A HTML alapjai és a szabványos weboldalak készítése

Ismerkedés a HTML-nyelvvel

A HTML alapjai

Mi a HTML?

A HTML a HyperText Markup Language rövidítése, amely magyarul hiperszöveg jelölő nyelvet jelent. A HTML tehát nem programozási nyelv, hanem úgynevezett leírónyelv, amely a honlaptartalmak leírására szolgál.
A HTML állomány egyszerű szöveges állomány, a kiterjesztése .html.
A tartalom úgynevezett tagekkel (címkékkel) írható le. Ezeket a böngészőprogram értelmezi, és úgy jeleníti meg az oldalt.
A képzés során a HTML jelenleg legújabb szabványával, a HTML5-el ismerkedünk.

A HTML-tagek (címkék)

A HTML-tagek két, egymás felé fordított relációs jelből (<>) és a közéjük helyezett kifejezésekből állnak. Például a bekezdés (p).
A tageknek alapvetően két típusa van. A páratlan vagy üres HTML-tagek olyan elemek, amelyekhez nem tartozik záró tag (pl.
 - sortörés). A páros HTML-tagek kizárólag egymással párban használhatók. Két elemből, egy nyitó és egy záró tagből állnak, amelyek közre fogják a formázni kívánt tartalmat. A záró tag kódja azonos a nyitó tagével, de egy / jel vezeti be (pl. ... - bekezdés).

Példa a bekezdésre és a sortörésre

```
<p>Ez az első bekezdés.</p>

<p>Ez a <br> második bekezdés.</p>

<p>Ez a harmadik bekezdés.</p>
```

A tagek egymásba ágyazása

A HTML-tagek egymásba ágyazhatók. Például, ha egy szövegrészt félkövéren () és dőlten () szeretnénk megjeleníteni. Mivel nincs dőlt-félkövér formázást eredményező elem, ezért az egyik taget a másikon belül helyezzük. Jelen esetben mindegy melyiket melyiken belül, az egymásba ágyazásnál alapvető szabály, hogy mindig azt a taget kell előbb lezárni, amelyiket később nyitottuk meg (...).

Példa a HTML-tagek egymásba ágyazására

```
<p>
  <b>félkövér</b>
</p>

<p>
  <i>dőlt</i>
</p>

<p>
  <b><i>félkövér és dőlt</i></b>
</p>
```

A tagek csoportosítása a megjelenés módja alapján

Megjelenítési mód szerint a HTML-tagek két nagy csoportra oszthatók.
A blokkszintű elemek előtt és mögött térköz található. Ilyen elem például a <p> tag, hiszen minden egyes bekezdés egymástól térközzel elkülönített új blokkba kerül.
Az inline (sorbeli) elemek soron belül jelennek meg, vagyis a böngésző nem jeleníti meg őket külön blokkokban. Ilyen elem például a szöveget félkövéren megjelenítő <b> tag.

Példa a blokkszintű és az inline (sorbeli) elemek közötti különbségre

```
<p>Ez egy bekezdés.</p>

<p>Ez egy bekezdés.</p>

<p>A bekezdésen belül van egy <b>félkövér</b>
szöveg.</p>
```

Megjegyzések készítése

Alkalmanként fontos lehet a kódokat megjegyzésekkel kiegészíteni. Ez igaz a programozási nyelvekre és a leíró nyelvekre is.
A megjegyzések olyan szöveges tartalmak, amelyek láthatók ugyan a HTML-állományban, de a böngésző nem jeleníti meg őket.
A megjegyzések lehetnek egy- vagy többsorosak is, de minden esetben <!-- és --> jelek közé kell elhelyezni őket.

Példa az egy- és a többsoros megjegyzésekre

```
<p>Ez egy bekezdés.</p>

<!--Egysoros megjegyzés-->

<!--Ez egy
több sorból álló
megjegyzés-->

<p>Ez egy bekezdés.</p>
```

A HTML-paraméterek (attribútumok)

A HTML-tageket esetenként paraméterekkel vagy más néven attribútumokkal látjuk el.
Ezek egy része úgynevezett logikai paraméter, azt adjuk meg vele, hogy az adott beállítás érvényre jusson vagy sem. Ha megadjuk a paraméter nevét, akkor érvényesül, ha nem adjuk meg, akkor nem. Ilyen például a hidden paraméter, amely elrejtésre szolgál.
A paraméterek másik részét értékekkel láthatjuk el. Ezekben az esetekben a paraméterek neve után egyenlőségjel van, amely után az érték idézőjelek között helyezkedik el. Például a lang paraméterrel a nyelv adható meg kód segítségével (lang=”en”).
Egy taghez akár több paraméter is tartozhat, ezeket szóközzel kell elválasztani egymástól.

Példák a hidden, a lang és a title paraméterek használatára

```
<p>Ez egy bekezdés.</p>

<p hidden>Ez egy rejtett bekezdés.</p>

<p lang="en" title="Hiperszöveg-jelölő nyelv">HyperText
Markup Language</p>
```

A HTML5-oldal alapfelépítése

A HTML5-oldalak tartalmát meghatározott struktúrában kell elhelyezni. Ez a struktúra némiképp eltér a HTML-nyelv különböző szabványaiban. Mi jelenleg a HTML5 alapfelépítésével foglalkozunk.
A HTML-kód mindig a dokumentumtípus megadásával kezdődik.

```
<!DOCTYPE html>
```

Az oldal legfelső szintű eleme a HTML-tag, minden más elemet ezen belül kell elhelyezni. Itt adjuk meg az oldal nyelvét is.

```
<html lang=”hu”>
```

Az oldal ezt követően két nagy egységre osztható.
A headben, a fejrészben kell megadni az oldal címét és egyéb metainformációkat (pl. az oldal karakterkódolása).

```
<head>
    <title>Cím</title>
    <meta charset=”utf-8”>
</head>
```

A tényleges tartalom a bodyba, a törzsrészbe kerül.

```
<body>
    tartalom
</body>
```

<!DOCTYPE>

A dokumentumtípus határozza meg, hogy melyik szabvány szerint dolgozunk. Ha nem adjuk meg, a böngészők nem biztos, hogy megfelelően jelenítik meg az oldalt.
A HTML5-szabvány esetén <!DOCTYPE html>

A <html> tag

Az oldalak legfelső szintű eleme. Páros elem, minden más elem ezen belül helyezkedik el. Mindig meg kell adni az oldal nyelvét a lang paraméterrel. Magyar nyelvű honlap esetén a paraméter értéke hu legyen.
Miért szükséges megadni az oldal nyelvét?
A látássérültek többnyire képernyőolvasó programokat vesznek igénybe. Ezek a programok mindig az itt beállított nyelven olvassák fel az adott szöveget. Ha kihagyjuk a nyelv megadását, akkor az angol nyelvű felolvasás lenne az alapértelmezett, azaz a program angol kiejtéssel olvasná fel a magyar szöveget is.

A <head> tag

A head háttérinformációkat tartalmaz az oldalról. Az oldal címe és a metainformációk mellett itt van lehetőség külső stíluslapokat és kliensoldali scripteket kapcsolni az oldalhoz.

A <title> tag

Használata szintén kötelező. Páros elem, a benne megadott szöveg a böngésző címsorában vagy a lapfülön jelenik meg.
A cím mindig utaljon az oldal tartalmára. Ha a felhasználó a könyvjelzők vagy a kedvencek közé teszi az oldalt, akkor a listában ez a szöveg jelenik meg.

Példa a <title> tag használatára

```
<!DOCTYPE html>
<html lang="hu">
<head>
   <title>Oldalcím példa</title>
</head>
```

A <meta> tag

Az oldalra vonatkozó, de a böngészőben közvetlenül nem megjelenő metainformációk egy páratlan elemmel, a <meta> taggel adhatók meg. A cím kivétel, mivel az látható a böngésző címsorában.
A charset paraméterrel az oldal karakterkódolását állíthatjuk be. Alapesetben az UTF-8 kódolást használjuk, mert ez teszi lehetővé, hogy a magyar nyelv és más nyelvek speciális karakterei és ékezetes betűi helyesen jelenjenek meg a böngészőben.
A name paraméter beállításával oldalleírást készíthetünk. A paraméter értéke legyen description, majd következik egy újabb paraméter, a content, amelynek az értéke a leírás szövege.
Ehhez hasonlóan szerzők és kulcsszavak is rendelhetők az oldalhoz. Előbbi esetben a name értéke author, utóbbi esetben keywords.

Példák a karakterkódolás és egyéb metainformációk beállítására

```
<head>

<meta charset="UTF-8">

<meta name="description" content="leírás">

<meta name="keywords" content="Kulcsszó1, Kulcsszó2">

<meta name="author" content="Szerző neve">

</head>
```

A <body> tag

A bodyban vagyis a dokumentumtörzsben szerepel az oldal tényleges tartalma. A páros <body> címkén belül kell elhelyezni az összes tartalmi elemet, például bekezdéseket, címsorokat, listákat, képeket és egyéb elemeket.

Gyakorlati feladat (Microsoft Teams 20230321_1.pdf)

Készíts egy index.html nevű, .html formátumú oldalt.
A fájl karakterkódolása UTF-8 legyen (szövegszerkesztő beállítása).
A dokumentumtípus meghatározásánál add meg, hogy az oldal a HTML5-szabvány szerint készült.
Készítsd el a HTML-oldal alapszerkezetét.
Az oldal legyen magyar nyelvű.
Az oldal karakterkódolása legyen UTF-8.
Az oldal címe legyen “HTML-oldal felépítése”.
Metainformációként add meg a saját neved az oldal szerzőjeként.
A dokumentumtörzsben készíts egy egysoros megjegyzést, amelynek a szövege “Tartalom” legyen.

Gyakran használt, alapvető HTML-tagek

Gyakori csoportosító és oldalszerkezeti elemek

A címsorok

A címsorok hasonló szerepet töltenek be a weboldalakon, mint a kiadványok fejezetcímei. Fontos részei az oldal tartalmi szerkezetének.
A HTML-nyelvben hat címsorszint van, amelyek páros tagekkel állíthatók be.
<h1>, <h2>, <h3>, <h4>, <h5>, <h6>
Az első szintű címsor az oldal címe, a főcím legyen. Ha vannak alcímek, azok legyenek a második szintű címsorok és így tovább.
A böngészők a címsorokat félkövér betűkkel jelenítik meg, méretük a címsor szintjétől függ.
A megfelelő címsorok használata keresőoptimalizálási (SEO) és akadálymentességi szempontból is fontos. A képernyőolvasó programokban megjeleníthető egy olyan speciális, tartalomjegyzékre emlékeztető ablak, amelyben megjelennek az adott oldalhoz tartozó címsorok.

Példák a HTML-nyelvben használt címsorokra

```
<h1>Címsor 1</h1>
<h2>Címsor 2</h2>
<h3>Címsor 3</h3>
<h4>Címsor 4</h4>
<h5>Címsor 5</h5>
<h6>Címsor 6</h6>
```

A bekezdés

A bekezdések a páros <p> tagekkel hozhatók létre. A <p> blokkszintű elem, vagyis a böngésző a bekezdés körül térközt, margót hagy ki. A térköz stíluslap segítségével módosítható.

Példa a bekezdések létrehozására

```
<p>Ez az első bekezdés.</p>
<p>Ez a második bekezdés.</p>
```

A sortörés

A páratlan <br> taggel sortörést állíthatunk be.
Térköz növelésére ne használjuk, azt stíluslap segítségével kell megoldani. A hosszú szövegekben inkább több bekezdés legyen.

Példa a sortörésre

```
<p>Nagy Imola <br>egyéni vállalkozó <br>Debrecen</p>
```

Az elválasztó vonal

A páratlan <hr> taggel elválasztó vonalat jeleníthetünk meg, ha egymástól logikailag elkülönülő egységeket vizuálisan is el szeretnénk választani.

Példa az elválasztó vonal használatára

```
<p>Ez az első bekezdés.</p>
<hr>
<p>Ez a második bekezdés.</p>
```

Szövegformázások

A félkövér és az erősen kiemelt szöveg

Az inline, páros <b> (angol bold) taggel félkövér formázást állíthatunk be. Elsősorban olyan kulcsszavak kiemelésére használjuk, amelyek a látogatót segítik az adott oldal áttekintésében. Az internetezők többsége először úgymond pásztázza az oldalt, hogy eldöntsék, valóban hasznos-e számukra a tartalom.
Az inline, páros tag a <b> taghez hasonlóan a böngészőben félkövér megjelenést hoz létre. Az erős kiemelés azt jelenti, hogy a látássérültek által használt képernyőolvasó programok más hangsúllyal olvassák fel ezeket a szövegrészeket.

Példa a félkövér és az erősen kiemelt szövegek használatára

```
<p><b>Figyelem!</b>November 30-án lejár a jelentkezési határidő.</p>

<p><strong>Vigyázz!</strong>A jelentkezés hamarosan lejár.</p>
```

A dőlt és a hangsúlyos szöveg

Az inline, páros <i> (italic) taggel dőlt formázást állíthatunk be. Akkor érdemes használni, ha egy szövegrész hangulatában, hangnemében különbözik a szöveg egészétől. Webes környezetben a szakkifejezések és az idegen nyelven leírt gondolatok jelölésére használható. Például latin neveket dőlt betűvel szokás formázni.

