehet:

`width` - Megadjuk vele a kép vízszintes méretét, pixelben.

`height` - Megadjuk vele a kép függőleges méretét, pixelben.

`alt` - Megadhatunk vele egy helyettesítő szöveget, ami akkor jelenik meg, ha valamiért a böngésző nem tudja megjeleníteni a képet

Például ha képünket 300x200 pixeles méretben szeretnénk megjeleníteni, helyettesítő szöveggel:

```

```

Linkek: <a>

Az <a> tag használatával tudunk szövegekre, képekre, vagy szinte bármilyen másik HTML tagre hivatkozást elhelyezni. Magát a linket a `href=""` attribútum értékeként kell megadni. Lezárása mindig kötelező a taggel.

Példa a tag használatára:

```
<a href="http://google.hu">Keresés a Google-el</a>
```

Ezzel elérjük, hogy a Keresés a Google-el szöveg klikkelhető lesz, és arra klikkelve megnyílik a <http://google.hu> oldal.

Továbbá azt is befolyásolhatjuk, hogy az újonnan megnyíló oldal a jelenlegi helyén, vagy új oldalként nyíljon meg a böngészőben. Ennek beállítására a `target=""` attribútumot használjuk.

Leggyakoribb értékei a következők:

`_blank` - Link megnyitása új ablakban, vagy fülön

`_self` - Link megnyitása azonos oldalon

Például ha a Google keresőt új ablakban szeretnénk megnyitni a linkre való klikkeléskor:

```
<a href="http://google.hu" target="_blank">Keresés a Google-el</a>
```

Listák: , →

A HTML nyelv háromféle lista létrehozását teszi nekünk lehetővé:

Rendezett, azaz sorszámozott lista.

Rendezetlen, azaz egy szimbólummal (kör, négyzet, kötőjel, ...) ellátott lista.

Definiálós, azaz meghatározásból és leírásból álló lista.

Rendezett lista

Jelölésére az taget használjuk, mely az angol Ordered List (Rendezett Lista) kifejezésből ered.

A lista elemek tartalmát a tagek közé vesszük fel, amely lehet szöveg, kép, vagy további HTML tag.

Példa egy egyszerű rendezetlen listára:

```
<ol>
    <li>Lista első eleme</li>
    <li>Lista második eleme</li>
    <li>Lista harmadik eleme</li>
</ol>
```

1. Lista első eleme
2. Lista második eleme
3. Lista harmadik eleme

Rendezetlen lista

Jelölésére az taget használjuk. A lista elemeit pedig ugyanúgy tagekkel definiáljuk mint a rendezetlen lista esetében.

Példa egy egyszerű rendezetlen listára:

```
<ul>
    <li>Lista első eleme</li>
    <li>Lista második eleme</li>
    <li>Lista harmadik eleme</li>
</ul>
```

- Lista első eleme
- Lista második eleme
- Lista harmadik eleme

Definíciós lista

Az előző két listától eléggé eltér, de mégis a listák közé tartozik. Jelölésére a <DL> taget használjuk. A listán belüli elemek pedig mindig két részből állnak: meghatározásból (DT) és leírásból (DD).

Általában fogalmak magyarázatához szoktuk használni ezt a lista típust.

Példa egy egyszerű definíciós listára:

```
<dl>
    <dt>Fogalom egy
    <dd>Fogalom egy magyarázata
    <dt>Fogalom kettő
```

```
<dd>Fogalom kettő magyarázata  
</dl>
```

```
Fogalom egy  
    Fogalom egy magyarázata  
Fogalom kettő  
    Fogalom kettő magyarázata
```

A rendezett és rendezetlen listáknál lehetőségünk van az egyes lista elemekhez alpontokat definiálnunk. Ezek segítségével létrehozhatunk komolyabb tartalomjegyzékeket, vagy menüket is akár.

