

Ebben a fejezetben átismétljük a világhálóval kapcsolatos alapfogalmakat, majd áttekintjük, hogy mi magunk hogyan készíthetünk és publikálhatunk weboldalakat.

## Az internet és a web kapcsolata

Előzetes tanulmányainkból már tudhatjuk, hogy az internet nem más, mint egy globális, az egész bolygónkat behálózó számítógépes hálózat.

Ez a hálózat valójában nem egy fizikai hálózatot jelent, hanem sokféle, önálló hálózatból, illetve hálózatfajtából áll, amelyek egymással összeköttetésben állnak. A hálózatra csatlakoztatott eszközök egy egyedi hálózati azonosítóval rendelkeznek, amelyet **IP-címnek** (Internet Protocol-címnek) nevezünk. Példa egy ilyen IP-címre: 84.206.104.74.



Az internethálózat sokféle információmegosztási, kommunikációs szolgáltatást támogat. Ilyen például az **e-mail** (elektronikus levél), a **chat** (csevegés) vagy a **fájltviteli szolgáltatás** (pl. FTP, SFTP, SCP), amely az állományok számítógépek közti átvitelét (feltöltését, letöltését) teszi lehetővé.

A **world wide web** (amelyet magyarul **világhálónak** nevezünk) szintén egy olyan szolgáltatás, amelyet az internethálózaton érhetünk el. Tehát a világháló és az internet nem ugyanazt jelentik, bár sokan (tévesen) szinonimaként használják a két fogalmat.

A world wide web megalkotása *Tim Berners-Lee*, valamint *Robert Cailliau* nevéhez fűződik, akik a CERN (Európai Nukleáris Kutatási Szervezet) munkatársai voltak. 1989-ben egy olyan rendszert kezdtek el kidolgozni, amely a világ különböző helyszínein dolgozó kutatók közti információmegosztást támogatta.

1993. április 30-án a CERN bejelentette, hogy a kidolgozott technológiát (www) mindenki szabadon, ingyenesen használhatja. Ebben az évben Magyarországon is elindult az első www-kiszolgáló, ami a `www.fsz.bme.hu` webcímen üzemelt.

1994-ben Tim Berners-Lee megalapította a *World Wide Web Konzorcium* nevű szervezetet (w3.org), amely mind a mai napig koordinálja a nyílt, webes szabványok kidolgozását.



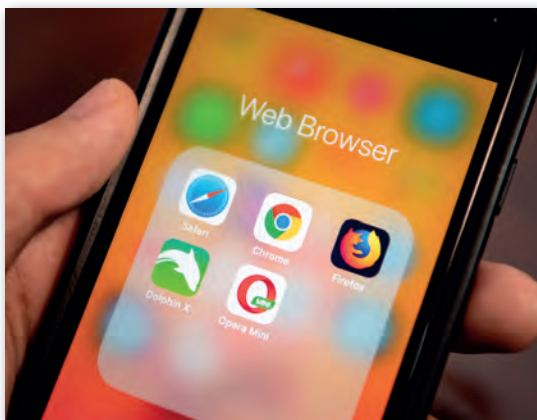
► Sir Timothy John Berners-Lee

## A www (világháló) építőkövei

A világháló egy olyan információs rendszer, amelyre jellemző, hogy **böngészőprogramok** segítségével különböző dokumentumokat (pl. weboldalakat, PDF-dokumentumokat stb.) és egyéb állományokat (pl. képeket, videókat, zenéket stb.) érhetünk el. Ezeket összefoglalóan **erőforrásoknak** (*resource*) nevezzük. Ezen erőforrások között a **hiperhivatkozások** (*linkek*) teremtenek kapcsolatot.

Így egy weboldaltól kiindulva a különböző hivatkozások követésével újabb és újabb oldalakra juthatunk el. Az viszont nem garantálható, hogy a visszafelé irányt is meg tudjuk tenni a hivatkozások segítségével, mivel a **hivatkozások egyirányúak**. A saját weboldalunkról bármilyen weboldalra hivatkozhatunk, de a hivatkozott oldalon ettől még nem biztos, hogy fognak hivatkozni a mi oldalunkra.

A www megalkotása során különböző kérdésekre kellett választ adniuk a tervezőknek. A következőkben ezekkel részletesebben foglalkozunk.



► Böngészőalkalmazások mobil platformon

### Milyen eszközzel érjük el a világhálón publikált anyagokat?

Tudjuk, hogy a világhálón publikált anyagokat jellemzően a **webböngésző programok** segítségével érhetjük el.

Böngészőprogramokat sokféle cég, szervezet fejleszt. A statisztikák szerint a legnépszerűbb webböngészők napjainkban a következők: *Google Chrome*, *Safari (Apple)*, *Mozilla Firefox*, *Samsung Internet*, *Microsoft Edge*, *Opera*.

Egy operációs rendszerre akár több, különböző böngészőprogramot is telepíthetünk.

### Feladatok, kérdések

Három-négy fős csoportokban vitassuk meg, hogy mely böngészőprogramokat kedveljük a legjobban, és miért? Állítsunk fel ez alapján népszerűségi rangsort! Ha használunk a felsoroltakon kívül más böngészőprogramokat, azoknak milyen előnyeik, egyedi funkcióik vannak? Minden csoport szóvivője ismertesse az eredményeket!

A böngészőprogram képes megjeleníteni a világhálón közzétett (publikált) tartalmakat, de ehhez meg kell adnunk azt a webcímet, ahol az adott tartalom elérhető. Folytassuk az ismerkedést ezzel!

## Hogyan hivatkozunk az elérendő dokumentumra?

A dokumentumok és más állományok elérési helyére az úgynevezett **URL** (*Uniform Resource Locator*), vagyis *Egységes Erőforrás Helymeghatározó* segítségével hivatkozhatunk. Az alábbiakban néhány példát látunk erre.

<a href="https://www.nava.hu/">https://www.nava.hu/</a>	A Nemzeti Audiovizuális Archívum weboldala.
<a href="https://www.mnm.hu/kiallitasok">https://www.mnm.hu/kiallitasok</a>	A Magyar Nemzeti Múzeum portál egy aloldala, amely a kiállításokat mutatja be.
<a href="http://www.fortepan.hu/_photo/display/154350.jpg">http://www.fortepan.hu/_photo/display/154350.jpg</a>	A Fortepan közösségi fotóarchívum weboldalán elérhető egyik kép webcíme.
<a href="https://www.w3.org/standards/">https://www.w3.org/standards/</a>	Webes szabványok a W3 konzorcium weboldalán.
<a href="https://hirmagazin.sulinet.hu/">https://hirmagazin.sulinet.hu/</a>	A Sulinet hírmagazin webportálja.
<a href="https://www.google.com/search?q=világháló">https://www.google.com/search?q=világháló</a>	A Google kereső találati oldala a világháló szóra keresve.
<a href="https://hu.wikipedia.org/wiki/Tim_Berners-Lee#Kitüntetései">https://hu.wikipedia.org/wiki/Tim_Berners-Lee#Kitüntetései</a>	A Tim Berners-Leeről szóló Wikipédia-oldal kitüntetésekkel foglalkozó oldalrészére mutató webcím.
<a href="file:///C:/honlapra/hobbi.html">file:///C:/honlapra/hobbi.html</a>	A saját számítógépünk c:\honlapra mappájában található <code>hobbi.html</code> nevű dokumentum.

Ahogy a példákban is látszik, az URL több részre tagolódik. Ezek a következők:

### 1. Séma

A séma utal arra, hogy milyen szabályrendszer (protokoll) szerint történik a kommunikáció a hálózat tagjai között. A protokoll leírja, hogy hogyan kell történnie a kapcsolatfelvételnek, hogyan történik az üzenetváltás, mi az üzenetek felépítése és az egyes üzenetekben milyen adatok lehetnek.



**Érdekesség:** A protokoll kifejezést hétköznapi értelemben is használjuk, a különböző viselkedési szabályokra, illetlani ismeretekre utalva.

A webes dokumentumok, állományok vonatkozásában leggyakrabban a `http` vagy `https` sémát szoktuk megadni. A séma neve után egy kettőspontot kell írni. A kettőspontot webes tartalmak esetén két perjel (`//`) követi.

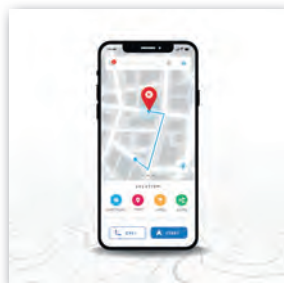
*Fontos! Amikor a böngészőprogramban megadunk egy webcímet, sokszor nem szoktuk begépelni az előtagot (pl. `http://`, `https://`), mégis megjelenik a weboldal. Ennek oka, hogy a böngészőprogramok ilyen esetekben alapértelmezetten a `http`-sémát használják.*

Amennyiben a saját számítógépünk egy mappájából nyitunk meg egy dokumentumot (pl. weblapot) a webböngészőben, akkor láthatjuk, hogy nem a `http` sémát, hanem a `file` sémát használja a böngészőprogram.

## 2. Útvonal

A séma után kell megadnunk az erőforrás elérési útvonalát. De ezt hogyan tudjuk megtenni?

*Ha egy épülethez szeretnénk eljutni, akkor használhatjuk a helyszín GPS-koordinátáit. Például a Parlament épületének a GPS-koordinátái: N47.50708 E19.04591. Ez a számsor jól használható egy navigációs szoftverben, de nehezen lehetne megjegyezni. Helyette így is megadhatnánk ugyanezt a címet: Budapest V. kerület, Kossuth tér 1–3. Ez utóbbi cím szintén azonosítja a helyet, és jól megjegyezhető.*



Láttuk korábban, hogy az internethálózatra csatlakoztatott eszközök egy egyedi IP-címmel rendelkeznek (pl. 84 . 206 . 104 . 74). A webböngésző programban akár ezen IP-cím segítségével is elérhetnénk a gépek szolgáltatásait. Ez azonban túlságosan bonyolult lenne, hiszen éppúgy egy hosszú, számokból álló azonosítót kellene használnunk, mint az előbbi példában a GPS-koordinátákat. Másrészt ezek az IP-címek változhatnak. Elképzelhető, hogy egy eszköz minden csatlakozáskor más-más IP-címet kap.