Az inline, páros tag is dőlt megjelenítést eredményez, de ezzel hangsúlyos kiemelést állíthatunk be, a felolvasóprogramok az így jelölt szövegrészeket más hangsúllyal adják vissza.

Példa a dőlt és a hangsúlyos szövegek használatára

```
<p>Tudtad, hogy az <i>Agathidum vaderi</i> bogárfajt
Darth Vaderről nevezték el?</p>

<p>Tudtad, hogy az <i>Agathidum vaderi</i> bogárfajt
<em>Darth Vaderről</em> nevezték el?</p>
```

Az idejétmúlt szöveg

Az inline, páros <s> taggel a szöveg vízszintes vonallal áthúzva jelenik meg. Az idejétmúlt információk jelölésére érdemes alkalmazni, ha valami miatt a szöveget nem szeretnénk átírni, így a látogatók nyomon követhetik a változást.

Példa az idejétmúlt szövegek jelölésére

```
<p>Indulási időpont:
<s>március 1., 8 h</s> március 1., 11 h</p>
```

Az alsó és a felső index

Egy szövegrészt alsó indexben megjeleníteni az inline, páros <sub> taggel lehetséges. A felső indexhez a <sup> taget kell használni.

Példa az alsó és a felső index beállítására

```
<p>A víz képlete: H<sub>2</sub>O</p>

<p>E=mc<sup>2</sup></p>
```

Gyakorlati feladat (Microsoft Teams 20230328_1.pdf)

Készíts egy .html formátumú oldalt a vízről.
Tartalom: https://codepen.io/webalap/pen/qBZjxKP
Az oldal főcímét (“Víz”) tagold egyes szintű címsorral.
Az oldalon található alcímeket (“Előfordulása a Földön”, “Élettani jelentősége”, “A víz sűrűségmaximuma”) tagold kettes szintű címsorral.
A többi szövegrészt tagold bekezdésekre (a bekezdéseket üres sorok jelzik).
Az utolsó bekezdésben a “Forrás:” után helyezz el egy bekezdésen belüli sortörést.
Az első bekezdésben emeld ki hangsúlyosan, félkövér stílussal a “víz” és a “dihidrogén-monoxid” szavakat.
Az első bekezdésben a víz latin kifejezését emeld ki hangsúlyosan, dőlt stílussal.
Az első bekezdésben keresd meg a víz képletét és a 2-es számjegyet helyezd alsó indexbe.
Az első alcímhez (“Előfordulása a Földön”) tartozó második bekezdés első mondatát formázd félkövér stílusúra.
Az utolsó alcím (“A víz sűrűségmaximuma”) első bekezdésében keresd meg a képletet. A képletben a 6-os és a zárójel utáni 2-es számjegyet tedd felső indexbe.
Az oldal utolsó bekezdésében a “Forrás:” szöveget tedd félkövérré, a hozzá tartozó URL-t pedig dőlten szedetté.

A HTML-oldalak validálása

A HTML-oldalak szabványossága

A HTML-leírónyelvek használati módját különböző szabványok rögzítik. Ha az oldal nem felel meg a szabványoknak, akkor megjelenítési hibák léphetnek fel a böngészőben. A nem szabványos oldalak akadálymentességi problémákkal is járhatnak.
A böngészők alapvetően kísérletet tesznek a nem szabványos oldalak korrigálására, azaz valamilyen módon ezeket az oldalakat is megjelenítik, de egyes böngészők ezt másképp teszik, így eltérően jelenhet meg az oldal.
Keresőoptimalizálási szempontból is sokkal előnyösebb, ha szabványos oldalakat készítünk.

A szabványosság ellenőrzése

A szabványosság ellenőrzését validálásnak nevezik. A HTML5 és más szabványok ellenőrzésére szolgál a The W3C Markup Validation nevű eszköz, ami a https://validator.w3.org oldalon érhető el.

Gyakorlati feladat (Microsoft Teams 20230328_2.pdf)

A feladatban egy meglévő HTML-fájlt kell validálnod.
Forrás: fold.html
Nyisd meg a W3C validátorát: https://validator.w3.org/

A validátorral ellenőrizd, hogy a html fájl megfelel-e a HTML5-szabványnak.
A validátorral keress hibákat és javítsd ki őket.

HTML-szerkesztőalkalmazások bemutatása

Szerkesztőkörnyezetek

Milyen szerkesztőkörnyezetet érdemes választani?

A weboldalak kódolásához igazából nincs szükség speciális szerkesztőprogramra, bármely egyszerű szövegszerkesztő (nem dokumentumszerkesztő) alkalmazással készíthetünk weboldalakat. A hatékony munkához azonban szükséges néhány olyan funkció, amelyekkel nem rendelkezik bármelyik szövegszerkesztő program.

Hasznos funkciók

Szintaxiskiemelés: a szerkesztőkörnyezet eltérő színnel jeleníti meg a címkéket, a paramétereket, az értékeket és a szövegeket. Ezáltal sokkal áttekinthetőbb a kód.
Forráskód sorszámozása: könnyedén odaugorhatunk egy adott sorszámú sorra például ha egy hibát kell kijavítani.
Kódkiegészítés: gépeléskor a program felajánlja a címkéket, a paramétereket és a tipikus értékeket, így kevesebbet kell gépelni kódoláskor. A funkció miatt nem is kell pontosan fejből tudni minden paraméter nevét, elég ha tudjuk, hogyan kezdődik.
Bővíthetőség: további kiegészítők telepítése.

Szerkesztőkörnyezetek

Ingyenes alkalmazások:
- ATOM
- Brackets
- Visual Studio Code
- Notepad++

A Visual Studio (VS) Code

Előnyei:
- több platformon (Windows, Linux, Macintosh) elérhető
- számos speciális funkciója van, amelyek jelentősen megkönnyítik a hatékony kódolást
- támogatja a verziókezelést

A VS Code felépítése és használata

A képernyő részei:

- menüsor
- oldalsáv
- szerkesztőterület
- tevékenységpanel
- státuszsor

A tevékenységpanelen beállítható, hogy milyen információk jelenjenek meg a mellette lévő oldalsávon (pl. Intéző ablak, bővítmények, stb.)
A szerkesztőterületen először Súgó jelenik meg.
Az ablak alján lévő státuszsorban különböző információk jelennek meg, például hogy a kódban van-e javítandó szintaktikai hiba.
A komponensek megjelenítése / eltüntetése a View (Nézet) menü Appearance (Megjelenés) menüpontjában lehetséges.

Mappa megnyitása a VS Code programban

Egy weblap általában nem egyetlen állományból áll, hanem több oldalból, amelyekben képek, videók is lehetnek, stíluslapok tartozhatnak hozzá. A munkát érdemes úgy kezdeni, hogy a File menü Open folder (Mappa megnyitása) menüpontjával megnyitjuk azt a mappát, amelyben dolgozni szeretnénk.
Mappát létrehozni a New folder (Új mappa), fájlt létrehozni a New file (Új fájl) opcióval itt is lehet.

Fájl létrehozása a VS Code programban

A honlapok kezdőlapját index.html néven kell elmenteni. A New file ikonra kattintás után meg kell adni a fájl nevét, majd Entert ütni.
A szerkesztőben egyszerre akár több fájl is megnyitható, ezek külön fül alatt jelennek meg, de a szerkesztőterület meg is osztható a fájlok között.

Az Emmet-rövidítések

Az alapstruktúrát nem kell feltétlenül begépelni, vagy bemásolni.
Az Emmet-rövidítések támogatják a gyors és hatékony kódolást, akár hosszabb kódokat is könnyedén előállíthatunk.
Az alapstruktúrához a html:5 kódot kell használni. A létrejött alapstruktúrában a lang paraméter értéke en, ezt magyar nyelvű oldal esetén módosítani kell hu értékre, valamint meg kell adni az oldal címét a <title> tagben.

Egyéb hasznos funkciók:

- automatikus taglezárás
- tagek, paraméterek elhelyezése
- hibák és figyelmeztetések
- kód összecsukása

- több tag kijelölése és módosítása
- többszörös kurzor használata
- több ablak használata
- bővítmények telepítése
- gyorsbillentyűk (billentyűkombinációk)

CTRL + B	Explorer ablak becsukása/kinyitása
CTRL + Ő	Terminál ablak megnyitása/bezárása
CTRL + S	Adott fájl mentése
CTRL + K, CTRL + C	Kijelölt szöveg elhelyezése megjegyzésbe
CTRL + K, CTRL + U	Megjegyzés megszüntetése a kijelölt szövegről
SHIFT + ALT + A	Megjegyzés be-ki kapcsolása a kijelölt szövegről
ALT + Z	Wordwrap, hosszú sorok automatikus tördelése
ALT + SHIFT +Lefele nyíl	Az adott sor duplikálása lefele
ALT + SHIFT +Felfele nyíl	Az adott sor duplikálása felfele
CTRL + D	Kijelöli a következő előfordulását is (egymás után többször is megnyomható)
CTRL + SHIFT + L	Kijelöli az összes előfordulását a fájlban
CTRL + ALT + felfele/lefele nyíl	Segítségével egymás alatti sorokban egyszerre írható be adott pozícióra ugyanaz a szöveg
CTRL + F	Keresés
CTRL + C	Másolás
CTRL + X	Kivágás
CTRL + V	Beillesztés
F1	VS Code command line elérése (például innen érhető el) Wrap Individual Lines with Abbreviations Emmetek is
CTRL + Ű	Több ablak megnyitása
ALT + SHIFT + 0 (nulla)	Ablakok vízszintes/függőleges elhelyezése egymás alá/mellé

Gyakorlati feladat (Microsoft Teams 20230328_3.pdf)

Hozz létre egy index.html fájlt és hajtsd végre az alábbi utasításokat.
A következő feladatokat a VS Code program beépített Emmet-parancsaival vagy billentyűkombinációival próbáld megoldani.
Alakítsd ki az oldal szerkezetét a HTML5-szabványnak megfelelően.
Az oldal főcíme legyen “Kedvenc gyümölcseim” és helyezd el egy egyes szintű címsorba. A böngésző címsorában is ez a cím jelenjen meg.
Készíts két bekezdést, amelyek tartalma egy-egy lorem töltelékszöveg legyen.
Egymás alá gépelj be négy gyümölcsöt és többszörös kurzor használatával készíts ezekből is bekezdéseket.
Ugyanezzel a módszerrel minden második bekezdést formázd kék színűre.
A bekezdések alá készíts billentyűkombinációk segítségével egy megjegyzést, és írd bele a nevedet a megjegyzésbe.

A stíluslapok használata

Bevezetés a stíluslapok használatába

A stíluslapok szerepe

A stíluslapok (CSS) használata

A HTML-nyelv a weboldalak tartalmát írja le (strukturális réteg), a megjelenésüket egy másik szabvánnyal, a CSS-el alakíthatjuk ki (reprezentációs réteg). A CSS a Cascading Style Sheets rövidítése, ami lépcsőzetes vagy rangsorolt stíluslapokat jelent. A weblapok megjelenésén túl ezekkel állíthatjuk be az optimális megjelenést a kisebb felbontású eszközökön (pl. okostelefon), de akár nyomtatásra szánt stíluslapokat is készíthetünk velük.
A stílusleírások egyébként közvetlenül a HTML-állományokba helyezhetők, ott is kétféle módon, de úgynevezett külső stíluslapok is készíthetők, ami javasolt. Utóbbi esetben egy szöveges állományt készítünk css kiterjesztéssel.
A stílusszabványok közül a CSS3 a legfrissebb.

A stíluslaphasználat előnyei:
- haladó formázási és elrendezési lehetőségeket nyújt
- az oldalak egységesen és konzisztensen jeleníthetők meg
- a weboldalak egyszerűen karbantarthatók (a módosítások minden oldal megjelenésére hatással lesznek)
- haladó akadálymentességi technikák alkalmazhatók
- az oldal különbözőképpen jeleníthető meg nagy képernyőn, kis képernyőn, nyomtatásban, stb.

A stíluslapok típusai:
- a szerző által készített stíluslapok
- felhasználói stíluslapok: a böngészőkben beállítható egyéni stíluslap
- a böngészőkben definiált stíluslapok: a böngészőkben az alapértelmezett megjelenésért felelnek

<u>A CSS-szintaxis</u>

A CSS-szabály

Az egyes elemek formázását úgynevezett CSS-szabályokkal lehet megadni. Minden CSS-szabály két részből áll, szelektor és deklarációs blokk.
A szelektor az úgynevezett kijelölő rész. Ezzel adjuk meg, hogy az oldalon melyik elemet vagy elemeket kívánjuk formázni. Egy szelektor segítségével kijelölhető például az oldal összes bekezdése.
A deklarációs blokk segítségével a kijelölt elem vagy elemek formázásának leírását adjuk meg.

Például:
```
h1{
        color: blue;
```

}
A h1 a szelektor, a deklarációs blokkban megadott formázás a HTML-állományon belül minden első szintű címsorra érvényes lesz.
A deklarációs blokk a kapcsos zárójelek között elhelyezkedő rész. A color tulajdonsággal a betűszínt adhatjuk meg, ami jelen esetben kék lesz.

A tulajdonság-érték párok

A deklarációs blokkon belül a tulajdonságokat és a hozzájuk tartozó értékeket kettősponttal (:) kell elválasztani egymástól. Egy blokkba több tulajdonság-érték pár is elhelyezhető. Ebben az esetben az utolsó kivételével pontosvesszővel (;) kell őket lezárni.