Példa egy többszintű, rendezetlen listára:

```
<ul>  
  <li>Lista elem 1</li>  
  <li>Lista elem 2  
    <ul>  
      <li>Lista elem 2 első alpontja</li>  
      <li>Lista elem 2 második alpontja</li>  
    </ul>  
  </li>  
  <li>Lista elem 3</li>  
</ul>
```

- Lista elem 1
- Lista elem 2
 - Lista elem 2 első alpontja
 - Lista elem 2 második alpontja
- Lista elem 3

Szöveg formázás: , , <u>,

Szöveg formázásához számos taget biztosít a HTML nyelv. Lezárásuk mindig kötelező és a hatásukat a közrefogott szövegen fejtik ki.

A leggyakoribbak amelyekkel találkozni fogunk:

```
<strong> - Félkövér  
<em> - Dőlt betűs  
<u> - Aláhúzott
```

Példa a szövegformázások használatára:

```
<strong>Ez a szövegrész vastagon szedett</strong>, még az <em>alábbi  
szavak dőlten vannak szedve</em>. <u>Ez a rész pedig át van  
húzva.</u>
```

Ez a szövegrész vastagon szedett, még az alábbi szavak dőlten vannak szedve.

Ez a rész pedig át van húzva.

**Sortörés:
**

A hosszabb sorokat a böngésző automatikusan tördeli, de néha szükség lehet manuális sortörés használatára, vagy üres sorok beszúrására a szövegben. Ehhez használatos a
 tag. Nem rendelkezik külön lezáró taggel, így szóköz / jellel kell ellátnunk.

Példa a tag használatára:

Ez a szöveg rész az első sorba kerül,
 még ez a másodikba.

**Ez a szöveg rész az első sorba kerül,
még ez a másodikba.**

**Tárolók: <div>, **

Ezek a tagek különleges szerepet tölt be a HTML nyelvben, hiszen az eddig olvasatoktól eltérően, ez a két elem semmilyen hatást nem fejt ki a benne elhelyezett tartalomra. Használatukkal egy általános módszert biztosítanak, amelyekkel saját struktúrákat, vagy formázásokat hozhatunk létre. Ezeket aztán CSS-el, vagy JavaScript segítségével tudjuk az igényeinknek megfelelően alakítani.

Ezekkel az elemekkel behatóbban a CSS alapok részben fogunk megismerkedni.

Táblázatok: <table> → <tr>, <td>

Sokszor előfordul, hogy táblázatba foglalt tartalmat kell létrehoznunk. Szerencsére a HTML ehhez is biztosít számos olyan taget, amellyel bármilyen bonyolultságú táblázatot létrehozhatunk.

Magát a táblázatot a <table> taggel definiáljuk, még ezen belül a <tr> taggel az egyes sorokat hozhatjuk létre. A sorokat aztán tetszőleges számú cellára bonthatjuk a <td> taggel.

Példa egy egyszerű, két soros, két oszlopos táblázat létrehozására:

```
<table>
  <tr>
    <td>Első sor, első cella</td>
    <td>Első sor, második cella</td>
  </tr>
  <tr>
    <td>Második sor, első cella</td>
    <td>Második sor, második cella</td>
  </tr>
</table>
```

```
</tr>  
</table>
```

Első sor, első cella	Első sor, második cella
Második sor, első cella	Második sor, második cella

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

Egy egyszerű példa a cellák egyesítésére:

```
<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>
```

Első oszlop, egy közös cella	Második oszlop, két egyesített cella	
	Második sor, első cella	Második sor, második cella

Űrlapok: <form>

Az űrlapok segítségével küllemféle adatbeviteli módokat valósíthatunk meg a weblapunkon. Az űrlap elemei lehetnek látható, vagy rejtett elemek is. Látható elemek például a szövegbeviteli mezők, gombok, legördülő listák, rádiógombok, vagy jelölőnégyzetek. A rejtett mezőkben pedig olyan adatokat tárolhatunk, amelyet az űrlapot feldolgozó programnak szánunk és nem szeretnénk, hogy ezeket az adatokat a felhasználó is lássa.

Miután egy űrlap kitöltésével végeztünk és submitoltuk (elküldtük feldolgozásra, mondjuk egy Mehet gomb lenyomásával) a feldolgozó program végighalad a mezőkön és azok értékein, majd végrehajtja az általunk kívánt műveleteket rajtuk. Például elmenti a beírt adatokat adatbázisba, vagy elküldi őket e-mailben.