Ezért fontos, hogy a webes kiszolgálókra ne csak IP-cím alapján, hanem egy könnyebben megjegyezhető, állandó névvel is hivatkozassunk. Ilyen a korábbi példánkban a város, utca, házszám. Ez az azonosító az **állomásnév**, vagy más néven **hosztnév** (hostname). A teljes elérési útvonalnak azonban nemcsak az állomásnév a része, hanem az úgynevezett doménnev is.

A **doménnev** (domain name) egy olyan egyedi azonosító, amely kettő vagy több részből áll, és ezeket pontok választják el. A doménnev végződése (a korábbi példákban a *hu*, *com* és *org*) a **legfelső szintű tartományt** jelenti. Ezek lehetnek országokra utaló, illetve kategóriákra utaló rövidítések. Például a *hu* kód Magyarországot jelöli, a *jp* Japánt, az *org* a szervezeteket, a *com* az üzleti weboldalakat, és így tovább.

A legfelső szintű tartomány előtt a **második szintű tartományt** találjuk. Például a *sulinet.hu* tartománynévben a *sulinet* a második szintű tartománynév.

Második szintű tartománynevet meghatározott díjért bárki lefoglalhat saját célra (pl. vállalkozása, alapítványa, családja számára), amennyiben az adott név még szabad.

**Tipp!** Azt, hogy egy doménnev lefoglalható-e még, az Internet Szolgáltatók Tanácsa weboldalán (<http://www.domain.hu/domain/szabad-e/>) lehet ellenőrizni.

### Feladatok, kérdések

Nézzük meg, hogy a saját családnevünkkel megegyező doménnev lefoglalható-e (pl. Balaton Éva esetén azt kell ellenőrizni, hogy a *balaton.hu* doménnev szabad-e).

Mit tapasztalunk? Az osztály mekkora hányada tudná regisztrálni a saját nevét?

A második szintű tartomány tulajdonosa saját hatáskörben akár további aldoménneveket (harmadik, negyedik szintű) is kioszthat, és **helyi állomásneveket** (**hosztnév**) is meghatározhat. Az állomásnév egyben doménnev is lehet. Például a *hirmagazin.sulinet.hu* cím esetén a doménnev a helyi állomásnévből (*hirmagazin*) és a *sulinet.hu* szülődoménnévből áll össze.

## Hivatkozás mappákra és a bennük található erőforrásokra

A webes kiszolgálókon (szervereken) az információk mappaszerűen vannak szervezve, kicsit hasonlóan ahhoz, mint a saját számítógépünkön is. Az elérési út azt írja le, hogy milyen útvonalon érhető el az adott erőforrás.

*Például a [http://www.fortepan.hu/\\_photo/display/154350.jpg](http://www.fortepan.hu/_photo/display/154350.jpg) webcím esetén a [www.fortepan.hu](http://www.fortepan.hu) tartományhoz tartozó szerver `_photo` mappájában van egy `display` mappa, amelyben a `154350.jpg` nevű képre hivatkozunk.*

Ha nem adunk meg útvonalat, vagy csak egy mappa nevét adjuk meg, akkor egy alapértelmezett tartalom fog megjelenni a kiszolgáló szerver beállításai alapján (pl. `index.html`, `index.php`, `default.aspx`)

## Feladatok, kérdések

1. Szerveződjünk három-négy fős csoportokba! Egy böngészőprogramban nyissuk meg a saját iskolánk weboldalát! A böngészés során próbáljuk kideríteni, hogy vajon melyik aloldalnak van a leghosszabb webcíme! Osszuk fel egymás között, hogy ki melyik részét térképezi fel a weboldalnak, például a menüpontok alapján. Jegyezzük fel a leghosszabb webcímet, és értelmezzük, hogy az elérési útvonalban mi mit jelenthet!
2. Vannak-e az iskolának harmadik szintű tartománynevei? Ha igen, akkor melyek ezek? Gyűjtünk össze párat!

## 3. Lekérdezési paraméterek

A webcímekben akár paraméterek is lehetnek, amelyek alapján a weboldal akár más-más tartalmat tud szolgáltatni. A paramétereket *kulcs = érték* formában lehet megadni, az első kulcs elé pedig kérdőjelet kell tenni.

*Például a <https://www.google.com/search?q=világháló> webcímében paraméterként egy *q* nevű kulcs van megadva, amelynek értéke a világháló szöveg. A *q* az angol *query* (lekérdezés) szó rövidítése. Ezt a kulcsot és értéket egy program feldolgozza, és eredményként azon weboldalak listáját mutatja meg, amelyekben szerepel a „világháló” szöveg.*

Ha több ilyen kulcs lenne, akkor azokat & jellel kellene elválasztani.

*Például a <https://www.google.com/search?q=világháló&num=3> webcím már két kulcsot is tartalmaz. A második a *num*, ami az angol *number* (szám) szó rövidítése. Ez a megjelenítendő találatok számát jelenti.*

## Feladatok

Próbáljuk ki a Google keresőjében azt, hogy közvetlenül a webcímekben módosítjuk a paraméterek értékét! Keressünk rá ezzel a módszerrel a „Duna” kifejezésre!

## 4. Oldalrész

A webcímek végén egy olyan azonosító is állhat a # jel után, amely egy oldalrészre hivatkozik. Ilyenkor a böngészőprogram az oldal adott elnevezésű részéhez fog ugrani.

**Fontos!** Csak akkor tudunk egy weboldal adott részére ugrani, ha az oldal készítője az adott oldalrészét ellátta egy egyedi névvel. Ezalól kivételt jelent az oldal teteje, amelyre név nélkül, csak a # jel megadásával hivatkozhatunk.

Például a [https://hu.wikipedia.org/wiki/Tim\\_Berners-Lee#Kitüntetései](https://hu.wikipedia.org/wiki/Tim_Berners-Lee#Kitüntetései) webcím a Tim Berners-Leeről szóló Wikipédia-oldal azon részére ugrik, ahol a kapott kitüntetéseiről van szó.

## Feladatok

A Wikipédia oldalán keressünk olyan szócikket, ami egy olyan témával kapcsolatos, amelyet mostanában tanultunk, és rendelkezik oldalrész-hivatkozással. Másoljuk ki az oldalrész-hivatkozást tartalmazó linket, és mentjük el egy állományba!

## Az URN és URI fogalma

**Érdekesség.** Nemcsak az elérési útvonal, hanem speciális esetben egy **név (azonosító) alapján is hivatkozhatunk** egyes objektumokra. Ezt az URN (*Uniform Resource Name*), vagyis *Egységes Erőforrás Név* segítségével tudjuk megtenni. Ebben az esetben viszont alapvető követelmény, hogy a név világviszonylatban (térben és időben) egyedi legyen, vagyis valóban egyértelműen azonosítson egy elemet.

A saját nevünk például nem lehet URN, mert nem garantálható, hogy egy másik embert (akár a világ másik részén) ne hívjanak ugyanígy, most, vagy a jövőben. De az is lehet, hogy a múltban is élt már ilyen névvel valaki.

De akkor mi lehet URN? Bizonyára találkoztál már az ISBN számmal a könyvek borítóin. Ez az iktatószám világszinten egyedi. Magyarországon ezen azonosítókat a kiadók kérésére az Országos Széchényi Könyvtáron belül működő Magyar ISBN és ISMN Iroda osztja ki. Például Magyarország Alaptörvényére ezzel az URN azonosítóval hivatkozhatunk: **urn:isbn:9786155269813**

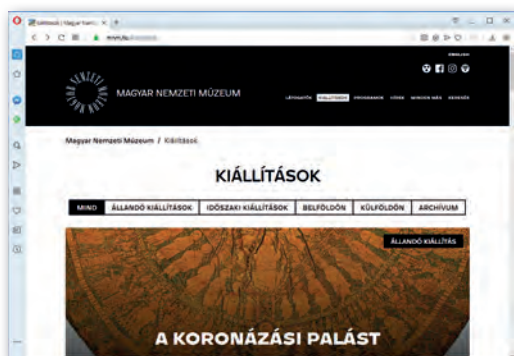
Mind a korábban látott URL-ek, mind az URN-ek beletartoznak egy **nagyobb kategóriába**, amely az **URI** (*Uniform Resource Identifier*) névre hallgat. Ezt magyarul *Egységes Erőforrás Azonosító*nak hívjuk. Vagyis minden URL egyben URI is, de nem minden URI-ra igaz, hogy URL.



## Hogyan történik a kommunikáció?

A világhálón jellemzően a http (*hypertext transfer protocol*, magyarul *hiperszöveg-átviteli protokoll*) szerint történik a kommunikáció, illetve biztonsági okokból gyakran a https-sémát használjuk a webböngészőkben, ami azt jelenti, hogy titkosított, biztonságos formában történik a kommunikáció.

A kommunikáció kérés-válasz formájában zajlik a kliens és a szerver között. A kliens (ami tipikusan a webböngésző program, de más alkalmazás is lehet) kérést küld a kiszolgálónak, vagyis a szervernek. Erre a kérésre válaszol a webszerver.



► Weboldal megjelenése egy böngészőben



A kommunikáció során a webböngésző megkapja az adott weboldal forráskódját is. A weboldal tartalmazhat képeket, videókat és egyéb elemeket is, amelyek szükségesek a megjelenítéshez. Ezeket a böngészőprogram szintén le kell, hogy kérje a szervertől újabb kérés-válasz ciklusokban, majd a kapott adatok alapján megjeleníti az oldalt.

## Hogyan, milyen nyelven írjuk le a dokumentumok tartalmát, illetve kinézetét?

### A tartalom leírása (HTML)

A világhálón jellemzően a HTML (*HyperText Markup Language*) jelelőnyelvet használjuk arra, hogy a weboldalak tartalmát leírjuk. Ennek a nyelvnek több verziója is van. A legutolsó és egyben legkorábbi változat a HTML5. Ezzel fogunk mi is foglalkozni.