Példa a több deklarációt tartalmazó CSS-szabályra

A képen látható CSS-szabály az egyes szintű címsort kék színnel, a törzsszövegnél háromszor nagyobb betűméretben jeleníti meg.

```
<head>
<style>
  h1 {
    color: blue;
    font-size: 300%
    }
</style>
</head>

<body>
  <h1>Címsor</h1>
  <p>Ez egy bekezdés.</p>
</body>
```

A megjegyzések

Míg a HTML-ben <!-- és --> karaktersorozatok között kell elhelyezni a megjegyzéseket, addig a CSS-ben /* és */ között. A megjegyzések lehetnek egy- és többsorosak is.
A CSS állományok sokszor nagyon hosszúak, ezért karbantartásuk sokszor nehézséget okoz. A könnyebb átláthatóság miatt érdemes az állományt megjegyzésekkel különböző szakaszokra tagolni valamilyen logika szerint.

Példa a CSS-nyelvben készített megjegyzésekre

```
/* Címsor beállítása */
h1 {
   color: blue;
   font-size: 300%
   }
```

A stíluslapok csatolási lehetőségei

A stíluslapok többféle módon rendelhetők a HTML-állományokhoz.
A deklaráció szerepelhet inline (elemközi) stílusként. Ezt a style (stílus) paraméterrel adhatjuk meg. Az elemközi stílus mindig az adott elemre, illetve annak beágyazott elemeire van hatással. Csak ritkán, speciális esetekben használjuk, különben a tartalom nehezen különíthető el a megjelenéstől.
A második lehetőség az úgynevezett beágyazott stíluslap. Ebben az esetben a head részben <style> tagek közé kell megadni a formázásokat.
A deklarációkat külső állományban is elhelyezhetjük és ezt érdemes használni. A külső állomány hivatkozását egy link taggel a head részben vagy egy style elemen belüli @import szabállyal adhatjuk meg. Ennek a megoldásnak jelentős előnye, hogy több HTML-oldalon is hivatkozhatunk ugyanarra a külső stíluslapállományra.

Az inline (elemközi) stílusmegadás

Közvetlenül az adott elem és annak beágyazott elemei formázhatók a style paraméterrel.
Hátránya, hogy kevésbé átláthatóvá teszi a kódolást, mert a HTML-tagek és a stílusmegadások keverednek egymással.
Indokolt esetek a használatára:

- a tartalomkezelő rendszerben nincs jogunk a központi stíluslap módosítására, azonban mégis szükség van elemek módosítására
- HTML-formátumú e-mail készítése
- a megjelenés dinamikus módosítása (pl. JavaScript módosítja az oldal megjelenését)

Példa az inline stílusmegadásra

```
<p style="color: green">Ez egy zöld bekezdés.</p>
```

A beágyazott stílusmegadás

Az oldal head részében található, ebben az esetben már elkülönül a HTML-tagektől, de még mindig ugyanabban az állományban helyezkedik el.

Példa a beágyazott stílusmegadásra

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

<style>
  h1 {
  color: blue;
  }
</style>
</head>

<body>
   <h1>Címsor 1</h1>
   <p>Lorem ipsum dolor sit amet, consectetur adipisicing
elit.</p>
   <h1>Címsor 2</h1>
   <p>Lorem ipsum dolor sit amet, consectetur adipisicing
elit.</p>
 </body>
 </html>
```

A külső stíluslap alkalmazása

Ha több oldalból álló weblapot készítünk, mindenképp érdemes külön CSS-állományban elhelyezni a stílusszabályokat.
A stíluslapra egy link utal az oldal fejlécében. A link tagnek két kötelező paramétere van. A stíluslap elérési útvonalát (URL) a href, a stíluslap és a HTML-állomány kapcsolatát a rel paraméter értékeként kell megadni. Ez utóbbi értéke stíluslap esetén stylesheet. Opcionálisan a fájl típusa is megadható a type paraméterrel. Értéke stíluslap esetén text/css.

Példa a külső stíluslap alkalmazására

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
  <title>Külső CSS</title>
  <link rel="stylesheet" type="text/css" href="pelda.css">
</head>

<body>
  <h1>Címsor 1</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  <h1>Címsor 1</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
  <h1>Címsor 1</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
</body>
</html>
```

A CSS-szelektorok (kijelölők)

A szelektorok (kijelölők)

A type selector (típuskijelölő)

A type selectort (típus- vagy elemkijelölő) akkor érdemes használni, ha minden adott típusú elemre ugyanazt a stílust szeretnénk beállítani, például minden címsort vagy bekezdést egységesen kívánunk formázni. A szelektor kódja megegyezik a formázni kívánt rész HTML-tag kódjával.

Például:

```
<style>
        h1{
                font-style: italic;
        }
        p{
                color: blue;
        }
</style>
```

A class selector (osztálykijelölő)

A class selectorokra (osztálykijelölő) akkor van szükség, ha egy stílust az elemek meghatározott csoportjára kell beállítani. A szelektort egy pont (.) vezeti be, ezt az osztály neve követi. Az osztálynév tartalmazhat betűket, számokat, aláhúzásjelet és

kötőjelet is, azonban betűvel kell kezdődnie. Az osztályokat a HTML-állományban a class paraméterrel hozhatjuk létre, és egy elemet akár több osztályba is besorolhatunk.

Például:

```
<style>
	.fontos{
		color: red;
	}
</style>
<h1 class=”fontos”>Címsor</h1>
```

A class selectorok más szelektorokhoz társítása

A class selectorok szükség esetén más típusú szelektorokkal is társíthatók. Ezáltal akár több elemnél is megadhatjuk ugyanazt az osztálynevet más-más tulajdonságokkal.

Például:

```
<style>
	h1.fontos{
		color: red;
	}
	p.fontos{
		border: 1px solid red;
	}
	.vastag{
		font-weight: bold;
	}
</style>
<h1 class=”fontos”>Címsor</h1>
<p class=”fontos vastag”>Bekezdés</p>
```

Az ID selector (azonosítókijelölő)

A HTML-állomány elemeihez szükség esetén egyedi azonosítók is rendelhetők az id paraméterrel. Ennek értékeként egy nevet szükséges megadni. A név nem tartalmazhat szóközt, és az adott oldalon belül egyedinek kell lennie. Az egyedi azonosítót kettőskereszttel (#) jelölhetjük ki.

Például:

```
<style>
	#elso{
		background-color: lightblue;
```

```
        }
</style>
<p id=”elso”>Bekezdés</p>
<p>Bekezdés</p>
```

A deklarációk csoportosítása

A csoportok kijelölése

Ha több elemnél is ugyanaz a deklaráció szerepel, akkor ezeket akár össze is vonhatjuk. A formázni kívánt részekre mutató szelektorokat egymástól vesszővel elválasztva kell megadni, majd utána a rájuk vonatkozó, közös deklarációs szabályokat.

Például ha az első, a második és a harmadik címsorra is ugyanazt a sans-serif betűtípust szeretnéd alkalmazni, akkor a következők szerint csoportosíthatod a deklarációkat:

```
h1 {font-family: sans-serif}
h2 {font-family: sans-serif}
h3 {font-family: sans-serif}
```

```
h1, h2, h3 {font-family: sans-serif}
```

A tulajdonság-érték párok egybevonása és a shorthand (összevont) megadás

Egy szelektorhoz akár több deklarációs szabály is rendelhető. Ebben az esetben az egyes szabályokat egymástól pontosvesszővel (;) elválasztva kell megadni. A tulajdonság-érték párok egy deklarációs blokkban való csoportosításának speciális esete az úgynevezett shorthand (összevont) megadás. A tulajdonságok jelentős részének két, egymástól kötőjellel elválasztott tagból álló elnevezése van. Az első általánosan utal a formázott részre, míg a konkrét beállítást a második jelenti. Például a font kezdetű tulajdonságok a betűk megjelenítését érintik, ezenbelül a weight a vastagságot, a size a méretet, a family a betűcsaládot jelöli. Összevont stílusmegadás esetén ilyenkor elegendő a font tulajdonság után az értékeket egymástól szóközökkel elválasztva megadni.

Példa a font kezdetű tulajdonságok shorthand megadására

```
h1 {font-family: helvetica}
h1 {font-size: 12pt}
h1 {font-weight: bold}

h1 {font-family: helvetica;    /*Tulajdonság-érték párok
egybevonása*/
    font-size: 12pt;
    font-weight: bold}

h1 {font: bold 12pt helvetica}  /*shorthand (összevont)
megadás*/
```

Gyakorlati feladat (Microsoft Teams 20230328_4.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/KKzyrmj
Készíts az egyes szintű címsorhoz egy type selectort, amely a cím betűszínét sötétzöldre (darkgreen) színezi.
Készíts az alcímekhez (kettes szintű címsor) egy type selectort, amely az alcím betűszínét sötétvörösre (darkred) színezi.
Készíts a félkövér stílussal kiemelt elemekhez egy type selectort, amely a félkövér szavak színét sötétkékre (darkblue) színezi.

Gyakorlati feladat (Microsoft Teams 20230328_5.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/gOrXNgz
Készíts a négy évszakhoz egy-egy class selectort és színezd ki az évszakra jellemző színekkel a megfelelő kettes szintű címsort az alábbi táblázat alapján:

Évszak	Osztálykijelölő neve	Betűszín
tavasz	spring	green (zöld)
nyár	summer	goldenrod (arany)
ősz	autumn	firebrick (téglavörös)
tél	winter	dodgerblue (kék)

A class selectorok elkészítése után rendeld a stílusokat a megfelelő alcímekhez a HTML-fájlban is.
A főcímre helyezz el egy main_title azonosítót, és készíts hozzá a stíluslapon belül egy ID selectort. A stílussal állítsd be a főcím betűszínét a kedvenc színedre. A szín

kiválasztásánál figyelj a betű és a háttérszín kontrasztjára. Ha nincs kedvenc színed, akkor színezd sötétebb kékre (midnightblue).

A mértékegységek és a színek

A mértékegységek

A relatív és az abszolút mértékegységek

A relatív mértékegységek egy másik tulajdonság értékétől függenek. Például egy szülőelem mérete hatással van az oda beágyazott elem tulajdonságaira. A leggyakoribb ilyen mértékegység a százalék.
Az abszolút mértékegységek értékei konkrét elemeket jelölnek, és így függetlenek más elemek értékeitől. Ilyen mértékegység például a képpont (px).
Az értékeket jelölő számokat mindig egybe kell írni a mértékegységek jelével, pl. 12px.

Az alapmértékegységek (CSS3)

Gyakran használt abszolút mértékegységek:
- inch (hüvelyk) - in
- centiméter - cm
- pont (1/72in) - pt
- pixel (képpont) - px

Gyakran használt relatív mértékegységek:
- em: az aktuális betűmérethez képest megadott méretet jelzi, pl. 1.2rem az aktuális betűméret 1,2-szeresét jelenti.
- %
- rem: a gyökérelem (HTML-oldalaknál a html elem) betűméretéhez képest megadott méretet jelenti.

A színek

A színjelölések színnevekkel

A színek beállításának egyik módja, ha a color tulajdonság után értékként angol színneveket adunk meg (https://www.w3.org/TR/css-color-3/#svg-color). Többnyire alapszíneknél használjuk ezt a megoldást, de egzotikusabb színek is megadhatók így.

Példa a színek színnevekkel való beállítására

```
h1 {
color: blue;
}

h2 {
color: lightsalmon
}
```

Az RGB-színkoordináta-rendszer

A színek színkoordináta-rendszerekkel is megadhatók. Az RGB-színkoordináta-rendszerben három szín található, a vörös (R), a zöld (G) és a kék (B) és ezek keverésével állítható elő a többi szín. Az egyes komponensek 0 és 255 közötti értékeket vehetnek fel. Ha mindhárom érték a maximális 255, akkor fehér színt kapunk, ha a minimális 0, akkor feketét. Az RGB kód százalékos értékekkel is megadható.

Példa az RGB-kód alkalmazására

```
<head>
<style>
  h1 {
  color: rgb(49,190,69);
  }
  h2 {
  color: rgb(92%,40%,50%)
  }
</style>
</head>

<body>
  <h1>Címsor 1</h1>
  <h2>Címsor 2</h2>
</body>
```

Az átlátszatlanság

Használhatunk úgynevezett RGBA-kódot is. Az utolsó paraméter az alfa-csatornára, vagyis az átlátszatlanság mértékére utal. 0 és 1 közötti értéket vehet fel. Ha az érték 1, akkor a formázott rész átlátszatlan marad, 0 érték esetén teljesen átlátszó lesz.

Példa az alfa-csatorna használatára

```
<head>
<style>
  h1 {
  color: rgba(0,0,0,1);
  }
  h2 {
  color: rgba(0,0,0,0.8);
  }
  h3 {
  color: rgba(0,0,0,0.5);
  }
  h4 {
  color: rgba(0,0,0,0.3);
  }
  h5 {
  color:rgba(0,0,0,0.1)
  }
</style>
</head>

<body>
  <h1>Címsor 1</h1>
  <h2>Címsor 2</h2>
  <h3>Címsor 3</h3>
  <h4>Címsor 4</h4>
  <h5>Címsor 5</h5>
</body>
```

A hexadecimális színkód

A kód minden egyes tízes számrendszerbeli értékét tizenhatos számrendszerbe kell konvertálni, majd az így kapott számjegyeket egymás után kell írni. A kód elé kettőskeresztet (#) kell tenni. A hexadecimális színkód hossza esetenként lerövidíthető, ha egy kódon belül ugyanazok a számjegyek ismétlődnek, elég őket egyszer leírni.
A képszerkesztő programok általában többféle színkoordináta-rendszerben is megmutatják a színkódokat.

