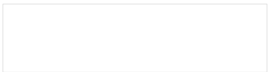


1. Mitől lesz web a web?

A web leíró nyelve a HTML. Ez egy mozaikszó, a Hypertext Markup Language angol kifejezés rövidítése, amelyet így fordítunk magyarra: hiperszöveges jelölőnyelv.

De mitől is lesz hiper egy szöveg? A kapcsolattól. Az egész web lényege, hogy a rajta elhelyezett tartalmak linkek segítségével összekapcsolódnak, és kesze-kusza nagy egésszé állnak össze. A következő feladatod tehát az lesz, hogy elkészíts egy mini webet. Ez érdekesen hangzik, igaz?

Mielőtt azonban belevágnál, nézzük meg, hogyan lehet linkeket elhelyezni egy HTML oldalon. Először készítünk egy linket a tanfolyam weblapjára:

``

``

← A weblap, ahová megyünk.

← Ide tesszük, hogy mire kell majd kattintani.

Link beillesztése

A fenti ábrán láthatod, hogyan hozzuk létre a linkeket. Meg kell adnunk a weblap címét, ahová el szeretnénk jutni, és meg kell adnunk azt, hogy mire kattintva jussunk el oda. Az alábbi példában a *Kattints ide* szövegre kattintva érjük el a tanfolyam weboldalát:

``
Kattints ide!
``

← Szövegre is kattinthatunk.

Példa szöveges linkre

Azonban nem csak szöveget, de képet is elhelyezhetünk a linkünkben, erre láthatsz példát a következő ábrán:

```
<a href="http://webdesigntanfolyam.com">
```

```

```

```
</a>
```

← Kép is kerülhet ide.

Példa képes linkre

Mindig csak egy dolog kerülhet egy linkbe! Ha több dolgot is szeretnél ugyanarra a weblapra linkelni, akkor azt tedd két külön linkbe, még akkor is, ha közel vannak egymáshoz a weblapon:

```
<a href="http://webdesigntanfolyam.com">Kattints ide!</a>
```

```
<a href="http://webdesigntanfolyam.com"></a>
```

A fenti példákön is jól látszik, hogy a képes link kódja sokszor nehezen áttekinthető kódot ad eredményül, ezért amíg a kisujjadban nincs a HTML-kódolás, érdemes először mindig szöveges linket készíteni, és ha az jól megy, akkor a szöveg helyére bemásolni a már szintén jól működő `` kódját. Ezzel a módszerrel sok-sok hibakereséssel töltött időtől kímélheted meg magad.

1. feladat: linkek más weboldalakra

Készíts egy weblapot, amelyen megjelenik az *Online Webdesign Tanfolyam* kifejezés, és erre kattintva jöjjön be a tanfolyam weboldala. Ez legyen az első `<p> ... </p>` paragrafusban. A második paragrafusban jeleníts meg egy *Facebook* ikont, és erre linkeld a tanfolyam Facebook csoportjának címét: <https://www.facebook.com/groups/webdesigntanfolyam/>

A weblap mappaszerkezetét a szokásos módon építsd fel: készíts egy új mappát a feladat számára, ebbe helyezd el a *css* és az *img* mappáidat, illetve a HTML fájldat is. A kép kerüljön az *img* mappába, a *css* mappa egyelőre legyen üres, de majd később használni fogjuk, ezért készíts el minden mappát most, egy kalap alatt!

Ügyelj a szép kódolásra is: a két-két `<p>` és `</p>` legyen egymás alatt, az `<a ...>` címke pedig kezdődjön beljebb! Ne felejtse el, hogy a weblapot *.html* kiterjesztéssel kell elmenteni!

Működik? Remek! Esetleg valami gond van a képpel? Akkor nyisd meg a második leckét, ott tippeket kaphatsz a hibakereséshez.

Többoldalas weblapok

A nagyobb weblapok nem csak egy, hanem több oldalból állnak. Általában van egy nyitóoldaluk, és vannak aloldalaik. A weblap nyitóoldalát `index.html`-nek kell elnevezni. Pontosabban elnevezheted másnak is, de amikor a böngésző letölti a webszerverről egy weblap címéhez tartozó fájlokat, a nyitóoldalt mindig `index.html` néven fogja keresni, és nagy pech, ha nem találja.

Fontos kérdés, hogy hogyan hozunk létre több lapos weboldalakat. Az elv nem is olyan bonyolult: először létrehozunk több, egymástól különálló weboldalt. Majd linkek segítségével kapcsolatot teremtünk köztük. Az alábbi példában az *Elérhetőségek* témájú oldalt fogjuk behívni:

```
<a href="elerhetosegek.html">  
  Elérhetőségek  
</a>
```

← Az elérhetőség szövegre kattintva bejön a weblap.

Saját weblapot linkelünk

Ahhoz, hogy a fenti kód működjön, az *elerhetosegek.html* oldalnak ugyanabban a mappában kell lennie, mint ahol a hívó weblap is található. Ha nem ott van, akkor azt a pontos útvonalat kell megadni, ahol a weblap elérhető. Hamarosan létre fogunk hozni egy többoldalas weblapot, de előtte még megismerkedünk egy-két hasznos tudnivalóval.