**Fontos!** A HTML nem programozási nyelv, hanem a webes dokumentumok tartalmának leírására szolgáló leírónyelv.

A HTML nyelvben úgynevezett tagek (kiejtve tegek), vagy magyarul címkék segítségével tudjuk leírni a dokumentum tartalmát. A címkék „<” jellel kezdődnek és „>” jellel végződnek. Például egy bekezdés kezdetét a `<p>` taggel jelöljük. A `p` az angol *paragraph*, vagyis *paragrafus* szó rövidítése.

Egy címke hatása addig tart, amíg meg nem adjuk a záró párját. A záró címke hasonló, mint a kezdő címke, csak egy „/” karakter van a címke neve előtt. Vagyis egy bekezdést így jelölhetünk a HTML nyelv segítségével:

```
<p>Ez egy bekezdés</p>
```

Vannak olyan címkék is, amelyeknek nincsen záró párjuk. Ilyen például a `<br>` tag is, amellyel az utána következő szöveget új sorba törhetjük.

```
<p>Ez egy bekezdés, <br> amiben sortörést alkalmaztunk.</p>
```

A címkék használata során akár paramétereket is megadhatunk, sőt bizonyos esetekben ez kötelező is. A paramétereket a címke nyitó részében kell megadnunk `paraméter="érték"` alakban. Egy címkéhez több paramétert is írhatunk, ekkor szóközzel kell elválasztanunk őket. Vannak úgynevezett globális paraméterek. Ezeket bármilyen tag esetén megadhatjuk. Ilyen például a `title`, `class`, `id`. Más paramétereket csak az adott tag esetén használhatunk.

Például egy kép beillesztéséhez az `<img>` címkét kell használnunk. Az `img` az angol *image* (kép) szó rövidítése. Viszont ebben az esetben kötelezően meg kell adnunk azt is, hogy hol érhető el a kép (mi az URL-je). Ez kerül az `src` paraméterbe. Az `src` a *source* (forrás) szó rövidítése. Sőt, azt is le kell írunk szövegesen, hogy mi látható a képen. Ez az `alt` paraméterbe kerül. Az `alt` az *alternate* (helyettesítő leírás) rövidítése.

```

```

Az előbbi példában a `goldi.jpg` képet illesztettük be az oldalra. Mivel nincs megadva semmilyen hosszabb útvonal, csak a fájl neve, ez azt jelenti, hogy a kép ugyanabban a mappában van, mint maga a weblap, amelybe be van illesztve. Ha esetleg a képet nem tudná letölteni a böngészőprogram, akkor helyette az `alt` paraméterben megadott leíró szöveget jeleníti meg. A vak emberek szintén ezen szöveg alapján tudják eldönteni, hogy mi látható a képen.



## A kinézet leírása (Stíluslap)

A HTML5 nyelv segítségével a weboldalak tartalmát írhatjuk le, a kinézetét viszont nem. Erre a CSS (*Cascading Style Sheets*) szabvány szolgál, amelyet magyarul legtöbbször *lépcsőzetes stíluslapoknak*, *csatolt stíluslapoknak* vagy *egymásba ágyazott stíluslapoknak* fordítanak.

A szabványnak több verziója van, a jelenlegi legkorszerűbb változat a CSS3.



A stíluslapok szakszerű alkalmazásának számos előnye van:

- Számos formázást (pl. színbeállítások, szövegbeállítások) és oldalfelépítést (pl. két-oszlopos elrendezés) támogat.
- A weboldalak karbantartása egyszerűsödik, mivel az oldal kinézetének leírása nem keveredik a tartalom leírásával.
- Segíti az akadálymentes oldalak előállítását (később erre még kitérünk).
- Különböző eszközökre (pl. okostelefonokra, nyomtatókra, okosórákra) külön-külön stíluslap készíthető, amely biztosítja az optimális elrendezést és használhatóságot.

Nézzünk egy egyszerű példát! Korábban láttuk, hogy a bekezdést a `<p>` címkével tudjuk jelölni. Hogyan tudnánk minden bekezdés kinézetét megváltoztatni a stíluslap segítségével úgy, hogy a szöveg kék színű legyen, és nagyobb betűmérettel jelenjen meg?

Ahhoz, hogy **kijelöljük** a formázáshoz az oldal összes bekezdését, először meg kell adnunk a tag (címke) nevét. Ezért ezt a részt **kijelölőnek** vagy **szelektornak** hívjuk.

```
p {  
    color: blue;  
    font-size: 120%;  
}
```

A szelektor után a { és } jelek közötti **deklarációs blokkban** megadhatjuk, hogy hogyan szeretnénk megjeleníteni az adott elemet. Itt meg kell adnunk a tulajdonságokat, illetve az ahhoz tartozó értékeket.

A tulajdonság-érték megadást **deklarációnak** nevezzük. A különböző deklarációkat pontosvesszővel kell elválasztani egymástól.

Például a `color` tulajdonság a szöveg színét állítja be. Értékként megadhatunk különböző színneveket angolul.

A `font-size` tulajdonság a betű méretét jelöli. A 120% érték azt jelenti, hogy az alapértelmezett méret 120%-át szeretnénk beállítani, vagyis kis mértékben meg szeretnénk növelni a betűméretet.

## CSS osztályok használata

Persze egyáltalán nem biztos, hogy minden egyes bekezdést ugyanúgy szeretnénk formázni. Ebben az esetben használhatunk **osztályokat**. A HTML-kódban az egyes elemeket osztályokba sorolhatjuk. Ehhez a `class` paramétert kell használnunk. Az osztály nevét mi találhatjuk ki. A névben számok, betűk, kötőjelek és aláhúzások lehetnek, de például szóköz és más speciális karakter nem!



Ha például ki akarunk emelni egy bekezdést a többi közül, akkor használhatjuk osztálynévként a `kiemelt` szót!

**Tipp!** Mindig próbáljunk meg olyan osztálynevet megadni, amely általánosan írja le a formázás jellegét. Például ne adjunk olyan osztálynevet, hogy *piros*, mert lehet, hogy később az elem kinézetét megváltoztatjuk lilára, és akkor ez az osztálynév félrevezető lesz.

A CSS-kód:

A HTML-kód

```
p.kiemelt {           <p class="kiemelt">Ez egy kiemelt bekezdés</p>
  color: blue;
  font-size:120%;    <p>Ez pedig egy normál bekezdés</p>
}
```

Látható, hogy a CSS-kódban pontosan ugyanazt a nevet kell használnunk, mint a HTML-kódban, úgy, hogy a bekezdést jelölő `p` után egy pontot teszünk, majd leírjuk az osztály nevét. Ebben az esetben a `kiemelt` nevű osztály csak a bekezdés esetén érvényesül.

Létrehozhatunk olyan általános osztályokat is, amelyeket akár minden elemnél felhasználhatunk. Ekkor a pont elé ne írjunk semmit!

A CSS-kód:

A HTML-kód

```
.szegely {           <p class="szegely">Szegélyezett
  border:1px solid blue; bekezdés</p>
}                   <h1 class="szegely">Szegélyezett
                   címsor</h1>
```

A fenti példában a `szegely` osztályban azt adtuk meg, hogy az elem körül legyen egy szegély (*border*), ami 1 képpont vastag (*1px*), folytonos vonallal van megrajzolva (*solid*), kék (*blue*) színnel. Mivel ez egy általános osztály, több tag esetén is érvényre jut a formázás. Használhatjuk bekezdésnél (`<p>`), címsornál (`<h1>`), és még sok más elemnél is.

## Reszponzív weboldalak



A weboldalakat napjainkban igen sokféle eszközön tekintjük meg. Használhatunk asztali számítógépeket extranagy felbontású monitorokkal, notebookokat nagy felbontású kijelzőkkel, közepes kijelzőjű tableteket és kis kijelzős okostelefonokat, az egészen apró kijelzőjű okosórákról nem is beszélve!

Bármelyik eszközzel is nézzük meg a weboldalt, nagyon fontos, hogy az oldal az adott kijelzőn optimális módon jelenjen meg, vagyis egyszerűen lehessen navigálni az oldalak között, jól olvasható legyen a tartalom, ne kelljen átméretezni, illetve indokolatlanul sokszor görgetni a tartalmat. Ha egy weboldal teljesíti ezeket a követelményeket, akkor **reszponzív kialakítású weboldalnak** nevezzük.

Amikor magunk készítünk weboldalakat, akkor is törekednünk kell arra, hogy az oldal rezponzív legyen. Később látjuk majd, hogy akár különböző sablonokból is kiindulhatunk a weblapok készítése során. Itt mindig próbáljunk olyan sablont, illetve stílust kiválasztani, amely rezponzív megjelenést tesz lehetővé!

Azt, hogy egy weboldal rezponzív-e, úgy tesztelhetjük le, hogy többféle eszközön is megtekintjük az oldalt, illetve például asztali számítógépen a böngészőablakot átméretezzük (lecsökkentjük, felnagyítjuk).

### Feladatok, kérdések

Keressünk a saját hobbinkhoz kapcsolódó weboldalt, és vizsgáljuk meg, hogy az rezponzív módon lett megvalósítva, vagy sem. Használjunk a teszteléshez okostelefont, notebookot és/vagy asztali számítógépet!

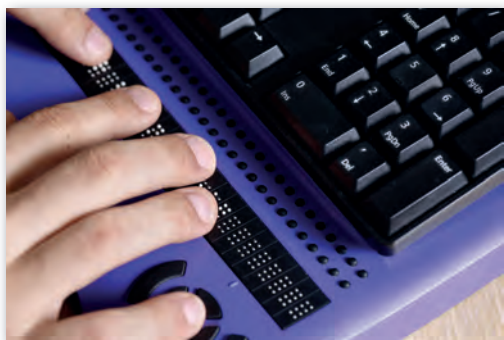
Amikor mindenki elkészült, osszesítsük az eredményeket! A vizsgált honlapok közül mekkora hányad volt rezponzív megvalósítású?