Példa a hexadecimális színkód használatára

```
<head>
<style>
  h1 {
  color: rgb(255,136,0);
  }
  h2 {
  color: #ff8800
  }
  h3 {
  color: #f80
  }
</style>
</head>

<body>
  <h1>Címsor 1</h1>
  <h2>Címsor 2</h2>
  <h3>Címsor 3</h3>
</body>
```

A HSL-színkoordináta-rendszer

H (hue) - árnyalat: 0 és 360 közötti értéket vehet fel, ezek a színkörön látható különböző fokoknak felelnek meg.
S (saturation) - telítettség: 0 és 100% közötti értéket vehet fel.
L (lightness) - fényesség: 0 és 100% közötti értéket vehet fel.
A (alpha channel) - átlátszatlanság: 0 és 1 közötti értéket vehet fel (opcionális).

Példa a HSL-színkóddal való beállítására

```
<head>
<style>
  h1 {
  color: hsl(32,100%,50%)
  }
  h2 {
  color: hsl(32,100%,50%,0.5)
  }
</style>
</head>

<body>
  <h1>Címsor 1</h1>
  <h2>Címsor 2</h2>
</body>
```

Színkoordináta-rendszerek közötti konvertálás: https://colorizer.org/

A betűk formázása

A betűtípusok beállítása

A betűtípust a font-family tulajdonsággal állíthatjuk be. Ennek értékeként érdemes megadni egy listát, ami tartalmazza a használni kívánt betűtípusokat. A lista tetszőleges számú, egymástól vesszővel elválasztott elemből állhat. A böngésző a sorban első betűtípussal jeleníti meg a tartalmat, ha az elérhető, ha nem elérhető, akkor a sorban következőket használja. A lista végén érdemes egy általános betűcsaládot megadni. Ha egyik betűtípus sem érhető el, akkor a böngésző alapértelmezett betűtípussal jeleníti meg a szöveget.
Ha a választott betűtípus nevében szóköz szerepel, akkor az egész elnevezést aposztrófok közé kell tenni, pl. ‘Times New Roman’.
Ha olyan egyedi betűtípusokat szeretnénk használni, amelyek valamennyi platformon egységesen jelennek meg, akkor igénybe vehetjük a Google Fonts szolgálgatást (https://fonts.google.com/). Kimásolhatjuk a használni kívánt betűcsaládok kódjait, link segítségével be is ágyazhatjuk őket a CSS állományba.

Példa a betűtípusok beállítására

```
<head>
<style>
  p {font-size: 24pt;}

  #szoveg1 {
    font-family: Times, 'Times New Roman', Georgia, serif;
    }
    #szoveg2 {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    }
    #szoveg3 {
    font-family: Courier, 'Loucida Console', monospace;
    }
</style>
</head>

<body>
  <p id="szoveg1">Szöveg</p>
  <p id="szoveg2">Szöveg</p>
  <p id="szoveg3">Szöveg</p>
</body>
```

Gyakran használt általános betűcsaládok

Serif: elsősorban nyomtatásban használt, talpas betűtípusok
Sans-serif: képernyőn jól mutató, modern megjelenésű, talp nélküli betűtípusok
Monospace: azonos szélességű karaktereket használó, írógéppel írt hatást keltő betűtípus
Cursive: kézírásra emlékeztető, nehezen olvasható, elsősorban dekoratív célokat szolgáló betűtípusok

A betűméret beállítása

A betűméretet a font-size tulajdonsággal adhatjuk meg.
Abszolút méret megadása:

- kulcsszavak: xx-small, x-small, small, medium (alapértelmezett), large, x-large, xx-large, xxx-large
- mértékegységek: px

Relatív méret megadása:

- kulcsszavak: smaller (kisebb), larger (nagyobb)
- relatív mértékegységek: em, %

Példa a betűméret abszolút megadására

```
<head>
<style>
   .fs1 {font-size: xxx-large}
   .fs2 {font-size: xx-large}
   .fs3 {font-size: x-large}
   .fs4 {font-size: large}
   .fs5 {font-size: medium}
   .fs6 {font-size: small}
   .fs7 {font-size: x-small}
   .fs8 {font-size: xx-small}
</style>
</head>

<body>
   <p class="fs1">Szöveg</p>
   <p class="fs2">Szöveg</p>
   <p class="fs3">Szöveg</p>
   <p class="fs4">Szöveg</p>
   <p class="fs5">Szöveg</p>
   <p class="fs6">Szöveg</p>
   <p class="fs7">Szöveg</p>
   <p class="fs8">Szöveg</p>
</body>
```

Példa a betűméret relatív megadására

```
<head>
<style>
   .fs1 {font-size: larger}
   .fs2 {font-size: smaller}
   .fs3 {font-size: 32px}
   .fs4 {font-size: 300%}
   .fs5 {font-size: 1.5em}
</style>
</head>

<body>
   <p class="fs1">Szöveg</p>
   <p class="fs2">Szöveg</p>
   <p class="fs3">Szöveg</p>
   <p class="fs4">Szöveg</p>
   <p class="fs5">Szöveg</p>
</body>
```

A betűstílus beállítása

A betűstílust a font-style tulajdonsággal adhatjuk meg. Értékei a következők lehetnek:

- oblique: dőlt stílust eredményez az eredeti betűk megdöntésével
- italic: az aktuális betűtípus dőlt változatát jeleníti meg, ha van ilyen a betűkészletben
- normal: a betűk stílusát normál stílusúra állítja (vissza)

Példa a betűstílus beállítására

```
<head>
<style>
   .fst1 {font-style: normal}
   .fst2 {font-style: oblique}
   .fst3 {font-style: italic}
</style>
</head>

<body>
   <p class="fst1">Szöveg</p>
   <p class="fst2">Szöveg</p>
   <p class="fst3">Szöveg</p>
</body>
```

A betűsúly (betűvastagság) beállítása

A font-weight tulajdonsággal adhatjuk meg a betűk súlyát, azaz a vastagságát. A bold érték beállítása erős kiemelést eredményez, de számértékek is megadhatók. A bold kiemelés a 700-as értéknek felel meg. A megadható értékek: 100, 200, 300, 400 (alapértelmezett), 500, 600, 700, 800 és 900. A gyakorlatban kevés olyan betűtípus van, ahol mind a kilenc vastagság változat megtalálható.

Példa a betűsúly beállítására

```
<head>
<style>
   .fw1 {font-weight: 400}
   .fw2 {font-weight: 700}
   .fw3 {font-weight: 900}
</style>
</head>

<body>
   <p class="fw1">Szöveg</p>
   <p class="fw2">Szöveg</p>
   <p class="fw3">Szöveg</p>
</body>
```

A betűváltozat beállítása

A font-variant tulajdonsággal betűváltozatok állíthatók be. A leggyakrabban használt érték a small-caps, amivel kiskapitális megjelenítés állítható be. Itt is használható a normal érték, ezzel normál stílusúra állíthatók vissza a szöveg betűi.

Példa a betűváltozat beállítására

```
<head>
<style>
  .fw1 {font-variant: small-caps}
  .fw2 {font-variant: normal}
</style>
</head>

<body>
  <p class="fw1">Szöveg</p>
  <p class="fw2">Szöveg</p>
</body>
```

A betűk nyújtása

A betűk nyújtását a font-stretch tulajdonsággal állíthatjuk be. A tulajdonság értékét kulcsszavakkal és százalékos számértékekkel is megadhatjuk:

- sűrítés:
  - ultra-condensed - 50%
  - extra-condensed - 62,5%
  - condensed - 75%
  - semi-condensed - 87,5%
- normál:
  - normal - 100%
- ritkítás:
  - semi-expanded - 112,5%
  - expanded - 125%
  - extra-expanded - 150%
  - ultra-expanded - 200%

A font-stretch tulajdonság nem minden betűtípusra van hatással, illetve előfordulhat, hogy egy betűtípussal csak ritkított vagy sűrített hatást lehet elérni.

Példa a betűk nyújtására

```
<head>
<style>
   .fsr1 {font-stretch: ultra-condensed}
   .fsr2 {font-stretch: extra-condensed}
   .fsr3 {font-stretch: condensed}
   .fsr4 {font-stretch: semi-condensed}
   .fsr5 {font-stretch: normal}
   .fsr6 {font-stretch: semi-expanded}
   .fsr7 {font-stretch: expanded}
   .fsr8 {font-stretch: extra-expanded}
   .fsr9 {font-stretch: ultra-expanded}
</style>
</head>

<body>
   <p class="fsr1">Szöveg</p>
   <p class="fsr2">Szöveg</p>
   <p class="fsr3">Szöveg</p>
   <p class="fsr4">Szöveg</p>
   <p class="fsr5">Szöveg</p>
   <p class="fsr6">Szöveg</p>
   <p class="fsr7">Szöveg</p>
   <p class="fsr8">Szöveg</p>
   <p class="fsr9">Szöveg</p>
</body>
```

Gyakorlati feladat (Microsoft Teams 20230418_1.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/OJNzMQJ
Készíts egy type selectort az egyes szintű címsorhoz, és állítsd be, hogy a főcím kiskapitális stílusú legyen.
Készíts egy type selectort a kettes szintű címsorhoz, és állítsd be, hogy az alcím betűmérete az alapértelmezettnél nagyobb (x-large) legyen.
A bekezdést állítsd Verdana betűtípusra egy type selector segítségével.
A HTML-fájlban pár tag is van, amelyek közül egyes elemekhez felkover, míg a többihez dolt osztályt rendeltek.

- *Készíts egy felkover nevű class selectort, és állítsd be, hogy a kijelölt elem félkövér stílusú és az alapértelmezett betűmérethez képest 10%-kal nagyobb betűméretű legyen.*
- *Készíts egy dolt nevű class selectort, és állítsd be, hogy a kijelölt elem dőlt stílusú legyen.*

Keresd fel a Google Fonts hivatalos weboldalát, és válassz ki egy neked tetsző, Handwriting kategóriába tartozó betűtípust. A kiválasztott betűtípust a felkínált importálás segítségével illeszd be a stíluslap elejére és alkalmazd a szöveg címsoraira.

Az általános csoportosító elemek és a dobozmodell

Az általános csoportosító elemek

A és a tag használata

.

A blokkszintű és az inline (sorbeli) elemek

A blokkszintű elemek előtt és után a böngészőprogram térközt (vagy margót) hagy ki. Ilyen elem például a bekezdések megadására szolgáló <p> tag hiszen a böngészők a bekezdéseket egymástól térközökkel különítik el.
Az inline (sorbeli) elemek ezzel szemben soron belül jelennek meg, a böngésző nem töri őket külön blokkokra. Inline elem például a szöveget félkövéren szedő <b> tag.

Az általános tárolóelem használata (tag)

A tag olyan blokkszintű, páros elem, amelynek nincs speciális jelentése, hanem elemek csoportosítására szolgál. Segítségével könnyedén formázhatjuk egyszerre az elemeket.

Példa a tag használatára

```
<div style="font-style: italic;">
  <p>Lorem ipsum dolor, sit amet consectetur adipisicing elit.</p>
  <p>Officiis obcaecati voluptas culpa quia aut odio amet.</p>
  <p>Quisquam exercitationem doloribus provident porro omnis cim.</p>
</div>
```

Az általános tárolóelem használata (tag)

A páros tag szintén általános csoportosító elem, de inline elemek csoportosítására használható. Segítségével például több szó formázható egyszerre.