E-mail cím linkelése

E-mail címet is megadhatunk a linkben, ekkor a linkben megadott szövegre vagy képre kattintva elindul a felhasználó alapértelmezés szerinti levelezőprogramja:

```
<a href="mailto:info@mailcim.com">  
  E-mail cím  
</a>
```

← E-mail címet is meg lehet adni.

Ha e-mail címet linkelünk, kattintásra elindul a levelezőprogram

Telefonszám és Skype bekötése

Az alábbi kód segítségével telefonszámot tudsz hívni, vagy *Skype*-hívást kezdeményezhetsz a weblapon keresztül:

```
<a href="tel:+36123456789">  
  (36) 123456789  
</a>
```

← Telefonszám hívása

```
<a href="callto:0123456789">  
  0123456789  
</a>
```

← Skype hívása

Telefonszám és Skype hívás

Megjegyzés: a fenti linkek abban az esetben működnek a felhasználó gépén megfelelően, ha fel van telepítve, és az adott funkcióhoz hozzá van rendelve a nekik megfelelő alkalmazás.

2. feladat: e-mail cím linkelése

Az első feladatban létrehozott fájlokat fogjuk használni, ezért most nem kell új mappaszerkezetet létrehozni, hanem nyugodtan folytathatod a már megkezdett fájlokat. Első részfeladatod, hogy keress a neten egy e-mail ikont, és mentsd el az *img* mappádba. Ez után az első feladatban készített weblapra helyezd el az ikont, és ellenőrizd, hogy jól működik-e. Ha rendesen megjelenik a kép, linkelj rá egy email-címet a fent megadott módon.

A fenti példák alapján kipróbálhatod a telefonszám és a Skype-hívást is!

Fájl letöltése

```
<a href="mappa/fájlneve.kiterjesztése" download> ← A pontos elérést kell megadni.  
  Fájl letöltése  
</a>
```

Fájlok letöltéséhez a *download* jellemzőt is meg kell adni

3. feladat: fájl letöltése

Másolj egy képet az *img* mappába, majd tedd a weblapon letölthetővé! Megpróbálkozhatasz pdf fájl letöltésével is.

Új böngészőlap megnyitása

```
<a href="http://webdesigntanfolyam.com" target="_blank">  
    Kattints ide, és új lapon fogok megjelenni!  
</a>
```

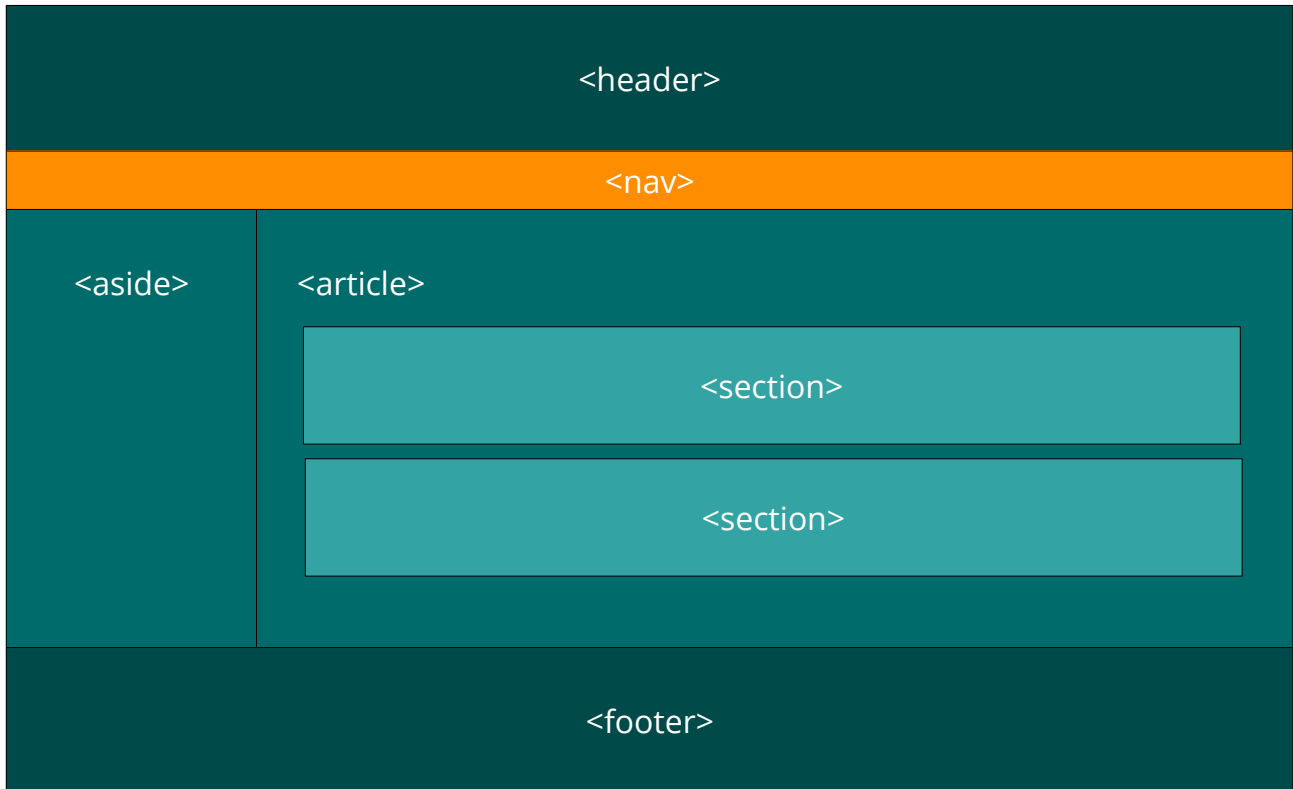
← Az _ jel fontos!