## Weboldalak akadálymentessége

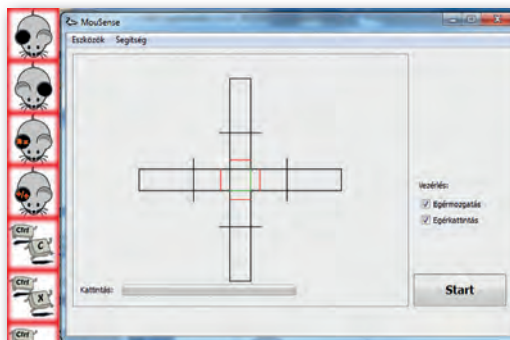
Nagyon sok, fogyatékkal élő embertársunk is böngészik a világhálón, ezért rendkívül fontos, hogy a weboldalakat olyan formában készítsük el, hogy mindenki számára, **akadálymentes** módon használhatók legyenek. Ha figyelünk a honlapok akadálymentes megvalósításra, az minden felhasználó számára előnyökkel járhat.



A fogyatékkal élő felhasználók jelentős hányadára jellemző, hogy a számítógép kezeléséhez, illetve a böngészőprogramok használatához valamilyen segédeszközt használnak. Ez lehet például egy speciális billentyűzet, olyan szájba vehető vagy fejre erősíthető pálca, amellyel le lehet nyomni a billentyűket, lehet Braille-kijelző a vak emberek által használt pontírás megjelenítésére, vagy akár egy speciális segédprogram (pl. képernyőolvasó program, fejegér, képernyőnagyító program) is.



- ▶ Braille-írás (pontírás) megjelenítésére képes kijelző a vak felhasználók számára



- ▶ Fejegér-alkalmazás (Mousense), amely lehetővé teszi, hogy az egeret ne kézzel, hanem fejmozgással irányítsák a felhasználók

A **vak emberek** nagyon gyakran úgynevezett képernyőolvasó programot használnak, amely felolvassa számukra a böngészőprogramban megnyitott weboldalak tartalmát, mégpedig a weblap forráskódja alapján. Ebben az esetben például nagyon fontos, hogy a **képek és más médiaelemek (pl. animációk, videók)** tartalmát **szövegesen** is leírjuk. Használjunk egyértelmű, jól érthető, a tartalomhoz megfelelően kapcsolódó **címsorokat**. Szintén alapvető, hogy a tartalom leírásakor mindig a megfelelő elemeket használjuk. Egy címsor nem attól lesz címsor, hogy nagy, félkövér betűkkel jelenítjük meg, hanem hogy a megfelelő HTML-címkét használjuk a kódban. Fontos, hogy lehetőleg **ne állítsunk be háttérzérzést** az oldalon, mivel ez zavaró a vak emberek számára (illetve nem csak számukra). A linkek szövegét úgy kell megfogalmaznunk, hogy abból kiderüljön, hogy milyen információt találunk rajta. Például, ha a „*Kattints ide*” szöveget állítjuk be hivatkozásként, akkor ebből nem derül ki, hogy mi fog történni a linkre kattintáskor.

A **gyengénlátó** emberek számára biztosítani kell, hogy jól olvasható, kontrasztos legyen a weboldal, és akkor is jól jelenjen meg, ha a betűméretet megnövelik a böngészőprogramban. Az animációk elkészítésénél is ügyeljünk a kontrasztarányra, illetve adjunk lehetőséget az animáció megállítására, kimerevítésére.

A **siket emberek** számára nagyon fontos, hogy a témát szemléletesen mutassuk be. Amennyiben meg tudjuk oldani, a videókat feliratozzuk, vagy a videók felhasználásakor részesítsük előnyben azokat, amelyek felirattal vannak ellátva. Az akadálymentesség egy mélyebb szintjét valósítják meg azok a videók, ahol jelnyelvi tolmácsolás is látható. A vak, illetve siket felhasználók számára a videó teljes szövegű átiratának megléte is nagyon fontos lehet. Ez az állomány tartalmazza a videó minden fontos történéseinek (párbeszédek, cselekmények) szöveges leírását, hasonlóan, mintha egy forgatókönyvet írnánk.

Nagyon sok **színtévesztő**, illetve **színvak** felhasználó nehézségekbe ütközik akkor, ha bizonyos információkat csak színekkel különböztetünk meg egymástól. Az alábbiakban egy olyan példát láthatunk, amikor a színnel való megkülönböztetés nem elegendő. Ilyen esetben például használhatunk ikonokat is a helyes, illetve helytelen válaszok jelölésére.

#### Minek a rövidítése a HTML?

- HyperText Markup Language
- HyperText Meta-Language
- HiperText Meta-Language
- HiperText Markup Line

► Teszt, melyben zöld és piros színekkel van megkülönböztetve a helyes és helytelen válasz

#### Minek a rövidítése a HTML?

- HyperText Markup Language
- HyperText Meta-Language
- HiperText Meta-Language
- HiperText Markup Line

► Ugyanez a teszt egy olyan ember szemével, aki zöld-sárga-piros tartományba eső színeket nem tudja megkülönböztetni

A **mozgáskorlátozott** emberek egy része nehézségekbe ütközik, ha billentyűzetet kell használnia, vagy az egérrel finom mozdulatokat kell elvégeznie. Ezért elegendő időt kell hagyni az adatbevitelre, illetve ügyelni kell arra, hogy ne kelljen nagyon kicsi területre kattintaniuk a felhasználóknak. Figyelni kell arra, hogy a weboldal csak billentyűzettel (egér nélkül) is használható legyen, de ez például a vak felhasználók számára is rendkívül fontos.

### Feladatok, kérdések

1. A *funkify.org* webcímen találunk egy ingyenesen kipróbálható szimulátort, amelyet a Google Chrome böngészőbe lehet telepíteni kiterjesztésként. Ez a szimulátor különböző állapotokat (pl. gyengénlátás, színtévesztés, remegő kezek) tesz kipróbálhatóvá. Próbáljuk ki ezeket a funkciókat úgy, hogy közben a saját iskolánk weboldala van megnyitva a böngészőprogramban!
2. Akár átmenetileg is kerülhetnek a felhasználók olyan helyzetbe, mint ha fogyatékkal élnek. Például ha rossz a hangkártya a számítógépben, akkor nem fogjuk hallani a számítógép hangját, akárcsak a siket emberek. Szerveződjünk három-négy fős csoportokba, és vitassuk meg, hogy milyen hétköznapi helyzetekben fordulhat elő az, hogy átmenetileg hasonló problémába ütközünk, mint a fogyatékkal élő embertársaink!

## Készítsünk weblapot!

A weboldalak készítésének több módja is van. Kisebb, pár oldalból álló honlapokat elkészíthetünk úgy, hogy a HTML-állományok forrását egy kódszerkesztőben készítjük el, vagy használhatunk segédprogramokat, amelyek hasonlítanak ahhoz, mintha egy szövegszerkesztő programban dolgoznánk, de az eredményt képesek elmenteni HTML-formátumban. Amennyiben a kódot közvetlenül szerkesztjük, majd azt publikáljuk, akkor valójában **statikus oldalakat** készítettünk.

### Dinamikus honlapok készítése

Összetettebb weboldalakat, amelyek számos oldalból állnak, és esetenként többben is szerkesztik a tartalmukat, már nem hatékony statikusan előállítani. Érdemesebb egy **CMS** rendszert (*Content Management System*), vagyis *Tartalomkezelő rendszert* használni, amely az oldalakat valós időben, **dinamikusan** állítja elő. A dinamikusság azt jelenti, hogy a webszerveren futó alkalmazás állítja elő azt a HTML-kódot, amelyet majd a böngésző megjelenít. Így például megtehetjük azt is, hogy egy közösségi oldalon megjelenő bejegyzés automatikusan megjelenik a weboldalon is anélkül, hogy nekünk kellene a tartalmat módosítanunk.



### Statikus honlapok készítése

Későbbi tanulmányainkban több tartalomkezelő rendszer használatával is megismerkedünk. Azonban ezekben a rendszerekben is előnyös, ha a HTML-kódot közvetlenül tudjuk módosítani, illetve a kinézetet leíró stíluslapállományt testre tudjuk szabni. Ezért először ismerkedjünk meg a statikus oldalak elkészítésének módjával!

Ahhoz, hogy egy statikus honlapot elkészítsünk, egy nagyon egyszerű kódszerkesztő programot is használhatunk.

Előnyös, ha ez a program rendelkezik a következő tulajdonságokkal:

- Kijelzi a sorok sorszámát, mert így egyszerűbben meg tudjuk találni az adott sorszámu sort, ha például egy tanári bemutatót követünk.
- Képes színnel megkülönböztetni a tageket, a paramétereket és a paraméterek értékeit. Ezt szintaxiskiemelő funkciónak nevezzük.
- Amikor elkezdjük gépelni a tageket vagy paramétereket, akkor felkínálja, hogy milyen tagek/paraméterek kezdődnek az adott karakterekkel, így nem kell feltétlenül pontosan emlékeznünk az adott kifejezésre, hanem akár ki is választhatjuk a megjelenő listából.

## Kódszerkesztő alkalmazások



- Kódkiemelés és paraméterek felkínálása a Brackets szerkesztőprogramban



- Kódkiemelés és paraméterek felkínálása a Notepad++ szerkesztőprogramban

Ha kiválasztottuk a megfelelő programot, amelyben kódolni fogunk, akkor induljunk ki az alábbi HTML5-sablonból! Ebben fogjuk elhelyezni az oldal tartalmát, a különböző HTML-címkéket használva.

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

- HTML5-alapstruktúra

Az első sor tartalmazza a **dokumentum-típust**. Ez azt írja le, hogy melyik szabvány szerint írjuk le a tartalmat, Ez a konkrét sor most arra utal, hogy a HTML5-szabvány szerint készítettük el az oldalt. Ez a sor nagyon fontos. Ha hiányzik, akkor a böngészőprogram nem biztos, hogy helyesen fogja megjeleníteni az oldalt.

A következő sorban a `<html>` taget helyezzük el, amely jelzi a böngészőnek, hogy egy HTML-dokumentumról van szó. A `lang` paraméterben az oldal nyelvét kell megadnunk. Magyar nyelvű tartalom esetén a `hu` értéket kell beírni, angol nyelvű tartalom esetén az `en` értéket.

A HTML-tagen belül két fő egységre tagolódik a tartalom, a **fejre** (`<head>`) és a **törzsre** (`<body>`).