Példa a tag használatára

```
<p> A HTML a <span lang="en" style="font-style:italic;">
HyperText Markup Language</span> rövidítése.</p>
```

Az elemek megjelenítési módja

Az elemek alapértelmezett blokkszintű vagy soron belüli megjelenését a stíluslapok használatával felül is írhatjuk. Ehhez a display tulajdonságot kell beállítani:

- none: az elem nem jelenik meg
- block: az elem blokkszintű elemként jelenik meg
- inline: az elem sorbeli elemként jelenik meg

Példa a blokkszintű bekezdéselemek inline-ná alakítására

```
<head>
<style>
  p {
  display:inline;
  }
</style>
</head>

<body>
  <p>Ez egy bekezdés.</p>
  <p>Ez egy bekezdés.</p>
</body>
```

Példa az inline elemek blokkszintűvé alakítására

```
<head>
<style>
  b {
  display:block;
  }
</style>
</head>

<body>
  <p>Ez egy <b>bekezdés.</b></p>
</body>
```

Gyakorlati feladat (Microsoft Teams 20230418_2.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/wvGOGVd
A általános tárolóelem segítségével alkoss egy csoportot a főcímből és az azt követő két bekezdésből.
Az általános tárolóelemre alkalmazd a doboz nevű class selectort.
Foglald csoportba az alcímet és az azt követő négy bekezdést is, illetve helyezd külön szakaszba (div) a forrás bekezdését is.
A Világbajnokság alcímhez tartozó szövegben a világbajnokok nevét helyezd el egy-egy span HTML-elembe.
A span elemekhez készíts egy type selectort és a segítségével alkalmazd a következő formázásokat a világbajnokok nevére:

- *betűszín legyen sötétkék*
- *a nevek legyenek kiskapitális stílusúak*
- *a betűméret legyen 20%-kal nagyobb az alapértelmezettnél*

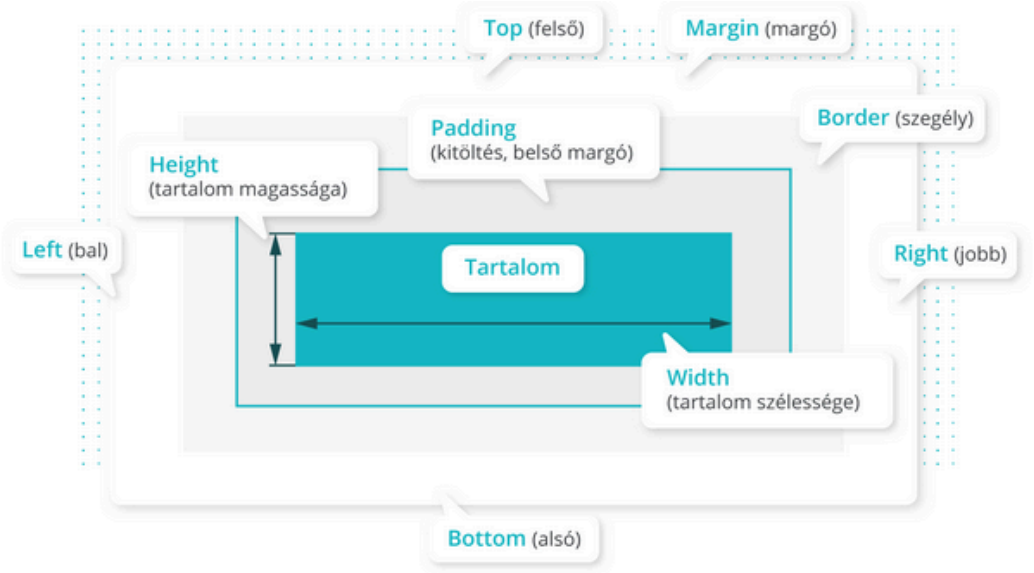
Gyakorlati feladat (Microsoft Teams 20230418_3.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/eYZXdZN
A display CSS-tulajdonság segítségével állítsd be, hogy az alapesetben blokkszintű bekezdések inline elemként jelenjenek meg a weblapon.
Állítsd be, hogy a dőlt stílusú szavak blokkszintű elemként jelenjenek meg az oldalon.
A forrás ne jelenjen meg az oldalon. Ehhez készíts egy eltunik nevű class selectort, és állítsd be a stíluson belül a display tulajdonságnak megfelelő értéket. Ezután alkalmazd az utolsó bekezdésre a class selectort.

A dobozmodell (formázásmodell)

Az alapértelmezett dobozmodell

A böngészőkben megjelenő elemek úgynevezett dobozokban helyezkednek el.



A legbelső rész maga a tartalom. Ennek a szélességét a width, magasságát a height tulajdonsággal adhatjuk meg.
A padding (kitöltés, belső margó) a tartalom és az oldal szegélye között helyezkedik el. A szegély és a többi elem közötti térközt pedig marginnak (margó) nevezzük.

A szegély beállítása

Egységes szegély beállítása

Az elemeket különböző méretű és stílusú szegélyekkel láthatjuk el. A border tulajdonsággal ugyanolyan szegélyt állíthatunk be mind a négy oldalra. Ennek értékeként a szegély vastagságát, stílusát és színét szükséges megadni.
A szegély vastagsága beállítható egyrészt angol kulcsszavakkal (thin - vékony, medium - közepes, thick - vastag), másrészt pixelben.0 px érték esetén az elemnek nem lesz szegélye.
A szegély stílusát kulcsszavakkal állíthatjuk be:

- solid (folytonos szegély)
- dotted (pontozott szegély)
- dashed (szaggatott szegély)
- double (dupla szegély)
- groove (barázdált hatású szegély)
- ridge (a képernyő síkjához képest kiemelt hatású szegély)
- inset (süllyesztett hatású dobozt eredményez)

- outset (kiemelt hatású dobozt eredményez)

A szegélyek színét angol színnevekkel vagy színkódokkal adhatjuk meg.

Példa az egy elemhez tartozó egyforma szegélyek beállítására

```
<head>
<style>
  .példa1 {
  border: 1px solid blue;
  }

  .példa2 {
  border: thick double #ddd;
  }
</style>
</head>

<body>
  <p class="példa1">Ez egy bekezdés.</p>
  <p class="példa2">Ez egy bekezdés.</p>
</body>
```

Oldalanként eltérő szegély

Egyedi szegélyt is beállíthatunk mind a négy oldalon. Ebben az esetben a border tulajdonság kiegészül egy kötőjellel, ezt a formázni kívánt oldal angol neve követi:

- top (felső)
- right (jobb)
- bottom (alsó)
- left (bal)

Példa az oldalanként eltérő szegély beállítására

```
<head>
<style>
  p {
  border-top: 1px dotted black;
  border-right: 4px solid red;
  border-bottom: 1px dashed gray;
  border-left: 1px solid red;
  }
</style>
</head>

<body>
  <p>Ez egy bekezdés.</p>
</body>
```

Példa az oldalanként eltérő szegély beállítására

Több gépeléssel jár ugyan, de az eltérő oldalakat úgy is megadhatod, hogy a deklarációs blokkon belül elkülöníted egymástól a szélességre (width), a stílusra (style) és a színre (color) vonatkozó beállításokat.

```
<head>
<style>
  p {
  border-top-width: 2px;
  border-top-style: dotted;
  border-top-color: green;
  }
</style>
</head>

<body>
  <p>Ez egy bekezdés.</p>
</body>
```

A szegély lekerekítése

A deklarációs blokkon belül a border-radius tulajdonsággal lekerekíthetjük a szegélyek sarkát. A tulajdonság értéke megadható pixelben és százalékban is. Egyedi lekerekítéseket is beállíthatunk:

- border-top-left-radius (bal felső sarok)
- border-top-right-radius (jobb felső sarok)
- border-bottom-right-radius (jobb alsó sarok)
- border-bottom-left-radius (bal alsó sarok)

Példa a szegély lekerekítésére

```
<head>
<style>
  .példa1 {
  border: 1px solid blue;
  border-radius: 5px;
  }

  .példa2 {
  border: 1px solid blue;
  border-top-left-radius: 10px;
  border-top-right-radius: 5px;
  border-bottom-right-radius: 1px;
  border-bottom-left-radius: 3px
  }
</style>
</head>

<body>
<div>
  <p class="példa1">Ez egy bekezdés.</p>
  <p class="példa2">Ez egy bekezdés.</p>
</div>
</body>
```

A szélesség és a magasság beállítása

A deklarációs blokkon belül az elemek szélességét a width, a magasságát a height tulajdonsággal állíthatjuk be. Értékét megadhatjuk abszolút és relatív mértékegységekkel is. Például a width: 70% tulajdonság-érték pár azt jelenti, hogy a formázott elem szélessége a szülőelem szélességének 70%-a lesz.
Megadható az elemek minimális és maximális mérete is a min-width, a max-width, a min-height és a max-height tulajdonságokkal.

Példa a szélesség és a magasság beállítására

```
<head>
<style>
  .példa1 {
  width: 150px;
  height: 100px;
  border: 1px solid blue;
  }

  .példa2 {
  width: 70%;
  height: 100px;
  border: 1px solid blue;
  }

  .példa3 {
  width: 70%;
  min-width: 200px;
  max-width: 600px;
  border: 1px solid blue;
  }
</style>
</head>

<body>
  <p class="példa1">Ez egy bekezdés.</p>
  <p class="példa2">Ez egy bekezdés.</p>
  <p class="példa3">Ez egy bekezdés.</p>
</body>
```

A padding (kitöltés, belső margó) használata

A deklarációs blokkon belül a padding tulajdonsággal állíthatunk be belső margót vagy más néven kitöltést. Ez a tartalom és a szegély közötti távolságot jelenti. Ha a padding után egy értéket adunk meg, akkor a doboz mind a négy oldalára ugyanaz a beállítás lesz érvényes.
A kitöltés értéke külön-külön is beállítható mind a négy oldalra. Ebben az esetben a padding tulajdonság kötőjel után kiegészül a formázni kívánt oldal angol nevével:

- padding-top
- padding-right
- padding-bottom
- padding-left

Példa a padding (kitöltés, belső margó) használatára

```
<head>
<style>
  .példa1 {
  width: 200px;
  border: 1px solid blue;
  padding: 30px;
  }

  .példa2 {
  width: 200px;
  border: 1px solid blue;
  padding-top: 20px;
  padding-right: 10px;
  padding-bottom: 0px;
  padding-left: 15px;
  }
</style>
</head>

<body>
  <p class="példa1">Ez egy bekezdés.</p>
  <p class="példa2">Ez egy bekezdés.</p>
</body>
```

A margó

A margin tulajdonság

Egyes elemeknek, mint például bekezdéseknek, alapértelmezett margójuk van. Ez akkor érvényesül, ha külön nem állítjuk be az adott elemnek a margóméretet. A margó értékét a margin tulajdonsággal módosíthatjuk vagy akár le is nullázhatjuk. Az értéket pixelben adhatjuk meg.
A margók beállításánál figyelni kell a függőleges margók összevonására. A stíluslapszabvány szerint a szebb megjelenésért a böngészőnek úgy kell összevonnia az egymás alatti dobozok alsó és felső margóit, hogy a két margóérték közül csak a nagyobb érvényesüljön.

Példa a margók használatára

```
<head>
<style>
  .példa1 {
  width: 200px;
  border: 1px solid blue;
  }

  .példa2 {
  width: 200px;
  border: 1px solid blue;
  margin: 0px
  }

  .példa3 {
  width: 200px;
  border: 1px solid blue;
  margin: 10px
  }
</style>
</head>

<body>
  <p class="példa1">Ez egy bekezdés.</p>
  <p class="példa1">Ez egy bekezdés.</p>
  <p class="példa2">Ez egy bekezdés.</p>
  <p class="példa2">Ez egy bekezdés.</p>
  <p class="példa3">Ez egy bekezdés.</p>
  <p class="példa3">Ez egy bekezdés.</p>
</body>
```

Az eltérő margóértékek beállítása

Az elemek egyes oldalaira akár egyedi margóértékeket is beállíthatunk, ha a margin tulajdonságot egy kötőjel után kiegészítjük a formázni kívánt oldal angol nevével:

- margin-top (felső margó)
- margin-right (jobb oldali margó)
- margin-bottom (alsó margó)
- margin-left (bal oldali margó)

Példa az eltérő margóértékek beállítására

```
<head>
<style>
  p {
  border: 1px solid blue;
  margin-top: 20px;
  margin-right: 10px;
  margin-bottom: 5px;
  margin-left: 15px;
  }
</style>
</head>

<body>
  <p>Ez egy bekezdés.</p>
  <p>Ez egy bekezdés.</p>
</body>
```

Középre igazítás margóval

A blokkszintű elemeket margóbeállítással akár középre is igazíthatjuk. Ehhez a bal és a jobb oldali margónak az auto értéket kell beállítani. Ha alul és felül azonos margóméretet használunk, például 20 px értékű margót, akkor a deklarációs blokkban elég a margin: 20px auto értéket megadni.

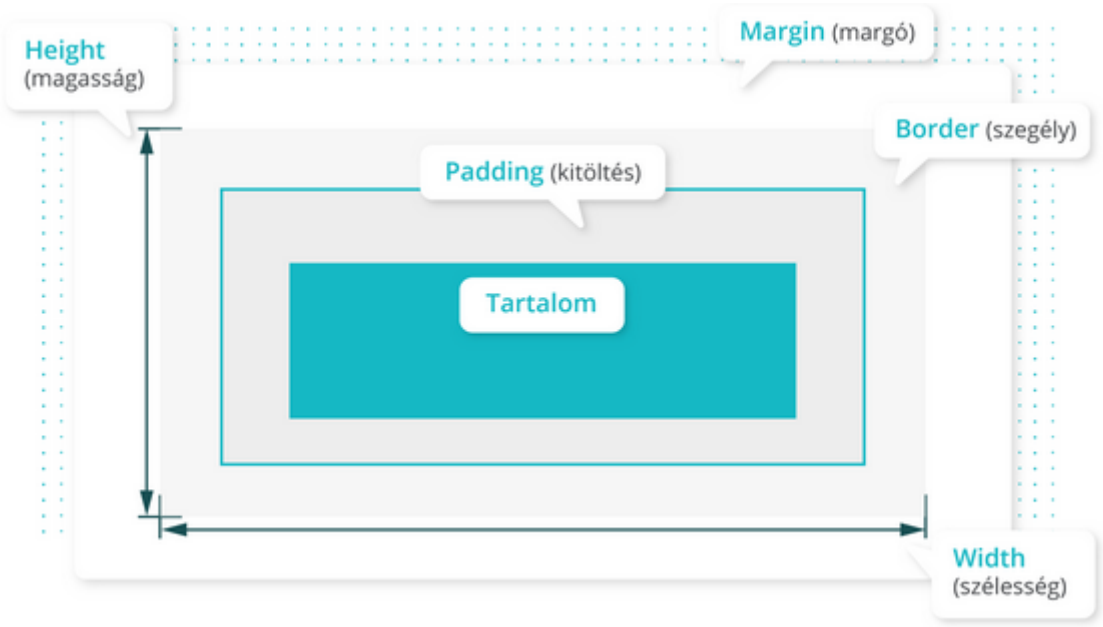
Példa a margókkal való középre igazításra

```
<head>
<style>
  .példa1 {
  border: 1px solid blue;
  width: 200px;
  margin-top: 20px;
  margin-right: auto;
  margin-bottom: 15px;
  margin-left: auto;
  }

  .példa2 {
  border: 1px solid blue;
  width: 200px;
  margin: 0 auto;
  }

  .példa3 {
  border: 1px solid blue;
  width: 200px;
  margin: 20px auto;
  }
</style>
</head>
```

```
<body>
  <p class="példa1">Ez egy bekezdés.</p>
  <p class="példa1">Ez egy bekezdés.</p>
  <p class="példa2">Ez egy bekezdés.</p>
  <p class="példa2">Ez egy bekezdés.</p>
  <p class="példa3">Ez egy bekezdés.</p>
  <p class="példa3">Ez egy bekezdés.</p>
</body>
```

A dobozméret módosítása

Nem mindig kényelmes az alapértelmezett dobozmodellt használni. Sokszor praktikusabb meghatározni, hogy egy doboznak a szegéllyel együtt mekkora a szélessége és a magassága. Úgy válthatunk erre a számítási módra, ha a deklarációs blokkban a box-sizing: border-box tulajdonság-érték párt adjuk meg. A content-box érték felel meg az alapértelmezett beállításnak.