Az űrlapok adatainak feldolgozásával részletesebben a PHP alapok részben fogunk megismerkedni.

Magukat az űrlap elemeket a <form> tagek között adjuk meg. Minden a tagen belül az űrlap része lesz.

Űrlap elemek, amelyeket a leggyakrabban alkalmazunk:

Szöveges beviteli mezők

Egyszerű szöveges bevittelt az `<input type="text" name="mezőnév" />` tagel valósítunk meg. Külön lezáró teggel nem rendelkezik, így `/` jellel a végén zárjuk.

Ha egy szöveges mezőt jelszó bevitelre szeretnénk használni, akkor az `<input type="password" name="mezőnév" />` formában definiáljuk azt. Ekkor a beírt szöveget a böngésző pontokkal fogja helyettesíteni, biztosítva ezzel, hogy mások ne tudják leolvasni a jelszót miközben azt begépeljük.

Mindkét esetben látható, hogy a mezők típusát a `type=""` attribútummal definiáljuk, még a `name=""` attribútummal egy nevet adhatunk a mezőnek. Ez utóbbi segítségével tudjuk ugyanis a későbbiekben kinyerni az adott mező értékét a feldolgozáshoz.

Lehetőségünk van a mezőknek előre beállítani értéket. Ezt a `value=""` attribútum megadásával érhetjük el. Például:

```
<input type="text" name="neved" value="János" />
```

Rejtett mezők

Ritkán, de szükségünk lehet ilyen mezőre is, ahol az adatot tárolnunk kell, viszont nem szeretnénk, hogy azt a felhasználó lássa, vagy módosítsa. Ilyen mező létrehozására a `type` attribútumnak a `hidden` értéket mell megadunk:

```
<input type="hidden" name="mezőnév" value="érték" />
```

A mező létezni fog a HTML dokumentumunkban, de az semmilyen vizuális formában nem fog megjelenni, viszont az űrlap elküldésekor ez a mező érték is elküldésre kerül.

Gombok

Alapvetően négyféle gombot hozhatunk létre, ezekből a leggyakrabban az űrlap adatainak elküldésére használatos submit gombot használjuk.

A gomb típusokat, hasonlóan a beviteli mezőkhöz, a `type` attribútumom értékének megadásával definiáljuk.

Lássuk a rendelkezésünkre álló gomb típusokat:

```
<input type="submit" value="Elküldés">
```

Megnyomásával elküldi az űrlap minden mezőjének értékét a feldolgozó programnak.

```
<input type="button" value="Megnézem">
```

Egyszerű gomb, melynek megnyomására semmilyen történés nem megy végbe. Funkciót például JavaScript segítségével programozhatunk neki.

```
<input type="reset" value="Újra">
```

Megnyomásával az űrlap minden elemét alaphelyzetbe állíthatjuk. Hasznos, ha nem szeretnénk minden adatot kitörölni, hogy előről kezdjük a kitöltést.

```
<input type="file" value="Feltöltés">
```

Fájl feltöltésre használjuk. Megnyomásával kitallózhatunk egy fájlt a gépünkről, amelyet aztán a feldolgozó programunk feltölt a tárhelyünkre.

Legördülő listák

Legördülő listát a `<select>` tagel hozhatunk létre. Lezárása a `</select>` tagel történik. A

kiválasztható elemeket pedig az <option> tagek között adjuk meg.

Példa a használatára:

```
<select name="választás">
  <option value="1">1. lehetőség</option>
  <option value="2">2. lehetőség</option>
  <option value="3">3. lehetőség</option>
</select>
```

Természetesen az az érték kerül feldolgozásra az űrlap elküldésekor, amelyet kijelöltünk a listából.

Szöveg területek

Nagyobb szöveges területet a <textarea> tagel hozhatunk létre. Ebbe akár több sornyi szöveget is írhatunk. Ideális hozzászólások szövegének használatához, vagy olyan helyeken, ahol a felhasználó kifejtheti a mondanivalóját.