A fej részben (`<head>`) kell beállítani például a karakterkódolást. Itt érdemes az UTF-8 kódolást használni, hogy a magyar ékezetes karakterek megfelelően jelenjenek meg az oldalon. Ehhez viszont a szerkesztőprogramban is ugyanezt a karakterkódolást kell beállítanunk a mentés előtt.

A `<title>` tagben az oldal címét kell megadni, amely megfelelően utal a weblap tartalmára.

A `<body>` és `</body>` tagek közti rész a dokumentumtörzs. Ebben helyezhetjük el a lap tartalmát, a címsorokat, bekezdéseket, képeket, videókat stb.

**Fontos!** A `<body>` tagben elhelyezett szöveget normál esetben nem tudjuk formázni sortörésekkel, tabulátorokkal, szóközökkel, vagyis ezek hatása nem fog megjelenni a böngészőprogramban. Térközöket a megfelelő tagek használatával, illetve majd a stíluslapok használatával tudunk beállítani.



## Készítsünk közösen egy weblapot!

A folytatásban készítsünk egy statikus honlapot, amelynek kapcsán megismerjük a legfontosabb cím-kéket!

A honlap a golden retriever kutyafajtáról szól. A hozzávalóit a letölthető állományok között találod a *goldi* mappában. Ebben megtalálhatók a felhasználható szövegek, a képek (képek mappája) és a videók (videók mappája) is.

Az *index.html* állományban megtaláljuk a korábban bemutatott alapstruktúrát. Nyissuk meg ezt az állományt a kódszerkesztő alkalmazásban!

Módosítsuk az oldal címét (a `<title>` tag tartalmát) úgy, hogy a szöveg utaljon a honlap tartalmára. Pl. így:

```
<title>Golden Retrieverek – Kezdőlap</title>
```

Először a honlap tartalmi részét fogjuk elkészíteni, és csak utána térünk rá arra, hogyan nézzen ki az oldal. Ezért kezdetben még nem a képen látható módon jelenik meg az oldal. Látni fogjuk ugyan a címsorokat, bekezdéseket, a fotókat és a videót is, csak az alapértelmezett megjelenéssel, fehér háttérrel, fekete szöveggel.

### Címsorok használata (`<h1>`, `<h2>`...)

A szövegszerkesztés témakörben megtanultuk, hogy a dokumentumot a címsorok segítségével kell tagolnunk annak érdekében, hogy az olvasó könnyen eligazodjon a tartalomban. Ez a weboldalak esetén sincs másként.

Kezdjük azzal, hogy elhelyezünk egy egyes szintű címsort, mindjárt a `<body>` nyitótag után. Erre a `<h1>` tag szolgál.

Szintén adjuk meg most az oldal alcímsorait is! Ezek kerüljenek kettes címsorba a `<h2>` taggel! Összesen hat címsorszintet lehetne használni a HTML nyelvben, vagyis akár a `<h6>` címkét is használhatnánk.

```
<h1>A golden retriever bemutatása</h1>
<h2>Előnyös tulajdonságai</h2>
<h2>Goldi, a kutya</h2>
```

► A begépelendő kód

Mentsük el az állományt, és nyissuk meg a böngészőprogramban! Ezt megtehetjük úgy, hogy a fájlt ráhúzzuk a böngészőprogram ablakára, de azt is megfelelő, ha duplán rákattintunk az állományra. Utóbbi esetben az alapértelmezett böngészőprogramban jelenik meg az oldal. Ha mindent jól csináltunk, akkor megjelennek a címsorok a böngészőprogramban. A címsorokat a böngésző nagyobb, félkövér betűvel jeleníti meg.



► Az elkészítendő honlap kezdőlapjának képe

## Megjegyzés beírása a forráskódba (<!-- -->)

Ha szeretnénk megjegyzést tenni a forráskódba, akkor ezt az `<!--` és `-->` jelek közé kell tennünk. A megjegyzésbe írt szöveg nem jelenik meg a böngészőben. Ha átmenetileg ki szeretnénk venni a megjelenítésből egy kódrészletet, akkor a megjegyzés jelekkel körbekeríthetjük. Azt azonban fontos tudnunk, hogy ha valaki megnézi a böngészőprogramban a forráskódot, akkor megtalálja, hogy milyen szöveget, illetve kódot tettünk megjegyzésbe. A megjegyzésbe tett szöveget általában eltérő színnel (pl. zöld) jelzik a kódszerkesztő programok.

A `<body>` tag után tegyünk be egy megjegyzést a kódba, amelyben a saját nevünk szerepeljen!

```
<!-- Nagy Tamara -->
```

## Bekezdés és sortörés használata (<p>, <br>)

Helyezzük el az első bekezdést a címsor alá! Használjuk ehhez a `<p>` taget! Próbáljuk ki a `<br>` tag hatását a bekezdésen belül, amely egy sortörést hoz létre. Ezt a „származik.” szó után helyezzük el. Vigyázzunk, ennek a címkének nincsen záró párja!

Ellenőrizzük a címke hatását a böngészőprogramban úgy, hogy frissítjük az oldalt!

Próbáljuk ki a kódot, és ellenőrizzük az oldal megjelenését a böngészőprogramban!

```
<p>A golden retriever egy kedves, barátságos kutya fajta, mely Skóciából származik. <br>A fajta tudományos megnevezése: golden retriever – Canis lupus familiaris.</p>
```

## Egyszerű szövegformázások (<b>, <i>)

Ismerkedjünk meg két új címkével, amelyekkel alapvető szövegformázásokat végezhetünk el. A `<b>` tag a szöveg félkövér formázására alkalmas. Főként kulcsszavak kiemelésére használatos. Például egy bekezdésen belül azon szavakat érdemes így kiemelni, amelyek fontosak a megértés szempontjából. A honlap látogatója ezen szavakat végigpáaszta gyorsabban áttekintheti a szöveget.

Az `<i>` tag segítségével dőlt betűs formázást állíthatunk be. Akkor használjuk, ha egy szövegrész hangulatban, hangnemben eltér a szöveg többi részétől. Az élőlények latin nevének leírásakor is ezt a címkét kell használni.

Próbáljuk ki a kódot, és ellenőrizzük az oldal megjelenését a böngészőprogramban!

```
<p>A <b>golden retriever</b> egy kedves, barátságos kutya fajta, mely Skóciából származik. <br>A fajta tudományos megnevezése: golden retriever – <i>Canis lupus familiaris.</i></p>
```

Félkövér és dőlt formázás azonban nem csak a fenti címkék használatával valósítható meg. De vigyázzunk, a hasonló megjelenés mögött, más-más jelentés lehet, amely a böngészőprogramokban nem vehető észre. Ha viszont egy vak felhasználó képernyőolvasó programot használ, akkor máshogy olvashatja fel a program ezeket a szövegeket.



## Fontos, hangsúlyos szövegek jelölése (<strong>, <em>)

A <strong> címkével olyan szövegeket emelhetünk ki, amelyen nagyon fontosak, ezért erősen ki szeretnénk emelni. Ezek megjelenítése éppúgy félkövér formázással történik a böngészőben, mint a <b> tag esetén.

Az <em> címke a hangsúlyos kiemelésre szolgál. Az élő beszédben is sokszor nyomatósítunk egy-egy szót, amely a mondat jelentését is befolyásolhatja.

Például az alábbi mondatoknak kissé módosul a jelentése, ha más-más szavakat hangsúlyozunk. Tegnap Ábel hiányzott az iskolából. Tegnap Ábel hiányzott az iskolából.

Az ilyenfajta hangsúlyok jelzésére alkalmas tehát az <em> tag.

A példánkban szereplő bekezdésben jelöljük meg, hogy melyek a fontos, hangsúlyos elemek!

```
<p><strong>Fontos!</strong> Ha gazdi szeretnél lenni, nézz utána,
hogya <em>felelős</em> kutyatartással kapcsolatban milyen ajánlások
és kötelezettségek vannak!</p>
```

Próbáljuk ki a kódot, és ellenőrizzük az oldal megjelenését a böngészőprogramban!

## Felsoroláslista használata (<ul>, <li>)

A következő téma a kutyusok előnyös tulajdonságairól szól. Ezeket egy felsoroláslistában fogjuk elhelyezni.

Felsoroláslistát úgy készíthetünk, hogy egy <ul> címkét nyitunk meg, és abban helyezük el a listaelemeket, amelyek <li> tagekbe kerülnek. Az ul az unordered list (felsorolás lista), a li a list item (listaelem) szavak rövidítése.

Próbáljuk ki a kódot, és ellenőrizzük az oldal megjelenését a böngészőprogramban!

```
<ul>
  <li>Okos, könnyen képezhető</li>
  <li>Szelíd, családbarát</li>
  <li>Nagyon jó vadászkutya</li>
</ul>
```

## Sorszámozott lista használata (<ol>, <li>)

A sorszámozott listák nagyon hasonlóan hozhatóak létre, mint a felsoroláslisták, csak ebben az esetben az <ol> taget kell használni. Az ol az ordered list (sorszámozott lista) szavak rövidítése.

A sorszámozott lista általános megadási módja:

```
<ol>
  <li>listaelem</li>
  <li>listaelem</li>
  <li>listaelem</li>
</ol>
```

A példánkban most nincs ilyen lista, ezért folytassuk a képekkel!

## Képek beillesztése (<img>)

A honlapokon képeket is elhelyezhetünk az `<img>` tag használatával. A legfontosabb paraméterei:

- src** A kép elérési útvonalát tartalmazza.
- alt** Alternatív szöveg, vagyis a kép rövid tartalmi leírását tartalmazza.
- title** A képhez rendelt cím, amely a böngészőkben akkor jelenik meg, ha a kép fölé visszük az egeret.
- width** A kép szélessége képpontokban.
- height** A kép magassága képpontokban.

### A kép forrása (src)

A kép forrását az **src** paraméterben kell megadnunk. Ennek egyik módja az **abszolút útvonalmegadás** (a kép teljes webcímének beírásával), de ennek előfeltétele, hogy a kép már publikálva legyen a világhálón. Pl. <http://tiny.cc/8mqblz>

Ha a honlapunk alkönyvtárszerkezete alapján adjuk meg az elérési utat, akkor **relatív útvonalmegadást** alkalmazunk. Fontos, hogy annak a HTML-állománynak az elhelyezkedéséhez képest adjuk meg a kép elérési útját, amelyben hivatkozni szeretnénk rá. Lássunk néhány példát!