Példa a border-box és a content-box értékek közötti különbségre

```
<head>
<style>
  div {
  border: 10px solid blue;
  width: 200px;
  padding: 10px;
  margin: 10px;
  }

  #blokk1 {box-sizing: content-box}
  #blokk2 {box-sizing: border-box}
</style>
</head>

<body>
  <div id="blokk1">Ez egy bekezdés.</div>
  <div id="blokk2">Ez egy bekezdés.</div>
</body>
```

Gyakorlati feladat (Microsoft Teams 20230425_1.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/QWNoBZV

Az inner-container azonosítójú tárolóelemre helyezz el egy 15 px vastag, folytonos, sötétzöld színű szegélyt.
Az inner-container azonosítójú tárolóelem belső margóját állítsd 10 px-re.
A bekezdéseket lásd el egy 10 px vastag, narancssárga színű, folytonos szegéllyel.
Az inner-container azonosítójú tárolóelem box-sizing értékét állítsd border-box-ra.
A bekezdés box-sizing értékét állítsd át border-boxra.

Gyakorlati feladat (Microsoft Teams 20230425_2.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/yLOwxOm
Készíts mindhárom dobozhoz egy-egy azonosítókijelölőt.
Az első doboz sarkait 25 px-el, a másodikét 50 %-al kerekítsd le.
A harmadik doboz egyes sarkait különböző mértékekkel kerekítsd le:

- *bal felső sarok 10 px*
- *jobb felső sarok 20 px*
- *jobb alsó sarok 30 px*
- *bal alsó sarok 40 px*

A szöveg formázása

A szövegigazítás beállítása

A szöveg vízszintes igazítási módját a deklarációs blokkon belül a text-align tulajdonsággal állíthatjuk be. Értékek:

- left (balra)
- center (középre)
- right (jobbra)
- justify (sorkizárás)

A böngészőkben a balra igazítás az alapértelmezett beállítás.

Példa a szövegigazítás beállítására

```
<head>
<style>
   .példa1 {text-align: left}
   .példa2 {text-align: center}
   .példa3 {text-align: right}
   .példa4 {text-align: justify}
</style>
</head>

<body>
  <p class="példa1">Lorem ipsum dolor sit, amet consectetur adipisicing elit. Vero quaerat minima ea soluta tenetur fugit vel.</p>
  <p class="példa2">Lorem ipsum dolor sit, amet consectetur adipisicing elit. Vero quaerat minima ea soluta tenetur fugit vel.</p>
  <p class="példa3">Lorem ipsum dolor sit, amet consectetur adipisicing elit. Vero quaerat minima ea soluta tenetur fugit vel.</p>
  <p class="példa4">Lorem ipsum dolor sit, amet consectetur adipisicing elit. Vero quaerat minima ea soluta tenetur fugit vel.</p>
</body>
```

A szövegdekoráció beállítása

A deklarációs blokkon belül a text-decoration tulajdonság használatával szövegdekoráció (aláhúzás, felülvonás, stb.) állítható be. Ennek értékeként a vonal típusát, stílusát és színét adhatjuk meg.
Vonaltípusok:

- underline (aláhúzás)
- overline (felülvonás)
- line through (áthúzott szöveg)
- none (normál, dekoráció nélküli szöveg)

Vonalstílusok:

- solid (folytonos)
- double (dupla)
- dotted (pontozott)
- dashed (szaggatott)
- wavy (hullámos)

Vonalszínek:

- angol színnevek

- színkódok

Példa a szövegdekoráció alkalmazására

```
<head>
<style>
   .td1 {text-decoration: underline}
   .td2 {text-decoration: underline overline blue}
   .td3 {text-decoration: underline wavy red}
   .td4 {text-decoration: line-through solid green}
</style>
</head>

<body>
   <p class="td1">Ez egy bekezdés.</p>
   <p class="td2">Ez egy bekezdés.</p>
   <p class="td3">Ez egy bekezdés.</p>
   <p class="td4">Ez egy bekezdés.</p>
</body>
```

A szövegtranszformáció

A szöveg transzformációját a deklarációs blokkban megadott text-transform tulajdonsággal állíthatjuk be. Értékei lehetnek:

- uppercase (minden betűt nagybetűvé alakít)
- lowercase (minden betűt kisbetűvé alakít)
- capitalize (minden szó első betűjét nagybetűssé alakítja)
- none (a szöveget meghagyja az eredeti formájában)

Példa a szövegtranszformáció alkalmazására

```
<head>
<style>
   .tt1 {text-transform: none}
   .tt2 {text-transform: capitalize}
   .tt3 {text-transform: uppercase}
   .tt4 {text-transform: lowercase}
</style>
</head>

<body>
   <p class="tt1">Lorem ipsum DOLOR sit amet.</p>
   <p class="tt2">Lorem ipsum DOLOR sit amet.</p>
   <p class="tt3">Lorem ipsum DOLOR sit amet.</p>
   <p class="tt4">Lorem ipsum DOLOR sit amet.</p>
</body>
```

Az első sor behúzása

A deklarációs blokkon belül a text-indent tulajdonsággal állíthatjuk be a bekezdések első sorának behúzását. Az értéket relatív és abszolút mértékegységgel is megadhatjuk.

Példa az első sor behúzására

```
<head>
<style>
   .ti1 {text-indent: 2em}
   .ti2 {text-indent: 16px}
   .ti3 {text-indent: 10%}
</style>
</head>

<body>
   <p class="ti1">Lorem, ipsum dolor sit amet consectetur adipisicing elit. Nisi doloremque officiis quibusdam dolores quaerat libero natus voluptate? Fugiat cum asperiores accusamus! Quae et sed aliquam facere minima quaerat, incidunt suscipit.</p>
   <p class="ti2">Lorem, ipsum dolor sit amet consectetur adipisicing elit. Nisi doloremque officiis quibusdam dolores quaerat libero natus voluptate? Fugiat cum asperiores accusamus! Quae et sed aliquam facere minima quaerat, incidunt suscipit.</p>
   <p class="ti3">Lorem, ipsum dolor sit amet consectetur adipisicing elit. Nisi doloremque officiis quibusdam dolores quaerat libero natus voluptate? Fugiat cum asperiores accusamus! Quae et sed aliquam facere minima quaerat, incidunt suscipit.</p>
</body>
```

A sormagasság beállítása

A deklarációs blokkon belül a line-height tulajdonsággal sormagasságot állíthatunk be. Ez a deklaráció különbözőképpen hat a blokkszintű és az inline elemekre:

- blokkszintű elemek: a sorok minimális magasságát jelöli
- inline elemek: az elem tényleges magasságát jelöli

A sormagasság mind relatív, mind abszolút mértékegységekkel megadható.

Példa a sormagasság beállítására

```
<head>
<style>
   .lh1{line-height: 1.5em}
   .lh2 {line-height: normal}
   b {line-height: 50px}
   .lh3 {line-height: 30px}
</style>
</head>

<body>
   <p class="lh1">Lorem, ipsum dolor sit amet consectetur adipisicing elit. Nisi doloremque officiis quibusdam dolores quaerat libero natus voluptate? Fugiat cum asperiores accusamus! Quae et sed aliquam facere minima quaerat, incidunt suscipit.</p>
   <p class="lh2">Lorem, ipsum <b> dolor </b> sit amet consectetur adipisicing elit. Nisi doloremque officiis quibusdam dolores quaerat libero natus voluptate? Fugiat cum asperiores accusamus! Quae et sed aliquam facere minima quaerat, incidunt suscipit.</p>
   <p class="lh3">Lorem, ipsum dolor sit amet consectetur adipisicing elit. Nisi doloremque officiis quibusdam dolores quaerat libero natus voluptate? Fugiat cum asperiores accusamus! Quae et sed aliquam facere minima quaerat, incidunt suscipit.</p>
</body>
```

A betűköz beállítása

A deklarációs blokkon belül a letter-spacing tulajdonság szolgál a betűköz beállítására. Pozitív érték esetén a betűk távolodnak egymástól, negatív érték esetén közelednek. A betűközt relatív és abszolút mértékegységgel is megadhatjuk, de a relatív em mértékegységet érdemes használni, mert ennek alapegysége a használt betűméret.

Példa a betűköz beállítására

```
<head>
<style>
   .ls1 {letter-spacing: normal}
   .ls2 {letter-spacing: 0.1em}
   .ls3 {letter-spacing: 4px}
   .ls4 {letter-spacing: -0.05em}
</style>
</head>

<body>
   <p class="ls1">Lorem, ipsum dolor sit amet consectetur adipisicing elit.</p>
   <p class="ls2">Lorem, ipsum dolor sit amet consectetur adipisicing elit.</p>
   <p class="ls3">Lorem, ipsum dolor sit amet consectetur adipisicing elit.</p>
   <p class="ls4">Lorem, ipsum dolor sit amet consectetur adipisicing elit.</p>
</body>
```

A szóköz beállítása

A szavak közötti távolság mértékét a deklarációs blokkon belül a word-spacing tulajdonsággal állíthatjuk be. A szóköz beállításához hasonlóan negatív értékkel csökkenthető a távolság, pozitív értékkel pedig növelhető. A szóköz méretét relatív és abszolút mértékegységgel is megadhatjuk, de praktikusabb itt is az em mértékegység használata.

Példa a szóköz beállítására

```
<head>
<style>
   .ws1 {word-spacing: normal}
   .ws2 {word-spacing: 0.3em}
   .ws3 {word-spacing: 10px}
   .ws4 {word-spacing: -0.1em}
</style>
</head>

<body>
   <p class="ws1">Lorem, ipsum dolor sit amet consectetur adipisicing elit.</p>
   <p class="ws2">Lorem, ipsum dolor sit amet consectetur adipisicing elit.</p>
   <p class="ws3">Lorem, ipsum dolor sit amet consectetur adipisicing elit.</p>
   <p class="ws4">Lorem, ipsum dolor sit amet consectetur adipisicing elit.</p>
</body>
```

Gyakorlati feladat (Microsoft Teams 20230502_1.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/oNxVPmq
A főcím legyen nagybetűs és középre igazított. A szavak közötti távolságot állítsd be 10 px nagyságúra.
Az alcímek legyenek kiskapitális stílusúak. Alkalmazz 1 px nagyságú betűritkítást az alcímek betűire.
Az alcímek legyenek aláhúzottak. Az aláhúzás stílusa legyen dupla vonal, a színe pedig sötétvörös.
A bekezdések első sorát húzd be 10 px-lel. Tedd a bekezdések szövegét sorkizárttá. A sor magasságát állítsd be 25 px nagyságúra.
Készíts egy forras nevű class selectort. A stílussal igazítsd jobbra a szöveget, és tedd dőlt stílusúvá. A class selectort alkalmazd az utolsó bekezdésre.

Képek és illusztrációk

Képek és illusztrációk beillesztése

Képek a weboldalakon

A képformátumok

A HTML-szabvány konkrétan nem írja le, milyen képformátumok illeszthetők be a weboldalakra. A különböző böngészőprogramok különböző kép- és animációs formátumokat támogathatnak. A legszélesebb körben támogatott formátumok:

- PNG (Portable Network Graphics)
- JPEG (Joint Photographic Expert Group)
- SVG (Scalable Vector Graphics)
- GIF (Graphics Interchange Format)

Az <img> tag használata

Képeket az <img> taggel illeszthetünk be. Paraméterek:

- src: a kép forrása (kötelező)
- alt: helyettesítő szöveg (kötelező)
- title: súgó (opcionális)
- width: a kép szélessége képpontban (opcionális)
- height: a kép magassága képpontban (opcionális)

Az src paraméter használata

Az src kötelező paraméter, a kép forrását adjuk meg vele. A kép elérési útvonalát abszolút és relatív módon is meg lehet adni a HTML-állományhoz képest, de a relatív megadást érdemes használni.
Ha a kép ugyanabban a mappában található, mint a HTML-állomány, akkor az src paraméterben elég csak a képfájl nevét megadni.
Ha a kép például egy kepek nevű mappában található a HTML-állomány mellett, akkor először a mappa nevét, majd egy perjelet és utána a képfájl nevét kell megadni.
Ha a kép a HTML-állományt tartalmazó mappa szülőmappájában található, akkor először két ponttal kell hivatkozni a szülőmappára, majd egy perjel és utána a képfájl nevét kell megadni.
Ha a kép interneten keresztül elérhető, akkor az src paraméter értékeként az URL-t kell megadni.