A target="_blank" új böngészőlapot nyit

Még hosszan sorolhatnánk az <a ...> címke erényeit, de a leghasznosabb funkciókkal megismerkedtünk. A továbbiakban a HTML törzsében lévő kódot fogjuk kicsit ráncbaszedni.

2. A HTML törzs szerkezete

A weblapok különféle részekből épülnek fel: fejléc, menüsor, oldalsávok, lábléc stb. Képzeld el egy weblapot, amelynek ilyen a szerkezete:



Nézzünk a fenti szerkezetre egy nagyon egyszerű példát:



Ahogy látod, ez nem a címlap, hanem a *Hangszereink* aloldal. A címsorban a cég nevét, a narancssárga sávban a menüsört, a jobb oldali oszlopban a termékek leírását, a lábjegyzetben pedig ismét a bolt nevét helyeztem el. Ez nyilván nem egy weblap, szándékosan egyszerűsítettem le a tartalmát. Gondoltam, nem szeretnél napokat tölteni az elkészítésével. Azonban a lényegét ezen is remekül meg fogod tanulni. A lényeg pedig az, hogy a <body> részbe írt kódokat is strukturáljuk.

```

<!DOCTYPE html>
<html lang="hu">
  <head>
    <meta charset="UTF-8">
    <title>Harsona Hangszerbolt</title>
  </head>
  <body>
    <header>
      <h1>Harsona</h1>
      Hangszerbolt
    </header>
    <nav>
      Címlap Hangszereink Webáruház Elérhetőségek
    </nav>
    <aside>
    </aside>
    <article>
      <h2>Hangszereink</h2>
      <section>
        <h3>Zongorák</h3>
        <p>
          Lorem ipsum dolor sit amet, consectetur adipiscing elit.
          Donec id lectus et felis fermentum ultricies. Ut finibus,
          finibus ultricies, mi orci gravida libero, sed auctor urna
          in augue. Donec venenatis non quam sed tempus. Nunc
          tempor nunc in nisi finibus, id cursus nulla vulputate.
          Maecenas id faucibus justo, a fringilla tortor. Nunc laoreet
          tellus eu justo molestie sodales.
        </p>
      </section>
      <section>
        <h3>Szintetizátorok</h3>
        <p>
          Lorem ipsum dolor sit amet, consectetur adipiscing elit.
          Donec id lectus et felis fermentum ultricies. Ut finibus,
          finibus ultricies, mi orci gravida libero, sed auctor urna
          in augue. Donec venenatis non quam sed tempus. Nunc
          tempor nunc in nisi finibus, id cursus nulla vulputate.
          Maecenas id faucibus justo, a fringilla tortor. Nunc laoreet
          tellus eu justo molestie sodales.
        </p>
      </section>
    </article>
    <footer>
      © Harsona Hangszerbolt
    </footer>
  </body>
</html>

```

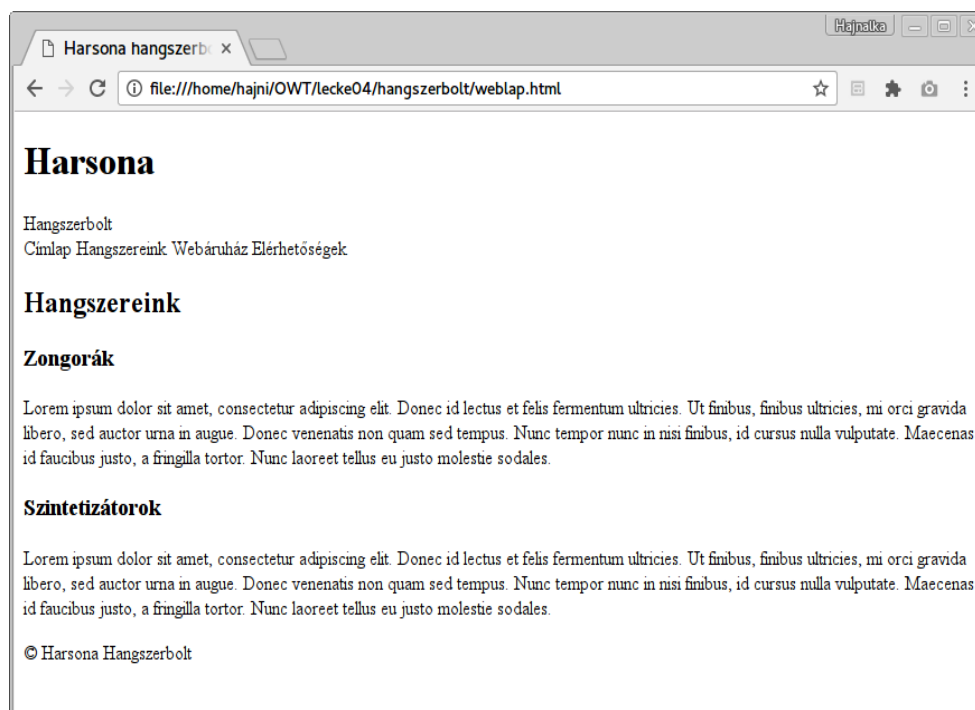

4. feladat

Nézd át alaposan a kódot. Elrejtettem egy hibát benne. Megtalálod?