`src="csaladikep.jpg"`

Ebben az esetben a kép ugyanabban a mappában van, mint a HTML-állomány.

`src="kepek/csaladikep.jpg"`

A kép most a kepek nevű mappában van. Ez a mappa pedig abban a könyvtárban van, amelyben a HTML-állomány.

`src="../../kepek/csaladikep.jpg"`

A kép most a kepek nevű mappában van. Ez a mappa pedig egy szinttel feljebbi könyvtárból nyílik, ahhoz képest, ahol a HTML-állomány található.

**Fontos!** Az a könyvtár, amelyben az `index.html` állományt elhelyezzük, lesz a gyöker-könyvtára a honlapunknak. Ezen belül tetszőleges könyvtárstruktúrát kialakíthatunk, de ne hivatkozzunk az `index.html` állományban (és a vele egy szinten lévő HTML-állományokban) olyan könyvtárra, amely feljebbi szinteken lévő alkönyvtárakra mutat (pl. `../kepek`), mert ez a weblap publikálásánál problémákat okozhat!

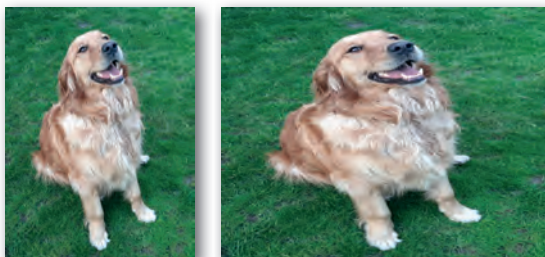
*Illesszünk be most két képet egymás mellé, azonos magasságban. A magasság legyen 200 képpont! Adjuk meg az alt és title paramétereket is!*

```


```

Próbáljuk ki a kódot, és ellenőrizzük az oldal megjelenését a böngészőprogramban!

**Fontos!** Amennyiben a szélesség (**width**) és a magasság (**height**) közül csak az egyiket adjuk meg, akkor a böngésző aránytartóan átméretezi a képet. Ha mindkét paramétert megadjuk, akkor vigyázzunk arra, hogy a kép arányai megváltozhatnak, ami a kép torzulásához vezethet.



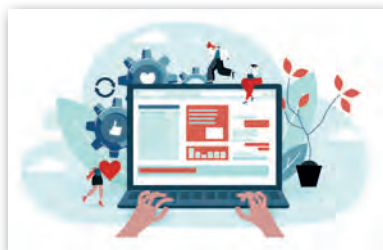
► Kép átméretezése aránytartó és torzított formában

### Ábrák, illusztrációk jelölése feliratokkal (<figure>, <figcaption>)

A HTML5-szabványban a <figure> címkét használhatjuk arra a célra, hogy illusztrációt adjunk meg. Ezen illusztrációhoz egy feliratot is társíthatunk a <figcaption> címkével, amelyet a <figure> tagen belül kell elhelyeznünk.

Ha az illusztrációnk egy (vagy több) kép, akkor természetesen az <img> taget (vagy tageket) kell elhelyeznünk a <figure> tagen belül.

A korábban már beillesztett képeket helyezzük el a <figure> tagen belül, és adjuk meg a kép feliratát is. Próbáljuk ki a kódot, és ellenőrizzük az oldal megjelenését a böngészőprogramban!



```
<figure>
  <img ...>
  <img ...>
  <figcaption>Fotók Goldiról</figcaption>
</figure>
```

**Fontos!** Az illusztrációk nemcsak fotók lehetnek. Grafikonokat, versrészleteket, programkód részleteket és más elemeket is elhelyezhetünk ebben a címkében.

### Hivatkozások (linkek) megadása (<a>)

A hivatkozások (linkek) segítségével el tudjuk érni, hogy az oldalunkról egy másik erőforrásra (weboldalra, médiaelemhez, egyéb fájlhoz) lehessen eljutni. Oldalon belüli ugrásra is használhatunk linkeket.

**Érdekesség:** Az angol link szó láncot jelent magyarul. A lánc pedig sokszor egy horgonyhoz kapcsolódik (pl. a hajók esetén). A horgonyt angolul anchornak nevezik. Ennek a rövidítéséből jött létre az <a> tag.



A link létrehozása a következő módon történik: `<a href="url">Link szövege</a>`  
Vagyis a `href` paraméterben kell megadnunk a hivatkozott erőforrás URL-jét. A címke nyitó és záró párja közti szöveg lesz a link szövege, amelyre majd rá tudunk kattintani.

*Készítsünk egy olyan linket, amelyet követve a honlapunk látogatói még több kutyás képet nézhetnek meg. A weboldal, amelyre hivatkozunk, a <http://tiny.cc/goldipix> rövidített címen elérhető Pixabay portál. A <https://pixabay.com/> portálon jogtiszta, ingyenesen felhasználható fotók találhatók. Csak rá kell keresned egy témára, és máris rengeteg jó minőségű fotót kapsz eredményül.*

Próbáljuk ki a kódot, és ellenőrizzük az oldal megjelenését a böngészőprogramban!

```
<p>Ha szeretnél még több kutyás fotót nézegetni, látogasd meg a  
<a href="http://tiny.cc/goldipix">Pixabay portál - Goldenek</a>  
fotó oldalát!</p>
```

### Hivatkozás oldalon belül

Azt is megtehetjük, hogy a hivatkozás segítségével **a lap egy adott pontjára ugrunk**. Ehhez azt a helyet el kell neveznünk az `id` paraméterrel. Ennek a névnek egyedinek kell lennie a teljes oldalon belül. Ezt a paramétert bármelyik címkénél felhasználhatjuk. Vigyázzunk, a név kis- és nagybetű-érzékeny.

Az egyedi nevet a link `href` paraméterében úgy kell megadnunk, hogy egy kettős keresztet teszünk elé (`href="#egyedinév"`).

Lássunk egy példát!

Van egy egyes címsorunk, amelynek a bevezető nevet adtuk.

```
<h1 id="bevezeto">Bevezető</h1>
```

Ha erre a pontra szeretnénk ugrani, akkor az `<a>` tag `href` paraméterében ugyanezt a nevet kell megadnunk, de elé egy kettős keresztet (hashtaget) kell írunk.

```
<a href="#bevezeto">Ugrás a bevezetőre</a>
```

Ha egy másik oldal (pl. `magamrol.html`) egy pontjára szeretnénk ugrani, akkor az oldal elérhetősége után kell megadnunk a # jelet és a nevet.

```
<a href="magamrol.html#bevezeto">Ugrás a bevezetőre</a>
```

*Helyezzünk el a weblapunk alján egy olyan linket, amellyel vissza tudunk ugrani az oldal elejére! Ehhez az egyes címsornak adjunk egy egyedi azonosítót, és a hivatkozásban ugyanezt a nevet adjuk meg!*

Próbáljuk ki a kódot, és ellenőrizzük az oldal megjelenését a böngészőprogramban!

```
<h1 id="eleje">A golden retriever bemutatása</h1>  
<p><a href="#eleje">Vissza az oldal tetejére</a></p>
```



## Hivatkozás saját oldalra, több oldalból álló weboldalak

Fejlesszük tovább úgy a weboldalt, hogy ne csak egy oldalból álljon. Mentsük el a korábban látott alapstruktúrát `tipusok.html` néven ugyanabba a mappába, mint ahol az `index.html` állomány van. Majd módosítsuk az `index.html` állományt, hogy a másik lapra is el lehessen jutni. Soroljuk be ezt a bekezdést a „menu” osztályba, hogy majd a stíluslap segítségével megváltoztathassuk a kinézetét!

```
<p class="menu"><a href="index.html">Kezdőlap</a> <a href="tipusok.html"> Típusok</a></p>
```

## Videó beillesztése (<video>, <source>)

A HTML5-szabvány jelentős újdonsága, hogy a multimédiás állományok beillesztését (pl. hangok, videók) teljesen leegyszerűsíti.

Videókat a `<video>` taggel tudunk az oldalba illeszteni. Hogy egy vezérlő eszköztár is megjelenjen (lejátszás, megállítás, hangerő stb. gombokkal), a `controls` paramétert is használnunk kell. Ez egy logikai paraméter, így elég leírunk a nevét, nem kell értéket adnunk.

A `<video>` tagen belül a `<source>` tag használatával adhatjuk meg a videó forrását. Ezen tag `src` paraméterében kell megadnunk a videó elérési útját. A legnagyobb böngészőtámogatottsága az mp4-formátumú videóknak van, így célszerű ilyen formátumban létrehozunk a videót. Az mp4 formátumú videók esetén a `video/mp4` értéket kell megadnunk a `type` (típus) paraméterben.

A `<video>` esetén is megadható a szélesség (`width`) és magasság (`height`) paraméter csakúgy, mint a képek esetén.

A videó beillesztésének általános módja tehát:

```
<video controls width="" height="">
  <source src="" type="video/mp4">
</video>
```

Próbáljuk ki a videóbeillesztést! A médiaelemek között találunk egy videót Goldiról. Hozunk létre egy bekezdést, amelyben bevezetjük szövegesen a videó tartalmát, majd ágyazzuk be az oldalra a videót!

Próbáljuk ki a kódot, és ellenőrizzük az oldal megjelenését a böngészőprogramban!

```
<p>A videón is láthatod, hogy Goldi nagyon szeret a földön hempergni, amely miatt általában koszosan jövünk haza a sétából.</p>
<video controls width="400">
<source src="videok/goldiseta.mp4"
  type="video/mp4">
</video>
```



## Hangok beillesztése (<audio>, <source>)

Hangokat nagyon hasonló módon lehet beilleszteni, mint a videókat. Ebben az esetben az <audio> taget kell használni, és érdemes mp3-formátumú állományokat használni a széles körű böngészőtámogatottság miatt.