Az alt paraméter használata

Kötelezően megadandó paraméter az alt is, amely helyettesítő szöveget jelent.
Az egyik fontos szerepe az akadálymentességi funkció, a látássérült felhasználókat segítő képernyőolvasó program az itt megadott szöveget olvassa fel, a felhasználó ezen szöveg alapján tudja meg, mi látható a képen.
A másik fontos szerepe a helyettesítő funkció, ha a böngésző valamiért nem tölti be a képet, akkor az alt paraméterben megadott szöveget jeleníti meg helyette.

Példa az alt paraméter használatára

```
<img src="esernyo.jpg" alt="Színes esernyők egy belgrádi utcában">
```

A title paraméter használata

Nem kötelező paraméter, mégis hasznos lehet, ha egy képhez magyarázó szöveget vagy súgót szeretnénk rendelni.
A megadott magyarázat szövegbuborékban jelenik meg a böngészőben, ha a kép fölé visszük az egeret.

Példa a title paraméter használatára

```
<body>

<img src="esernyo.jpg" title="Belgrádban készült fotó" alt="Színes esernyők egy belgrádi utcában">

</body>
```

A width és a height paraméter használata

A beillesztett kép méretét is megadhatjuk, de a paraméter használata nem kötelező. Ha megadjuk, a böngésző a képet nem a valós fizikai méretében, hanem a megadott képpont paramétereknek megfelelően jeleníti meg.
A width paraméter a kép szélességét, a height a magasságát jelenti. Ha közülük csak az egyik paramétert adjuk meg, a másikat a böngészőprogram automatikusan alakítja át úgy, hogy ne változzanak a kép méretarányai.
Ha mindkét paramétert megadjuk, akkor figyelnünk kell rá, hogy ne torzuljon a képarány.

Példa a width és a height paraméter használatára

Ha a width és a height paramétereket is megadod, figyelned kell, hogy ne torzuljon a képarány, ahogy az alábbi példán is látható.

```
<body>

<img src="esernyo.jpg" width="200" height="100"
alt="Színes esernyők egy belgrádi utcában">

</body>
```

Illusztrációk a weboldalakon

A <figure> tag használata

Illusztrációk beillesztésére a HTML5-szabványban megjelent egy új tag, a figure. Ezen tárolóelemen belül kép, szöveg, forráskód, videó helyezhető el.
Az illusztrációkhoz feliratot is társíthatunk a <figcaption> tag segítségével. Egy <figure> tagen belül akár több elemet is elhelyezhetünk, így akár két vagy több kép is lehet illusztráció közös képaláírással.

Példa a <figure> és a <figcaption> használatára

```
<body>
<figure>
  <img src="esernyo.jpg" alt="Színes esernyők egy belgrádi
utcában">
  <figcaption>Színes esernyők egy belgrádi
utcában</figcaption>
</figure>
</body>
```

Gyakorlati feladat (Microsoft Teams 20230502_2.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/JjXqWeb
A főcím után illeszd be a kölyökkutyát ábrázoló képet.
Előfordulhat, hogy a kép valamiért nem jeleníthető meg, ezért a tanult HTML-attribútum segítségével állítsd be a kép helyettesítő szövegét “Kutya” feliratra.

Szintén HTML-attribútum felhasználásával oldd meg, hogy jelenjen meg a “Kiskutya” felirat, ha a kép fölé viszed az egeret.
Készíts a képhez egy elemkijelölőt, és állítsd be a kép szélességét 400 px-re, míg a magasságát 380 px nagyságúra.
A képet keretezd be egy 5 px vastagságú, sötétvörös (darkred) színű, folytonos szegéllyel.
A képet igazítsd vízszintesen középre, függőleges eltartását (margóját) állítsd be 20 px nagyságúra.

Gyakorlati feladat (Microsoft Teams 20230502_3.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/jOqoBzd
A képet helyezd el egy figure tárolóelembe.
A képhez készíts képaláírást, amelynek szövege legyen “Lukas Schweizer képe: Green ears of wheat”.
A tárolóelem tartalmát igazítsd középre.
A képaláírás legyen dőlt stílusú.

Gyakorlati feladat (Microsoft Teams 20230502_4.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/jOqoYLQ
A feladat megoldása során használd a stíluslapban előre definiált ID selectorokat.
A képeket keretezd be a következő információk alapján:
- *Az első kép szegélyét állítsd szaggatott stílusúra.*
- *A második kép szegélye legyen 3 px vastag, narancssárga (orange) színű, folytonos vonal.*
- *A harmadik kép szegélye legyen 7 px vastag, kék színű (R: 51, G: 51, B: 170), dupla szegélyű vonal.*

Kerekítsd le a képeket a következő beállítások alapján:
- *Az első kép minden sarkát 50%-ban kerekítsd le.*
- *A második kép minden sarkát 25 px-re kerekítsd le.*
- *A harmadik kép bal felső és jobb alsó sarkát 30 px-re kerekítsd le.*

A második képre vízszintes irányban állíts be 20 px-es külső margót.

A háttérbeállítások

A háttérszín

A háttérszín beállítása

Az egyes elemek háttérszínét a background-color tulajdonsággal állíthatjuk be. Ennek értéke a színek angol neve, hexadecimális, RGB-, vagy HSL-kódja lehet.

Nagyon fontos, hogy a háttérszín-előtérszín párosítást mindig úgy kell kiválasztani, hogy a szöveg jól olvasható legyen. A web-akadálymentesítési útmutatóban leírtak szerint a kontrasztarány értékének legalább 4,5-nek kell lennie ahhoz, hogy a szöveg jól olvasható legyen.

Példa a háttérszín beállítására

```
<head>
<style>
   .bg1 {background-color: lightyellow;}
   .bg2 {background-color: black; color:white;}
   .bg3 {background-color: rgb(200,100,70);}
</style>
</head>

<body>
   <p class="bg1">Lorem, ipsum dolor sit amet consectetur adipisicing elit.</p>
   <p class="bg2">Lorem, ipsum dolor sit amet consectetur adipisicing elit.</p>
   <p class="bg3">Lorem, ipsum dolor sit amet consectetur adipisicing elit.</p>
</body>
```

Az átlátszatlanság beállítása

Az átlátszatlanság mértékének megadásához RGBA-színadatokat kell használni, ahol az utolsó tulajdonságban az 1-es érték jelenti a teljes átlátszatlanságot, a 0 pedig a teljes átlátszóságot.
Ha például az oldal háttere fehér és a bekezdések háttérszínét barnára állítjuk, az alfa-csatorna csökkenő értékének hatására az egyes bekezdések háttere egyre átlátszóbbá válik, 0 értéknél pedig egyáltalán nem is látszik, hiszen teljesen átlátszó.

Példa az átlátszatlanság beállítására

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Példa</title>

<style>
  .bg1 {background-color: rgba(200,100,70,1)}
  .bg2 {background-color: rgba(200,100,70,0.8)}
  .bg3 {background-color: rgba(200,100,70,0.6)}
  .bg4 {background-color: rgba(200,100,70,0.4)}
  .bg5 {background-color: rgba(200,100,70,0.2)}
  .bg6 {background-color: rgba(200,100,70,0)}
</style>
</head>
```

```
<body>
  <p class="bg1">Lorem, ipsum dolor sit amet consectetur
adipisicing elit.</p>
  <p class="bg2">Lorem, ipsum dolor sit amet consectetur
adipisicing elit.</p>
  <p class="bg3">Lorem, ipsum dolor sit amet consectetur
adipisicing elit.</p>
  <p class="bg4">Lorem, ipsum dolor sit amet consectetur
adipisicing elit.</p>
  <p class="bg5">Lorem, ipsum dolor sit amet consectetur
adipisicing elit.</p>
  <p class="bg6">Lorem, ipsum dolor sit amet consectetur
adipisicing elit.</p>
</body>
</html>
```

A háttérkép

A háttérkép beállítása

Háttérképet a background-image tulajdonsággal állíthatunk be, értékként a kép URL-jét kell megadni. A none értékkel letilthatjuk a háttérkép megjelenését.
A böngészőprogramok alapértelmezetten mozaikszerűen rendezik el a kiválasztott háttérképet, ez azonban felülbírálható.

Fontos, hogy olyan háttérképet használjunk, amin olvasható marad a tartalom.

Példa a háttérkép beállítására

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Példa</title>

<style>
  p {background-color: rgba(255,255,255,0.9)}
  body {background-image:url(CSS3.png);}
</style>
</head>
```

A háttérkép ismétlődése

A háttérkép ismétlődésének a módját a background-repeat tulajdonsággal állíthatjuk be. Az alapértelmezett értéke a repeat, ez jelenti a mozaikszerű elrendezést.
background-repeat: no-repeat; - a háttérkép nem ismétlődik és a böngésző bal felső sarkába kerül
background-repeat: repeat-x; - a háttérkép csak vízszintesen ismétlődik a böngészőben
background-repeat: repeat-y; - a háttérkép csak függőlegesen ismétlődik a böngészőben

A háttérkép kezdőpozíciója

A háttérkép kezdőpozícióját a background-position tulajdonság-érték párral változtathatjuk meg. Az első érték a vízszintes, a második a függőleges igazítást állítja be. Megadhatunk hosszúságértéket és százalékos értéket is.
A background-position tulajdonság értékét egyrészt kulcsszavakkal (top left - bal felső, center top - felső középső, center center - középső, left bottom - bal alsó, right bottom - jobb alsó), másrészt különböző mértékegységekkel (px, %) lehet megadni. Utóbbi esetben két értéket kell megadni, az első a vízszintes, a második a függőleges mértéket határozza meg.

A háttérkép mérete

A háttérkép méretét a background-size tulajdonsággal állíthatjuk be.

A hosszúságértékeket px-ben vagy a szülőelem méretéhez képest százalékban mérve tetszőlegesen állíthatjuk be. Ha csak egy értéket adunk meg, azt a böngésző a kép szélességére vonatkoztatja és a magasságot a képaránynak megfelelően automatikusan változtatja meg. Ha két értéket adunk meg egymás után, a szélességet és a magasságot is beállítjuk.
A mértékegységek mellett kulcsszavakat is használhatunk:

- cover: a tartalmazóelem szélességére vagy magasságára méretezi a háttérképet, de a képarányt nem változtatja meg
- contain: a böngészőben a legkisebb olyan méretre állítja be a képet, hogy az teljesen elférjen az elemben, a képarány itt sem változik
- auto: a háttérkép megtartja eredeti képarányát

Példa a háttérkép méretének px-ben való megadására

```
<style>
   body {background-image: url(https://i.ibb.co/7GJx2yz
/esernyo.jpg);
   background-repeat: no-repeat;
   background-size: 170px 80px;}
</style>
```

Példa a háttérkép méretének kulcsszavakkal való beállítására

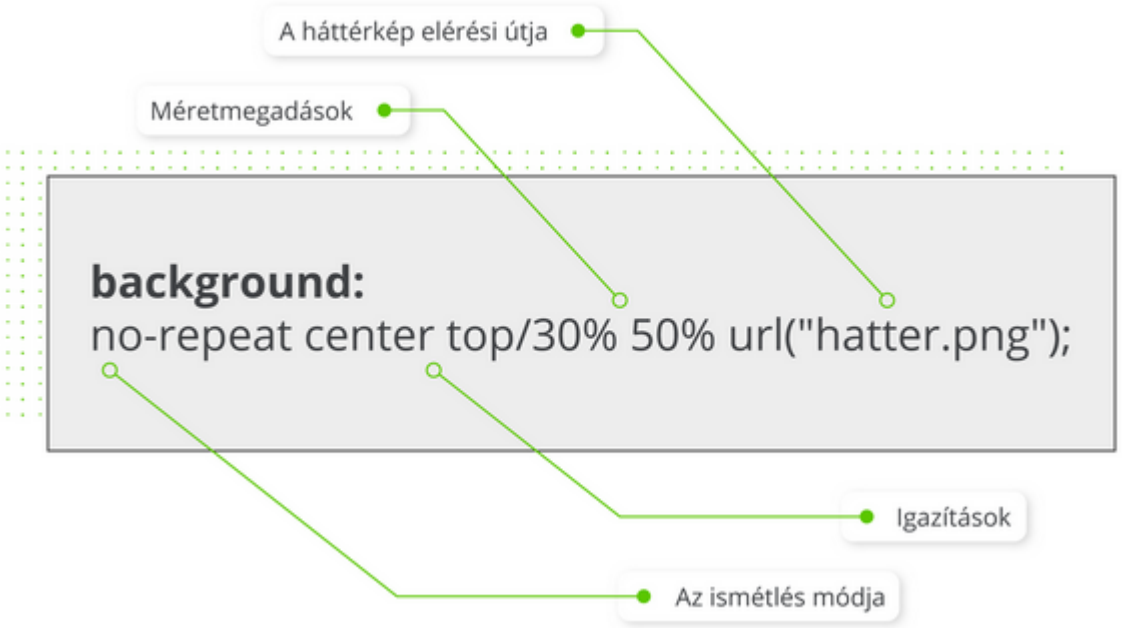
```
<style>
   body {background-image: url(https://i.ibb.co/7GJx2yz
/esernyo.jpg);
   background-repeat: no-repeat;
   background-size: cover;}
</style>
```

A háttérkép rögzítése

A háttérkép viselkedését az oldal gördítésekor a background-attachment tulajdonsággal állíthatjuk be.
A scroll érték (alapértelmezett beállítás) megadásával a hátteret az elemhez rögzítjük, így a kép a tartalommal együtt gördül.
A fixed értékkel a háttérképet a böngészőablakhoz rögzítjük, így a kép nem gördül a tartalommal.