5. feladat

<html> és </html> utasítások közé gépeld be az editorodba ezt a kódot, és mentsd el *hangszereink.html* néven! Ne nyugodj addig, amíg a kódotat el nem fogadta a validátor! A *lorem ipsum* részt természetesen nem kell begépelni: próbáld ki a *lipsum.com* weboldalt, az segít.

Ez az első igazán komplex feladatod. Bár az elvek egyszerűek, ha még nem vagy gyakorlott, néha igazán sok időt el lehet vacakolni egy-egy ilyen feladattal. Kérlek, légy türelmes: esetenként nagyon sok időbe is telhet, mire sikerül. Ha sok időbe telik, vagy nehézségeid támadnak, az csupán azt jelenti, hogy jó úton jársz, és szükséged van még erre a gyakorlásra.



Ezt a képernyőképet kapjuk a HTML kód begépelése után

Ha az imént az a botor gondolatod támadt, hogy túl hosszú ez a feladat, ezért inkább elpasszolnád, akkor szeretném jelezni, hogy erre a kódra később még szükségünk lesz. :-)

6. feladat

Most gyakoroljuk egy kicsit a stíluslap beállítást (ezt a 3. leckében már megtanultuk)! A 4. leckéhez kaptál egy CSS kódot *style.css* néven. Másold be ezt a fájlt a *css* nevű mappába, majd írd át a Harsona Hangszerbolt HTML kódjának fej részét, hogy mostantól ezt a stíluslapot jelenítse meg a böngészőprogramod.

7. feladat

Készíts saját színvilágot a weblaphoz: írd át az 4. feladatban használt *style.css* fájlban a színeket a saját színeidre. Ne ijedj meg, ha a kódban nem ismeresz minden jellemzőt, most direkt az ismeretlen kódban turkálást gyakoroljuk. Erre sokszor szükséged lehet majd a designer tevékenységed során.

3. Beadandó gyakorlati feladat

A lecke következő részében egy giga-mega nagy, és kifejezetten komplex feladatot fogunk megoldani. Többoldalas weblapot készítünk, melynek során lekódoljuk a hangszerbolt weblapjának alapjait. Ez már nem kétperces munka, kérlek, hogy készülj fel lélekben egy igazán nagy feladatra, és készíts a kezed ügyébe kétnapi hideg élelmet is! :-)

Elsőként a menü alapján tervezzük meg, hogy hogyan fognak kinézni az egyes oldalak:

```
<nav>
```

Címlap Hangszereink Webáruház Elérhetőségek

```
</nav>
```

- | | |
|---|-------------------|
| – a címlap, vagyis a főoldal: | index.html |
| – a termékinformációt tartalmazó aloldal: | hangszereink.html |
| – a webáruház aloldal: | webaruhaz.html |
| – és az elérhetőség aloldal: | elerhetoseg.html |

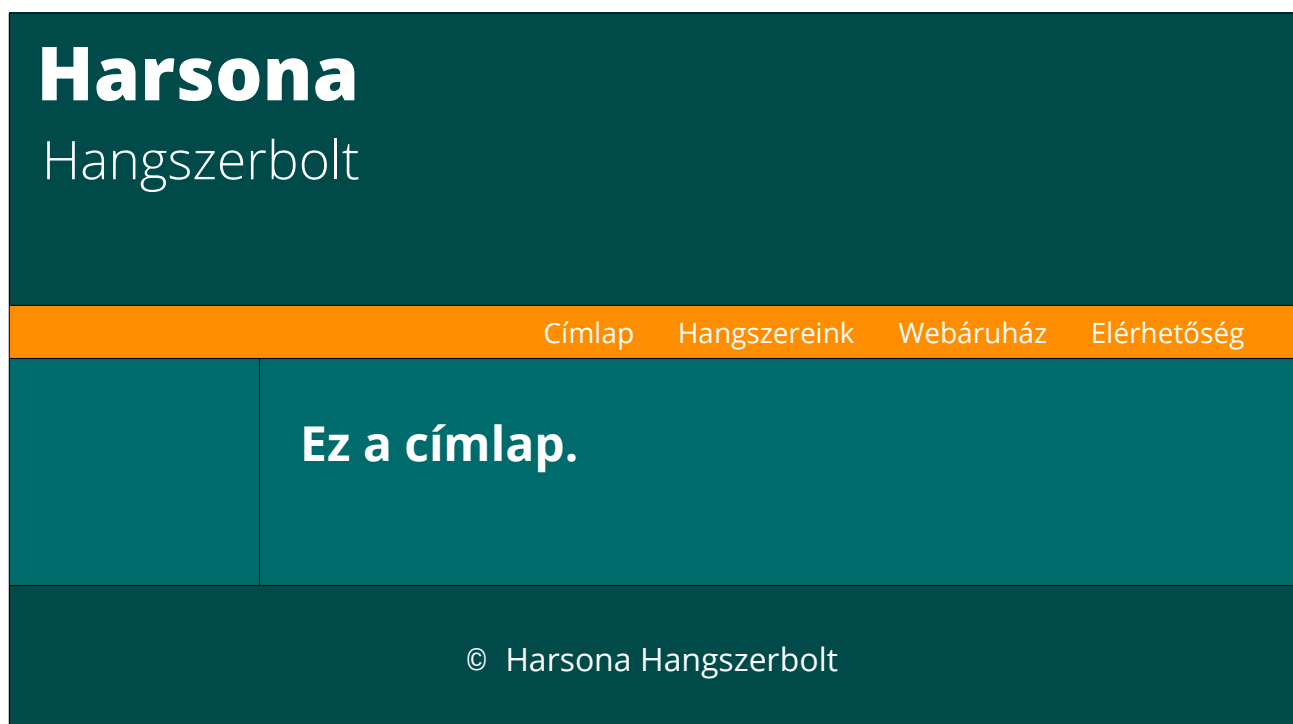
Ennyi lenne a feladatunk. Nézzük, hogyan oldjuk meg ezt a gyakorlatban! A hangszereink aloldalt már korábban elkészítettük (lásd 3. feladat: *hangszereink.html*), ezt felhasználhatjuk a többi oldal elkészítéséhez is. Elsőként elővesszük ezt a kódunkat, és létrehozuk a fenti négy HTML oldalt:

8. feladat: létrehozzuk a fenti négy HTML fájlunkat

Ha már lezártad, nyisd meg a kész *hangszereink.html* fájlot, és mentsd el először *index.html*, majd *webaruhaz.html* és legvégül *elérhetoseg.html* néven! Tehát négy pontosan ugyanolyan HTML fájlod lett, csak a nevük különbözik.

Nem valami menő weblap, amelynek minden oldala ugyanolyan, ezt nem tűrhetjük! Kérlek nyisd meg mindegyik oldaladat, és módosítsd a kódodat a következő oldalakon látható módon!

Az *index.html* legyen ilyen (töröld a kódból, ami nem kell):



Csak látszólag könnyű ez a feladat, hiszen csak ki kell törölnünk az ide nem illő részeket a kódból. Azonban nem könnyű egy kódból úgy törölni, hogy a végén kerek egész eredményt kapjunk.

Ha kódrészletet törölsz, mindig ügyelj arra, hogy ha törölsz egy címkét a benne lévő elemekkel együtt, akkor a címke lezárását is törölnöd kell:

~~<címke>~~

~~tartalom~~

~~</címke>~~

Harsona

Hangszerbolt

[Címlap](#) [Hangszereink](#) [Webáruház](#) [Elérhetőség](#)

Hangszereink

Zongorák

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec id lectus et felis fermentum ultricies. Ut finibus, velit eu finibus ultricies, mi orci gravida libero, sed auctor urna magna in augue. Donec venenatis non quam sed tempus. Nunc tempor nunc in nisi finibus, id cursus nulla vulputate. Maecenas id faucibus justo, a fringilla tortor. Nunc laoreet tellus eu justo molestie sodales.

Szintetizátorok

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec id lectus et felis fermentum ultricies. Ut finibus, velit eu finibus ultricies, mi orci gravida libero, sed auctor urna magna in augue. Donec venenatis non quam sed tempus. Nunc tempor nunc in nisi finibus, id cursus nulla vulputate. Maecenas id faucibus justo, a fringilla tortor. Nunc laoreet tellus eu justo molestie sodales.

A `webaruhaz.html` **fájlt** módosítsd a következőképpen:



Amíg dolgozol, halkan elmesélem, hogy ezekkel a feladatokkal kettős célunk van. Egyrészt gyakoroljuk a kódkezelést, másrészt komoly tervünk is van: hamarosan egy nagy weblappá fogjuk ezeket az oldalakat összefűzni.

Csak annyit változtatunk a kódon, hogy minimális munkával azért még látványos eredményt kapjunk. Ha te úgy érzed, hogy szeretnél komolyabb eredményt elérni, bátran fejleszd tovább a weblapodat. Beletehetsz képeket, további szövegeket, listákat, táblázatokat és ami még eszedbe jut. Azonban csak annyit vállalj, amennyit a kódolási gyakorlatod megenged, és minden módosítás után ellenőrizd, hogy az adott műveletet az elképzeléseid szerint jeleníti-e meg a böngészőprogram. Validálnod ezt a feladatot nem kell, de ha elakadtál, a validátor esetleg segíthet a hiba megkeresésében.

Arra is szeretnék megkérni, hogy mentsd a munkádat gyakran, egyre növekvő verziószámokkal, így ha egy ponton végleg sikerül összekutyulni a kódot, egyszerűen visszatérhetsz egy előző verzióhoz. A fájlnevében ne legyen szóköz és ékezet!

Egy részfeladatunk van még: az `elerhetoseg.html` fájlt az alábbiak szerint alakítjuk át (az e-mail cím legyen kattintható):



Ha végeztél mindezzel, kérlek, jutalmazd meg magad valami finomsággal, mert megérdemled. A csoportban kérlek, meséld is el nekünk, mert szeretnénk tudni, hogy mi volt az. :-) Rendben, már egészen jól állunk. Most következik a weblapok összeillesztése.

A menü

Elkészült a négy HTML fájlunk:

| | | | |
|-------------------------|--------------------------------|-----------------------------|-------------------------------|
| <code>index.html</code> | <code>hangszereink.html</code> | <code>webaruhaz.html</code> | <code>elerhetoseg.html</code> |
|-------------------------|--------------------------------|-----------------------------|-------------------------------|

Ezeket a fájlokat fogjuk linkek segítségével összefűzni. Nézzük is meg, hogyan néz ki a navigációs rész kódja!