A hang beillesztésének általános módja:

```
<audio controls>  
  <source src="" type="audio/mpeg">  
</audio>
```

**Figyelem!** Ha videókat, illetve hangokat illeszt be az oldalra, akkor abban az esetben lesz akadálymentes a weboldal, ha a videó vagy hangállomány tartalmát szövegesen is leírod. Ha a videó rövid, akkor magában a weboldal szövegében is lehet utalni arra, hogy mi látható a videón. Hosszú leírást érdemes külön weblapon elkészíteni, és az adott videó vagy hang alá elhelyezni egy hivatkozást, amely erre a leírásra mutat. Videó esetén egy forgatókönyvszerű leírást érdemes mellékelni, hang esetén pedig az elhangzó szöveget érdemes leírni.



## Táblázatok használata (<table>, <tr>, <th>, <td>)

Ha táblázatos formában szeretnénk elhelyezni adatokat a weboldalunkon, akkor több címkével is meg kell ismerkednünk.

A <table> tagben kell elhelyeznünk a teljes táblázatot. Ebben megadhatjuk a táblázat feliratát (<caption>) is.

A táblázat sorait a <tr> címke jelöli. A táblázat soraiban elhelyezhetünk fejléccellákat (<th>), amelyek leírják, hogy az adott sorban, illetve oszlopban milyen jellegű adatok vannak. Az adatokat tartalmazó cellákat pedig <td> tagek közé kell zárunk. Az egy sorban lévő elemeket balról jobbra haladva kell leírunk.

Nézzünk egy példát! Ha az alábbi táblázatot szeretnénk egy honlapon elhelyezni, amely az iskolai futóverseny eredményeit tartalmazza, akkor az itt látható kódot kell használnunk.



Az 5 km-es futóverseny eredményei

Helyezés	Név	Osztály
1.	Kiss Réka	9.C
2.	Hegedűs Ábel	9.A
3.	Horváth Kristóf	9.B

```

<table>
<caption>Az 5 km-es futóverseny eredményei</caption>
<tr>
    <th>Helyezés</th>
    <th>Név</th>
    <th>Osztály</th>
</tr>
<tr>
    <td>1.</td>
    <td>Kiss Réka</td>
    <td>9.C</td>
</tr>
<tr>
    <td>2.</td>
    <td>Hegedűs Ábel</td>
    <td>9.A</td>
</tr>
<tr>
    <td>3.</td>
    <td>Horváth Kristóf</td>
    <td>9.B</td>
</tr>
</table>

```

## Feladatok

Készítsük el önállóan az itt látható táblázatot, és helyezzük el a `tipusok.html` oldalon! Ne csak a táblázatot valósítsuk meg, hanem a címsort, a főmenüt és a forrásra mutató hivatkozást is!

A Golden Retriever bemutatása	
<a href="#">Kezdőlap</a> <a href="#">Típusok</a>	
<b>Típusok</b>	
A Golden Retriever típusai	
Típus	Típus leírása
Munkatípus	Könnyedebb felépítésű. Vékonyabb, hosszabb fej, sötétebb szőrszín, élénk temperamentum, karcsúbb, ruganyosabb, gyorsabb.
Kiállítási típus	Dúsabb, hosszabb szőrzet, robusztusabb felépítés, világosabb szőrszín, határozottabb.
Európai típus	Szőrszíne az arany világosabb árnyalata. Rövidebb szőrzet, mint az amerikai. Lába rövidebb az amerikaiénál, így megjelenése zömökebb. Jól pigmentált. Jó szögellésű elől, hátul.
Amerikai típus	Orr és fej része finomabb. Színe középarány, rézvörös, de sosem krém. Mellesz szögellésel szerényebbek, hátul olykor túlszögelt.
Forrás: <a href="http://hu.wikipedia.org/wiki/Golden_retriever">hu.wikipedia.org/wiki/Golden_retriever</a>	
<a href="#">Vissza az oldal tetejére</a>	

## A legfontosabb HTML5-címkék összefoglaló táblázata

Tag (címké)	Leírás
<code>&lt;p&gt;Bekezdés&lt;/p&gt;</code>	Bekezdés ( <i>paragraph</i> )
<code>&lt;h1&gt;Címsor 1&lt;/h1&gt;</code>	Egyes címsorszint ( <i>heading</i> )
<code>&lt;h6&gt;Címsor 6&lt;/h6&gt;</code>	Hatos címsorszint ( <i>heading</i> ) (a közbelső szintek értelemszerűen <code>h2</code> , <code>h3</code> , <code>h4</code> , <code>h5</code> )
<code>&lt;b&gt;szöveg&lt;/b&gt;</code>	Félkövér ( <i>bold</i> ) szöveg (kulcsszavak jelölése)
<code>&lt;i&gt;szöveg&lt;/i&gt;</code>	Dőlt ( <i>italic</i> ) kiemelés (eltérő hangulatú szöveg, latin nevek)
<code>&lt;strong&gt;fontos szöveg&lt;/strong&gt;</code>	Fontos, erősen kiemelt ( <i>strong</i> ) szöveg
<code>&lt;em&gt;hangsúlyos szöveg&lt;/em&gt;</code>	Hangsúlyosan kiemelt ( <i>emphasis</i> ) szöveg
<code>&lt;br&gt;</code>	Sortörés ( <i>break</i> )
<code>&lt;img src="uri" alt="leírás" width="" height=""&gt;</code>	A megadott forráson ( <i>source</i> ) elérhető kép ( <i>image</i> ) beszúrása leírással ( <i>alternate</i> ), megadott szélességben ( <i>width</i> ), magasságban ( <i>height</i> ).
<code>&lt;figure&gt; (ide illesztjük be a képet, stb.) &lt;figcaption&gt;Felirat &lt;/figcaption&gt; &lt;/figure&gt;</code>	Illusztráció ( <i>figure</i> ), felirattal ( <i>figure caption</i> )
<code>&lt;ul&gt; &lt;li&gt;listaelem&lt;/li&gt; &lt;/ul&gt;</code>	Felsoroláslista ( <i>unordered list</i> ), listaelemmel ( <i>list item</i> )
<code>&lt;ol&gt; &lt;li&gt;listaelem&lt;/li&gt; &lt;/ol&gt;</code>	Sorszámozott lista ( <i>ordered list</i> ), listaelemmel ( <i>list item</i> )
<code>&lt;a href="uri"&gt;Link szövege&lt;/a&gt;</code>	Hiperhivatkozás (link) ( <i>a=anchor</i> , <i>href=hyperlink reference</i> )
<code>&lt;video controls width="" height=""&gt; &lt;source src="mp4_videó_url" type="video/mp4"&gt; &lt;/video&gt;</code>	Videó (mp4) ( <i>video</i> ) beillesztése, vezérlő eszköztárral ( <i>controls</i> ), megadott szélességben ( <i>width</i> ), magasságban ( <i>height</i> ). A forrás elérhetőségét ( <i>source</i> ) és típusát ( <i>type</i> ) is meg kell adjuk.
<code>&lt;audio controls&gt; &lt;source src="mp3_hang_url" type="audio/mpeg"&gt; &lt;/audio&gt;</code>	Hangállomány (mp3) ( <i>audio</i> ) beillesztése, vezérlő eszköztárral ( <i>controls</i> ). A forrás elérhetőségét ( <i>source</i> ) és típusát ( <i>type</i> ) is meg kell adjuk.
<code>&lt;!-- megjegyzés --&gt;</code>	Megjegyzés ( <i>comment</i> ) elhelyezése a HTML-kódban
<code>&lt;table&gt; &lt;caption&gt;&lt;/caption&gt; &lt;tr&gt; &lt;th&gt;&lt;/th&gt; &lt;td&gt;&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt;</code>	Táblázat ( <i>table</i> ) elhelyezése: <ul style="list-style-type: none"> <li>• <code>table</code>: táblázat</li> <li>• <code>caption</code>: felirat</li> <li>• <code>tr</code>: táblázat sora (<i>table row</i>)</li> <li>• <code>th</code>: fejléccella (<i>table heading</i>)</li> <li>• <code>td</code>: adatcella (<i>table data</i>)</li> </ul>

## A stíluslapok (CSS) használata

Most már elkészült a honlapunk tartalma. De hogy kapja meg azt a kinézetet, amelyet a korábbi képen láthattunk? Csatoljunk hozzá egy megfelelő stíluslapot! A CSS mappában megtalálható az a stíluslap (goldi.css), amely a fotón látható kinézetet eredményezi. Csatoljuk hozzá a HTML-állományhoz.

A stíluslapok csatolása sokféle módon történhet a dokumentumokhoz. Most a külső stíluslap csatolását mutatjuk meg.

### A <link> tag használata

A külső stíluslapokat kétféle módon is hozzácsatolhatjuk a HTML-állományokhoz. Az első módszer, hogy használjuk a <link> taget a HTML-állomány <head> részében, az alábbi módon:

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

A href paraméterben kell megadnunk a külső stíluslapállomány elérhetőségét.

A mi esetünkben a CSS-állomány a CSS mappán belül helyezkedik el, goldi.css néven. Ezért a következő kódot kell használnunk az oldal <head> részében:

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

Próbáljuk ki a kódot, és ellenőrizzük az oldal megjelenését a böngészőprogramban! Ugye megváltozott az oldal kinézete a korábban látott képnek megfelelően?

### A @import szabály használata

Külső stíluslapot a CSS-szabvány segítségével is lehet csatolni a dokumentumhoz. Ahhoz, hogy ezt kipróbáljuk, tegyük megjegyzésjelek közé a <link> taget vagy töröljük ki!

Ezt a módszert követve, a <head> részben el kell helyezni egy lapon belüli stílusleírást a <style> taggel. Ennek az elején lehet használni a @import szabályt.

```
<style>
  @import url("");
</style>
```

A mi esetünkben a következő kódot kell használnunk:

```
<style>
@import url("css/goldi.css");
</style>
```

Próbáljuk ki a kódot, és mentjük el az eredményt! Frissítsük az oldalt a böngészőben! Ugye most is érvényre jut a csatolt stíluslap?

## A stíluslap módosítása

A korábbi fejezetben bemutattuk, hogy hogyan tudunk a CSS-szabvány segítségével kijelölni elemeket (szelektorok), és hogyan adhatjuk meg a tulajdonságokat a deklarációs blokkban. Most ismerkedjünk meg a témával a gyakorlatban is!