Rendelkezik külön lezáró taggel: </textarea>

Tehát szöveges mezőt az alábbi példa szerint vehetünk fel:

```
<textarea name="vélemény"></textarea>
```

Jelölő négyzetek

Jelölő négyzeteket az <input type="checkbox" name="opció" value="érték" /> formában hozhatunk létre. Bármennyit bejelölhetünk belőlük (akár mindet), de üresen is hagyhatjuk őket. Ha a jelölőnégyzeteket azonos névvel látjuk el, akkor feldolgozáskor az értékeiket vesszővel elválasztva kapjuk vissza.

Lehetőségünk van továbbá előre bejelölté tenni ezeket a négyzeteket, a checked attribútumom megadásával.

Példa:

```
<input type="checkbox" name="kedvenc_gyumolcsok" value="Alma"
checked="checked"/>Alma
<input type="checkbox" name="kedvenc_gyumolcsok" value="Barack"
/>Barack
<input type="checkbox" name="kedvenc_gyumolcsok" value="Körte"
/>Körte
```

A fenti példában akár mindegyik gyümölcsöt kiválaszthatjuk mint kedvenc étel, viszont az Alma opció már alpból bejelölésre kerül. Mivel azonos névvel láttuk el őket, így a feldolgozó programunk ezen az egy néven kap meg minden kijelölt checkbox értéket.

Rádió gombok

A rádiógombok nagyon hasonlóan működnek a jelölőnégyzethez. Deklarálásuk a type="radio" attribútummal történik. Viszont ha névnek azonos értéket adunk meg nekik, akkor kiválasztásnál maximum egyet jelölhetünk be. Előre bejelöléshez itt is a checked attribútumot kell alkalmaznunk.

Példa:

```
<input type="checkbox" name="nemed" value="férfi"
checked="checked"/>Férfi
<input type="checkbox" name="nemed" value="nő" />Nő
```

A példa egy eldöntendő opciót fog a rendelkezésünkre bocsájtani. Vagy Férfi, vagy Nő lehetőséget tudjuk kiválasztani., kettőt egyszerre nem.

Példa egy egyszerű regisztrációs űrlapra:

```
<form>
    Felhasználó: <input type="text" name="felhasznalo" />
    Jelszó:      <input type="password" name="jelszo" />
    Nemed:      <input type="checkbox" name="nemed"
value="férfi" />Férfi
               <input type="checkbox" name="nemed"
value="nő" />Nő
    Életkor:    <select>
                 <option value="18_alatt">18 év
alatti</option>
                 <option value="18_felett">18 év
feletti</option>
               </select>
    Hírlevél:   <input type="checkbox" name="hirlevel"
value="igen" />Kérek
    Megjegyzés: <textarea name="megjegyzes"></textarea>
    <input type="submit" value="mehet" />
</form>
```

Felhasználó:

Jelszó:

Nemed:

Férfi Nő

Életkor:

18 év alatti ▼

Hírlevél:

Kérek

Megjegyzés:

Mehet

Feladat: Egy egyszerű képgaléria weboldal elkészítése az eddigi ismeretek alapján

Most, hogy megismerkedtünk az alapvető HTML elemekkel, készítsünk a felhasználásukkal egy egyszerűbb weboldalt, ami ebben a feladatban egy kis képgaléria lesz:

Képgaléria

Képek megnyitásához kattints a kisképekre



2014 - Az én nevem

Ahogy a HTML kódban is látszik, mindent nagyon egyszerűen, az alap HTML tagekből építettünk fel.

A <h1> elemek között az oldalunk címét helyeztük el, hogy jó látható legyen. Majd ezt követően egy paragrafusba egy rövidke útmutatást a látogatónak, hogy a képeket ki is nagyíthatja, ha szeretné.

Magukat a képeket egy 3x2-es táblázatba foglaltuk, hogy szépen illeszkedjenek egymás mellett/alatt, valamint megadtuk, hogy maximum 250 pixel szélesek legyenek (ha nem adunk meg magasságot, akkor az arányosan fog igazodni a megadott szélességhez).

Minden képet belinkeltünk önmaga fájljához, amelyek klikkeléskor új ablakban nyílnak meg, persze ekkor már eredeti méretükben.