A navigációs rész

```
<nav>
  Címlap Hangszereink Webáruház Elérhetőségek
</nav>
```

Így néz ki jelenleg a menü

Hogyan is lesz ebből igazi menü? Úgy, hogy linkeket teszünk a menü elemeire:

```
<nav>
  <a href="index.html">Címlap</a>
  <a href="hangszereink.html">Hangszereink</a>
  <a href="webaruhaz.html">Webáruház</a>
  <a href="elerhetoseg.html">Elérhetőségek</a>
</nav>
```

Ilyen lesz a menü, ha linkeket raksz a menüpontjaidra

Figyeld csak meg: minden menüpontunk megmaradt. Itt van a Címlap, a Hangszereink, a Webáruház és az Elérhetőségek menüpont is, csak most már kattintható formára hoztuk.

Ahhoz, hogy bármelyik oldalról el tudjuk érni a többit, ugyanezt el kell végeznünk a többi HTML kódunkban is. A fent jelzett eredeti `<nav>` és `</nav>` közötti kódrészletet tehát cseréld ki az alatta látható linkes formára mind a négy HTML fájlban! S tá-dááámmmm: lett egy négyoldalas weblapod!

9. Szorgalmi feladat

Bővítsd ki ezt a weblapot kedved szerint! Például dolgozd ki az egyes HTML oldalak `<article>` részét, tegyél ide képeket például, vagy színezd át kedvedre a CSS fájlot stb. Arra kérlek viszont, hogy tartsd meg a weblap eredeti felépítését, és ami nem megy, azt ne erőltess. Egyelőre nincs szükség elemek középre igazítására például.

4. Készítsünk űrlapokat!

Új téma következik, űrlapokat fogunk készíteni. Az űrlapot az előző feladat *elerhetosegek.html* fájlba fogjuk beilleszteni, de ha egyszerűbb megoldásra vágysz, nyugodtan nyithatsz neki egy teljesen új HTML fájlt is.

A form címke

Az űrlapok (angolul *form*) elemeit a `<form>` és `</form>` címke közé tesszük. És milyen elemek kerülhetnek egy űrlapra? Ezt fogjuk megnézni a továbbiakban. Az űrlap elemeinek meg kell adnunk a típusát (*type*), és adnunk kell neki egy nevet (*name*), amellyel majd később hivatkozni fogunk rá.

Szöveges beviteli mező: text

Vezetéknév:

Keresztnev:

Elsőként egy beviteli mezőt helyezünk el a kódban. Ennek a típusa: `text`. A beviteli mező segítségével adatokat kérhetünk be a weblap látogatójától. Az alábbi példában például megkérdezzük a látogató vezetéknév- és keresztnévét.

```
<form>
  Vezetéknév: <br>
  <input type="text" name="vezeteknev"><br>
  Keresztnev: <br>
  <input type="text" name="keresztnev"><br>
</form>
```

Bekérjük a látogató vezetéknévét és keresztnévét

A fenti *name* szónak semmi köze a vezetéknévhez és a keresztnévhez, csak véletlen egybeesés. A *name* jellemző segítségével csupán az adott beviteli mező nevét adhatjuk meg. Így ha később hivatkozni szeretnénk rá, akkor tudjuk, hogy melyik mezőről van szó. A *name* jellemző értéke ugyanúgy lehetne traktortípus vagy kutya fajta is.

Nyomógomb: submit

Elküldés

Nyomógomb segítségével küldjük el a bekért adatot a szervernek. Szintén meg kell adnunk a beviteli (input) elem típusát (`type="submit"`), és az értékét (`value="Elküldés"`). Amit a `value` értékben megadunk, az a felirat jelenik meg a nyomógombon.

```
<form>
  ...
  <input type="submit" value="Elküldés">
</form>
```

Elküldjük az űrlapot a szervernek feldolgozásra

Hosszabb szöveg: textarea

Néha arra is szükségünk lehet, hogy többsoros szöveget kérjünk a felhasználótól. Ilyen például tipikusan egy üzenet mező, ahol többsoros üzenetet is írhat nekünk a látogató. Az alábbi példában a nyomógomb fölött egy tízsoros és harmincoszlopos szöveges területet jelenítünk meg:

Ide az a szöveg kerül, amit
alapértelmezés szerint
szeretnénk megjeleníteni.

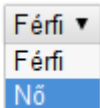
Elküldöm

```
<form>
  ...
  <textarea name="message" rows="10" cols="30">
    Ide az a szöveg kerül, amit alapértelmezés szerint szeretnénk megjeleníteni.
  </textarea>
  <input type="submit" value="Elküldöm">
</form>
```

Többsoros üzenet küldése

Legördülő választómező: select

A választómező segítségével egy legördülő menüből lehet a megfelelő értékeket kiválasztani. A value részhez azt a szót kell megadni, amellyel hivatkozni szeretnénk az adott opcióra. A tényleges szöveget, amely a legördülő listában is megjelenik, az `<option...> </option>` rész közé írjuk.