### Feladatok

Nyissuk meg a `goldi.css` állományt abban a szerkesztőprogramban, amelyet a HTML-állományok szerkesztésére is használtunk. Megnyitás után láthatjuk, hogy milyen CSS-szabályokat alkalmaztunk a stíluslapállományban. A jobb megértés érdekében minden sorhoz egy magyarázatot is elhelyeztünk.

Annak érdekében, hogy stíluslapot tudjunk készíteni, vagy meglévőt módosítani, az alábbi táblázatban elhelyeztük a leggyakrabban használt tulajdonságokat és magyarázatukat.

<code>text-align</code>	A szöveg vízszintes igazításának módja. Értékei: <code>left</code> (balra), <code>center</code> (középre), <code>right</code> (jobbra), <code>justify</code> (sorkizárt módon). Példák: <code>text-align:center;</code> <code>text-align:right;</code>
<code>font-size</code>	A betű mérete. Megadható például képpontokban (px), vagy akár százalékosan (%) is. Használható az <code>em</code> mértékegység is, amellyel az aktuális betűméret valahányszorosát állíthatjuk be. Példák: <code>font-size:16px;</code> <code>font-size:120%;</code> <code>font-size:1.5em;</code>
<code>font-weight</code>	A szöveg félkövér megjelenítése. Tipikus értéke a <code>'bold'</code> . Ha vissza akarjuk állítani a normál megjelenést, használjuk a <code>'normal'</code> értéket! Példák: <code>font-weight:bold;</code> <code>font-weight:normal;</code>
<code>font-style</code>	A szöveg dőlt megjelenítésére szolgál. Tipikus értéke az <code>'italic'</code> . Ha vissza akarjuk állítani a normál megjelenést, használjuk a <code>'normal'</code> értéket! Példák: <code>font-style:italic;</code> <code>font-style:normal;</code>
<code>font-family</code>	A betűtípus beállítására szolgál. Konkrét betűtípusokat fel lehet sorolni, a végén egy általános betűcsaládot kell megadni (pl. <code>serif</code> = talpas betűk, <code>sans-serif</code> = talp nélküli betűk). Ha a betűtípus neve szóközt tartalmaz, akkor aposztrófjelek közé kell tenni. Példák: <code>font-family:Arial,Verdana,sans-serif;</code> <code>font-family:'Times New Roman',Times,serif;</code>
<code>border</code> <code>border-top</code> <code>border-right</code> <code>border-bottom</code> <code>border-left</code>	Az elemet körbevevő szegély beállítására szolgál. Megadhatjuk a szegély vastagságát képpontokban (px), a szegély stílusát ( <code>solid</code> = folytonos, <code>dotted</code> = pontozott, <code>dashed</code> = szaggatott, <code>double</code> = dupla stb.), valamint a színét. Ha a <code>border</code> tulajdonságot használjuk, akkor mind a négy oldali szegély ugyanolyan lesz. Ha nem ezt akarjuk, akkor használhatjuk csak az adott oldalra vonatkozót ( <code>top</code> =felső, <code>right</code> =jobb, <code>bottom</code> =alsó, <code>left</code> =bal) Példák: <code>border-top:1px solid blue;</code> <code>border:4px double #8b0000;</code>
<code>border-radius</code>	A szegély lekerekítettségét állítja be. Például: <code>border-radius:20px;</code>
<code>float</code>	Az adott elemet lehet balra vagy jobbra lebegtetni. A lebegtetés azt jelenti, hogy a környező elemek körbefolyják az adott elemet. Például: <code>float:left;</code> <code>float:right;</code>



padding padding-top padding-right padding-bottom padding-left	Kitöltés megadása, ami a tartalom és az ezt körbevevő szegély közti térközt jelenti. Belső margónak is hívják. Ha a <code>padding</code> tulajdonságot használjuk, és egy értéket adunk meg, akkor mind a négy oldali beállítás ugyanakkora lesz. Ha nem ezt akarjuk, akkor használhatjuk csak az adott oldalra vonatkozót.
margin margin-top margin-right margin-bottom margin-left	A margó megadására szolgál. A margó az adott elem szegélye és az őt körülvevő többi elem közti távolságot jelenti. Ha a <code>margin</code> tulajdonságot használjuk és egy értéket adunk meg, akkor mind a négy oldali beállítás ugyanakkora lesz. Ha nem ezt akarjuk, akkor használhatjuk csak az adott oldalra vonatkozót. Az elemek középre igazítására is a margóbeállítást kell használni, ilyenkor <code>'auto'</code> értéket kell adni a bal és jobb margónak. Ekkor a <code>margin-left:auto;</code> és <code>margin-right:auto;</code> tulajdonságokat kell beállítani.
color	A szöveg színének beállítására szolgál. Megadhatunk színnevet (pl. blue), de akár RGB színkódot is. A képszerkesztő programokban gyakran a színkódot hexadecimális kóddal is megtaláljuk, amelynek az elején egy # karakter áll. Az alábbi példában ugyanazon szín megadását láthatjuk különböző módokon: <code>color:darkred;</code> <code>color: #8b0000;</code> <code>color: rgb(139,0,0);</code>
background-color	Háttérszín beállítása. A színek hasonlóan adhatóak meg, mint a szövegszín esetén.
width	Az elem szélességét állítja be. A méret megadható képpontokban és akár százalékban is. Példák: <code>width:600px;</code> <code>width:90%;</code>
height	Az elem magasságát állítja be. Hasonlóan állítható be, mint a szélesség.
min-width min-height	Az elem minimális szélességét, illetve magasságát jelenti, amely alá nem lehet átméretezni a böngészőprogramban.
max-width max-height	Az elem maximális szélességét, illetve magasságát jelenti, amely fölé nem méretezi át a böngésző az elemet.
background-image	A háttérkép beállítására szolgál. A háttérkép elérhetőségét az <code>url()</code> szövegen belül kell megadni, az aposztrófok között. Például: <code>background-image:url('kepek/mancs.png');</code>
background-repeat	A háttérkép ismétlődésének beállítására alkalmas. Az értékei a következők lehetnek: <code>repeat</code> (mozaikszerűen ismétlődő), <code>no-repeat</code> (nem ismétlődő), <code>repeat-x</code> (csak vízszintes irányba ismétlődő), <code>repeat-y</code> (csak függőleges irányba ismétlődő). Például: <code>background-repeat:no-repeat;</code>
background-position	háttérkép bal felső sarkának koordinátái. Először az x tengelyre vonatkozó, utána az y tengelyre vonatkozó értéket kell beállítani. Például: <code>background-position:0 20px;</code>

## Feladatok

Készítsünk másolatot a `goldi.css` állományból, és csatoljuk ezt az új állományt a már elkészült HTML-oldalakhoz. Módosítsuk az oldal kinézetét saját elképzeléseinknek megfelelően. Próbáljunk ki új színsémákat, igazítási módokat!

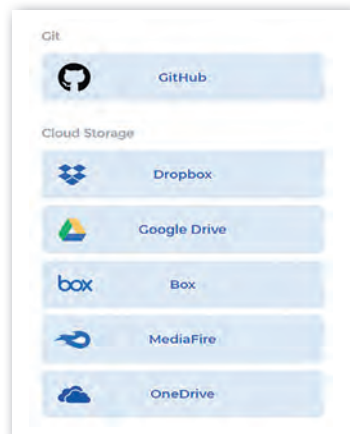
## A statikus honlap publikálása

Ahhoz, hogy az általunk készített statikus honlap mindenki számára elérhető legyen a világhálón, publikálni kell azt egy webes kiszolgálón (webszerveren). Ez azt jelenti, hogy egy segédprogram segítségével csatlakozni kell a webszerverhez, és a megfelelő mappájába fel kell tölteni az összes állományt.

Vannak olyan ingyenes szolgáltatások (pl. fast.io), amelyek lehetővé teszik a statikus honlapok publikálását egy webcímen.

Ilyenkor a regisztráció után meg kell adnunk, hogy milyen webcímen szeretnénk elérhetővé tenni a honlapunkat, majd az ahhoz tartozó összes állományt a portálon ismertetett módon fel kell töltenünk.

Ne feledjük, a kezdőlapot `index.html` néven kell elneveznünk!



- ▶ A fast.io szolgáltató sokféle tárhelyhez tud doménnevet rendelni

### Feladatok, kérdések

1. Derítsük ki, hogy a saját iskolánkban van-e lehetőség a honlapok publikálására. Ha nincs, akkor publikáljuk az elkészített honlapot egy ingyenesen elérhető szolgáltató segítségével (pl. fast.io, gitHub.com)!
2. Három-négy fős csoportokban készítsünk el tetszőleges témában egy honlapot az alábbi elvárások szerint:
  - A weblap legalább két oldalból álljon! Mindegyik oldalba legyen beillesztve egy menü, amellyel el lehet jutni a többi oldalra.
  - Az oldalcímeket precízen töltsük ki mindegyik oldal esetén!
  - A tartalom legalább öt bekezdésből álljon! A bekezdésben lévő szöveget formázzuk meg a tanult címkék segítségével!
  - Legyen az oldalon felsorolási lista és sorszámozott lista is!
  - A tartalom címsorokkal legyen tagolva! Használjunk több címszintet!
  - Legyen egy oldalmenü az oldalon, amellyel az oldalon belüli címsorokra rá lehet ugrani!
  - Legyen az oldalon beillesztve kép és videó! A beillesztésnél figyeljünk az akadálymentességi irányelvekre! (Szorgalmi feladatként hangot is elhelyezhetünk az oldalon, ha van a témához kapcsolódó hangállomány.)
  - Legyen egy olyan táblázat beillesztve, amely legalább három sorból és legalább két oszlopból áll! A táblázatnak legyen felirata! Használjunk fejléc- és adatcellákat is!
  - Csatoljunk a honlaphoz egy stíluslapot!
  - Dolgozzunk jogtiszta forrásból! Felhasználhatunk saját készítésű képeket, videókat, hangállományokat is.
  - Minden felhasznált forrást tüntessünk fel és linkeljünk be!
  - Publikáljuk a weblapot, és a webcímet osszuk meg a többi csoporttal is!

A munka felosztásánál ügyeljünk arra, hogy minden tagnak jusson olyan feladat, amelyben tartalmat kell összegyűjteni, és azt a HTML nyelven le kell írni.