Az Ön neve: 

```
<form>
...
  Az Ön neve:
  <select name="valaszolo_neme"> ← A szóköz helyett mindig _ jelet írj!
    <option value="ferfi">Férfi</option>
    <option value="no">Nő</option>
  </select>
</form>
```

Legördülő listából választhatjuk ki a lehetőségeket

10. feladat

Egyelőre ennyi űrlap elem elég, most kérlek, készíts az `elەرhetoseg.html` fájlodba a hangszerbolt címe alá egy űrlapot, ami bekéri a felhasználó nevét, e-mail címét, és egy rövid üzenetet. A nyomógombról se feledkezz meg!

Miért nem működik az űrlap?

Mert nem közöltük vele, hogy mit csináljon. Nézzük, hogyan is működnek a weben az űrlapok. Először is, ott ül a felhasználó a weblap előtt. Beírja a név mezőbe a nevét, a e-mail mezőbe az e-mail címét, írja-írja az üzenetet, és ekkor...

No, igen: mi történik ekkor? Jó esetben a nyomógombot lenyomva a böngésző elküldi ezeket az adatokat egy szerveren futó programnak, ami feldolgozza ezt. Például elmenti a beírt adatokat, vagy elküldi a weblap tulajdonosának e-mailben, vagy statisztikákat készít ebből. Hogy éppen mit tesz vele, az attól függ, hogy hogyan írták meg a szerveren futó programot.

Hol tudjuk beállítani, hogy melyik az a program, amely a gomb megnyomásakor lefut? A `<form>` címke tulajdonságaként kell beállítani: `action="program_neve"`. Például, ha ez egy PHP nyelvű program, és az a neve, hogy `varazslat`, akkor így:

```
<form action="varazslat.php">
```

...

← Ide az űrlap kódját írod.

```
</form>
```

A PHP kód beállítása

PHP? Varázslat?

Nos, a PHP egy olyan programozási nyelv, amellyel szerveren futó programokat lehet írni. Azért van szükség arra, hogy a szerveren fusson a program, mert a szerveren futnak össze a látogatók gépeiről befutó adatok.

Ugye egyszerű esetben ülsz a gép előtt, netezel, futtatod a böngészőprogramodat. A böngésződ ilyenkor a te géped processzorán fut, és a te géped memóriájába tölti be az adatokat. Amint rákattintasz egy weboldalra, a te böngésződ lekér egy weblapot valami távoli webszerverről, ahonnan a weblap adatai megérkeznek ismét a te géped memóriájába. Ha még emlékszel, ezzel a magyarázattal kezdtük az 1. leckét.

A weblaphoz azonban nem csak a weblap konkrét kódja, vagy képek, videók tartozhatnak, hanem további olyan programok is, amelyekről te felhasználóként mit sem tudsz. Ezek a programok a szerver processzorán futva feldolgozzák a te gépedtől visszaérkező adatokat. Milyen adat lehet ez? Például a neved, e-mail címed, amit eltárol egy szerver oldali program a szerver merevlemezén, vagy más módon feldolgozza azt. Ezt a programot hívtuk a fenti példában `varazslat.php`-nek.

Szóval tanuljuk a HTML-t, tanuljuk a CSS-t, és akkor a PHP harmadik nyelv, amit meg kellene tanulni? Jó hírem van a számodra: igen! Sőt, lehetne még sorolni még további nyelveket is. A még jobb hírem viszont, hogy a szerver oldali programozás valójában egy teljesen külön szakma, és a PHP-vel is csak minimálisan kell majd foglalkoznod, amikor WordPress sablonokat fogsz készíteni.

Egyelőre elég annyit tudnod, hogy az űrlapodat egy lehet, hogy most még nem is létező program fogja feldolgozni valami általad ismeretlen szerveren.

11. feladat

Indítsd el az első leckében megismert html-online editort (<https://html-online.com/>). Az első leckében még inkább a bal oldalon, a szövegszerkesztő részen mozogtál otthonosan. Most viszont olyat teszünk, ami már túlhaladja ennek a szövegszerkesztőnek a képességeit: űrlapot hozunk benne létre. Nézd csak meg jobban: ez a szerkesztő tényleg nem képes erre alapból!

Rendben, akkor most fogj egy űrlappal kapcsolatos kódot, és másold az editor jobb oldalára a legördülő mező HTML kódját. A `<form>` és a `</form>` -ot ki ne hagyd! Mi történt? Mit szólsz? Első órán még a szerkesztő tanított minket, mostanra pedig túlhaladtuk a mester tudását: mi tanítjuk meg a szövegszerkesztőt valamire, amit alapból nem tud. Kérlek, folytasd, hozz létre egy teljes űrlapot. A témát te találd ki hozzá, a lényeg, hogy legyen füle-farka. Küldj az eredményről egy képernyőképet a csoportba, hogy mi is megnézhessük! (Szintén saját editorba dolgozz, és csak a végén másold a html-online editorba!)

12. szorgalmi feladat

Tervezz és kódolj le egy egyszerű weblapot, amely egy kérdőívet jelenít meg. A hátteret, betűszíneket, tehát a formázást külön CSS fájlban helyezd el! Lehet benne választómező, legördülő lista, bármi, ami egy kérdőívben előfordulhat. Az aljára helyezz el egy nyomógombot is „Küldés” felirattal. A gombnak nem kell működnie, csak legyen ott.

Ennek a HTML-kódját is szívesen megnézném.

Olvasmány

Most, hogy már túl vagyunk az első néhány leckén, talán megpróbálkoztál a neten új HTML címkéket vagy CSS jellemzőket keresgélni. Ez jó ötlet, hiszen ez alatt a néhány lecke alatt csupán az alap összefüggéseket tanuljuk meg, de nem tudjuk az összes lehetőséget számbavenni. A net minden információt tálcán kínál, érdemes kihasználni.

Azonban, ha körülnézel a neten található leírások között, sok nehézséggel és ellentmondással is szembe kell nézned. Ennek pedig az az oka, hogy többféle HTML és CSS verzió létezik. Ezért gyakran belefuthatsz olyan leírásokba is, amelyek jól működtek tíz évvel ezelőtt, és még most is működőképesek, de már elavult technikát mutatnak be. Miért is van mindez? Mert a böngészőprogramok mindent igyekeznek megjeleníteni, amit csak lehetséges. Így fordulhat elő, hogy ugyanúgy megbirkóznak az évtizedes technikákkal, mint a legfrissebbekkel.

Gyakran belefuthatsz például olyan leírásokba, amely régebbi Explorer böngészőkkel kapcsolatos tippeket-trükköket tartalmaznak. Ennek az az oka, hogy a régebbi Explorerek üzletpolitikai okokból fittyet hánytak a mások által használt szabványokra (a saját szabványaikat szerették volna elterjeszteni). A régebbi weblapokat külön Explorer trükkökkel kellett felvértetni, ami jelentősen megnövelte a weblapok fejlesztési költségeit, és egyben lassította a technológia fejlődést is. Ma már szerencsére az újabb Explorerek is törekednek a szabványos működésre, a régieket pedig már jóformán senki nem használja, tehát nem is biztos, hogy számodra fontos egy-egy Explorer trükk, amelyre a neten rábukkanhatsz.

Mire érdemes tehát figyelni, ha a netről szerzel be információkat? Például nézd meg, hogy az adott információ vagy tutorial mikor került fel a netre. A HTML legújabb verziója a HTML5 2014-ben kezdett elterjedni. A HTML5 előtti leírások, bár sokszor megoldják a feladatot, több ponton is elavultak lehetnek. A HTML5 leírásokat viszont bátran használhatod.

A CSS esetén például a CSS3-at és az utáni leírásokat érdemes felhasználni. Természetesen nem jelenti azt, hogy minden, ami ez előtti, csakis rossz lehet, de sok olyan dolgot egyszerűbben is meg lehet oldani, amit régen csak nyakatekert módon lehetett megvalósítani.

Vannak a neten olyan régebbi HTML leírások is, amelyek a HTML kódban formázást is tartalmaznak. Például színt, méretet stb. állítanak be. Ezeket a formázásokat rendben meg is jeleníti a böngészőprogramod, azonban rossz kódolási szokásokat vehetsz fel általuk. Nem érdemes használni ezeket.

Szóval hogyan is lehet eligazodni ebben az útvesztőben? Nagyon egyszerűen: kerüld a HTML kódban a formázó részeket, és minden formázást a CSS fájlban végezz. És mivel ezt nem lehet elégszer elmondani, és egy-egy érdekes tutorialt látva könnyű megfélekezni róla, ideírom nagy betűkkel is:

Minden formázást a CSS fájlban kell végezni!

Tényleg!

A „bűn”

És akkor most ünnepélyesen be kell vallanom egy „bűnömet” neked. Ha jobban belegondolsz, mi is tettünk a HTML kódba formázást, mégpedig, amikor az `` részen a kép szélességét, és magasságát adtuk meg. Ez is egyfajta formázás. A kép méretét formáztuk a segítségével, és remekül működött. Igaz?

Azért tanítottam így, hogy az első perctől tudj kulturáltan képet elhelyezni a weblapodon. De most már megint előre haladtunk kicsit az időben, egyre többet tudunk. Ezért ezt a két kódrészt kiszedjük az `` kódból, sőt, egy s mindenkorra el is felejtjük, mintha sosem láttuk volna:

```

```

És helyette a CSS kódban állítjuk be ugyanezt:

```
img {  
    width: 650px;  
    height: 330px;  
}
```

Ezzel tovább egyszerűsödik, és jobban átláthatóvá válik a HTML kód, és könnyebb lesz formázni is.

Mostantól tehát mondd nemet a formázásra: szedj ki minden egyéb formázást a HTML kódból! Mindent, amit csak alkalmaztál, a neten tanultál vagy magadtól tettél bele. Mostantól minden formázás, még a legapróbb is a CSS kódba kerüljön. Nincs width, nincs style, legyél a HTML kódoddal szigorú és kíméletlen! :-)

Szóval mostantól tényleg csak színtiszta tartalmat teszük a HTML kódba. (Ez alól egyelőre kivétel lehet egy-egy olyan kódrész, amelyet konkrét ajánlás alapján másolsz, például Google térkép vagy egyéb hasonló kódrészlet.)

Segítség, nem volt elég időm! Mit tegyek most?

Olvasd el a lecke végét! De tényleg! :-) Ezt a leckét direkt úgy építettem föl, hogy bármедdig is jutsz benne, fogod tudni folytatni a következő leckével, mert a 4. leckében főleg a HTML kóddal kapcsolatos ismereteket bővítettük, az 5. leckében pedig főleg a CSS kóddal fogunk foglalkozni. Nem épül egymásra a két anyag. Olvasd el az *Olvasmány* és A „bűn” részt, és haladj a leckében, ameddig csak időd engedi!