A háttérkép rögzítése összevont módon

A hátteret shorthand vagyis összevont módon is megadhatjuk, ha a background tulajdonság után felsoroljuk a kívánt értékeket. Az értékek sorrendje nem kötött, tetszőleges sorrendben is megadhatjuk őket.
A shorthand megadási mód felülírja a korábbi beállításokat.
Ha a háttér méretét is szeretnénk megadni, akkor azt / jellel kell elválasztani az igazítási értékektől.



Gyakorlati feladat (Microsoft Teams 20230502_5.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/XWdwvMw
A forrásoldalon láthatod, hogy az oldal háttérképeként egy speciális kialakítású, ún. seamless kép van beszúrva. A háttérkép nem ismétlődik és a bal alsó sarokba van pozícionálva. Végezd el az alábbi háttérbeállításokat!
A háttérkép méretének értékét állítsd be a contain kulcsszóra, majd utána a cover kulcsszóra. Hasonlítsd össze a két méret megadása közötti különbséget. Végül állítsd be a háttérkép méretét 250 px nagyságúra.
A háttérkép pozícióját helyezd át a jobb felső sarokba.
A háttérkép ismétlődését módosítsd úgy, hogy először csak vízszintesen ismétlődjön, majd próbáld ki, hogyan néz ki, ha csak függőlegesen ismétlődik a háttérkép.
Végül töröld az ismétlődésért felelős CSS tulajdonság-érték párt és nézd meg a végeredményt.

Gyakorlati feladat (Microsoft Teams 20230502_6.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/oNxRKMY
A következő feladat megoldása során használd a stíluslapban előre definiált ID selectorokat.
Állítsd be az oldal háttérképét a következő szempontok alapján:

- *Az oldal háttérképeként állíts be egy neked tetsző mancsot ábrázoló ikont.*
- *Az ikon pozícióját helyezd át a jobb felső sarokba.*
- *Az ikon csak függőlegesen ismétlődjön.*
- *Az ikon szélességét állítsd méretarányosan 150 px-re.*

Már megvan a tartalom azonosítójú elem háttérképe, de még végezd el a következő beállításokat:

- *A háttérképet igazítsd a tárolóelem közepére.*
- *A háttérkép ne ismétlődjön.*
- *A háttérkép méretarányosan feszüljön a tárolóelemre.*

Már csak a címsor hátterét kell beállítanod:

- *A háttérszín legyen 40%-ban áttetsző fehér.*
- *A címsor háttérképe egyezzen meg az oldal háttérképével, azonban most ne ismétlődjön a háttérkép és a bal felső sarokba pozícionáld.*
- *A háttérkép szélességét állítsd méretarányosan 50 px-re.*

A dobozárnyékok és az elemek lebegtetése

A dobozárnyékok

A böngészőprogramban megjelenő dobozok akár árnyékot is vethetnek, ennek beállításához a box-shadow tulajdonságra van szükség.
Az első érték vízszintesen tolja el az árnyékot. Pozitív szám megadásával jobbra, negatív szám megadásával balra tolhatjuk el az árnyékot, 0 érték esetén nincs eltolás.
A második érték függőlegesen tolja el az árnyékot. Pozitív szám megadásával felfelé, negatív szám megadásával lefelé tolhatjuk el az árnyékot, 0 érték esetén nincs eltolás.
A harmadik érték az árnyék elmosódásának mértékét adja meg.
A negyedik érték az árnyék színének beállítására szolgál.
Az ötödik, opcionális érték inset névre hallgat, ezzel belső árnyékot hozhatunk létre.

Példa a dobozárnyék beállítására

```
<head>
<style>
   div {width: 200px; height: 200px;
   border: 3px solid red;
   margin: 40px;}

   .pl1 {box-shadow: -10px 20px salmon;}
   .pl2 {box-shadow: 5px 10px 5px salmon inset;}
</style>
</head>

<body>
   <div class="pl1"></div>
   <div class="pl2"></div>
</body>
```

Az elemek lebegtetése

Az elemek lebegtetésére a float tulajdonságot használhatjuk. Egy adott elemet lebegtethetünk balra (left) vagy jobbra (right), ekkor a környező elemek körbefolyják az adott elemet. A float tulajdonság alapértelmezett értéke a none, vagyis az elem nem lebeg.
Ahhoz, hogy a képek és körbefolyatott szöveg között elegendő térköz legyen, érdemes a margókat is beállítani a margin-left (bal margó) és a margin-right (jobb margó) tulajdonságokkal.

Példa a képek lebegtetésére

```
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width,
initial-scale=1.0">
   <title>Példa</title>
<style>
   #kep1 {width: 100px; float: left; margin-right: 10px;}
   #kep2 {width: 100px; float: right; margin-left: 10px;}
</style>
</head>
```

```
<body>
<p>
<img id="kep1" src="esernyok.jpg">
Lorem ipsum dolor sit amet consectetur adipisicing elit.
Itaque quia, minus autem vero culpa soluta perspiciatis,
corrupti modi eveniet, eligendi repellendus nemo expedita
hic quod quae voluptatibus dolore omnis enim. Lorem
ipsum, dolor sit amet consectetur adipisicing elit.
Reprehenderit modi excepturi laborem magnam
assumenda velit quaeret odio consectetur quia!
</p>
<p>
<img id="kep2" src="esernyok.jpg">
Lorem ipsum dolor sit amet consectetur adipisicing elit.
Itaque quia, minus autem vero culpa soluta perspiciatis,
corrupti modi eveniet, eligendi repellendus nemo expedita
hic quod quae voluptatibus dolore omnis enim. Lorem
ipsum, dolor sit amet consectetur adipisicing elit.
Exercitationem facere molestiae, eveniet debitis aliquid
cupiditate velit? Lorem ipsum dolor sit amet, consectetur
adipisicing elit.
</p>
</body>
```

A körbefolyás megszüntetése

Ha nem szeretnénk körbefolyatni a lebegtetett elemek után következő elemeket, a clear tulajdonságot kell alkalmazni:

- clear: left (az elem nem kerülhet balra lebegtetett elem mellé)
- clear: right (az elem nem kerülhet jobbra lebegtetett elem mellé)
- clear: both (az elem nem kerülhet sem balra, sem jobbra lebegtetett elem mellé)
- clear: none (az elem lebegtetett elem mellé kerül, alapértelmezett érték)

Példa a körbefolyás megszüntetésére

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Példa</title>
<style>
  #kep1 {width: 200px; float: left; margin-right: 10px;}
  #szoveg{clear:left;}
</style>
</head>

<body>
  <img id="kep1" src="Esernyok1.jpg">
  <p>Szöveg 1.</p>
  <p id="szoveg">Szöveg 2.</p>
</body>
</html>
```

Gyakorlati feladat (Microsoft Teams 20230509_1.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/JjXQGPQ
A következő feladat megoldása során a stíluslapban előre definiált azonosítókijelölőket használd a képek igazításánál.
Az első képet igazítsd balra úgy, hogy a szöveg jobb oldalról vegye körbe.
Figyelj rá, hogy a második bekezdés ne kússzon fel az első kép mellé.

- *Hozz létre egy clear nevű osztálykijelölőt.*
- *Az osztálykijelölőben állítsd be a megfelelő CSS tulajdonság-érték párral, hogy az adott elem jobb és bal oldalára ne kerüljön úsztatott elem.*
- *Alkalmazd az osztálykijelölőt a második bekezdésre.*

A második képet igazítsd jobbra úgy, hogy a szöveg balról vegye körbe.
A harmadik képet igazítsd vízszintesen középre.
A harmadik képre alkalmazd a clear osztálykijelölőt, hogy ne csússzon fel a jobbra igazított kép mellé.
Módosítsd a képekhez tartozó elemkijelölőt úgy, hogy a képek és a szöveg közötti távolság minden irányban 10 px nagyságú legyen.

Gyakorlati feladat (Microsoft Teams 20230509_2.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/JjXQGZG
A következő feladat megoldása során használd a stíluslapban előre definiált szelektorokat.
Az első képre készíts árnyékot a következő értékekkel:

- *Az árnyékot told el 2 px-lel vízszintes és függőleges irányba.*
- *Az árnyék terjedelme (elmosódottsága) legyen 5 px nagyságú.*
- *Az árnyék színe legyen fekete.*

A második képre is készíts árnyékot:

- *Az árnyékot vízszintesen told el balra 5 px-lel, függőlegesen lefelé 5 px-lel.*
- *Az árnyék elmosódottsága 10 px legyen.*
- *Az árnyék színe legyen sötétbarna (R: 71, G: 43, B: 5).*

A tárolóelemre készíts belső árnyékot:

- *Az árnyékot ne told el az elemhez képest.*
- *Az árnyék elmosódottsága 15 px legyen.*
- *Az árnyék színe legyen goldenrod.*
- *Állítsd be, hogy az árnyék vetüljön az elem belsejére.*

A stíluslapok validálása és a böngészők fejlesztői eszközei

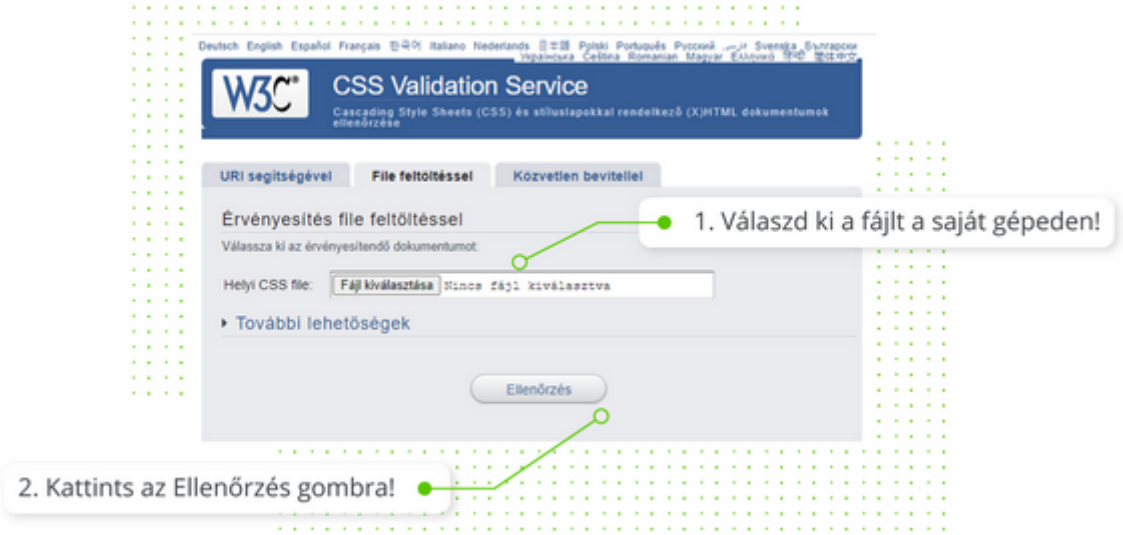
A stíluslapok validálása

A szabványosság ellenőrzése

A stíluslapok validálása ugyanúgy történik, mint a HTML-oldalaké. A szolgáltatás a W3C CSS Validation Service (https://jigsaw.w3.org/css-validator/) weboldalon érhető el.
Ha a világhálón már publikált stíluslapot szeretnénk validálni, akkor meg kell adni annak webcímét.

Érvényesítés fájl feltöltésével és közvetlen bevitellel

A saját gépünkről is feltölthetünk CSS-állományt.



Közvetlen bevitelt is alkalmazhatunk, ekkor be tudjuk másolni a CSS-kódot.

A szabványos CSS-kód

Amennyiben a CSS-kód szabványos, üzenetet kapunk, hogy sikeres a validálás, nincsenek hibák. Az is megjelenik az üzenet alatt, hogy melyik stíluslapszabvány szerint történt a validálás.

A nem szabványos CSS-kód

Ha a CSS-kód nem szabványos, a megjelenő képernyőn láthatjuk a hibák számát, a hiba leírását, a hibát tartalmazó sorok sorszámait, valamint azt is, hogy melyik stíluslapszabvány szerint történt a validálás.

A böngészőprogramok fejlesztői eszközeinek használata

Megvizsgálhatjuk velük az oldalak HTML és CSS kódját, sőt akár módosíthatjuk is. Természetesen csak a saját gépünkön.
Eltérő menüponton eltérő néven találhatók meg a különböző böngészőprogramokban.

A linkek beillesztése és formázása

A linkek beillesztése

Mi a link (hivatkozás)?

A linkekkel kapcsolatot lehet létrehozni az egyes erőforrások (különböző oldalak, médiaelemek, dokumentumok, stb.) között.

A böngészőprogramok a szöveges linkeket (nem csak szöveg lehet link) alapesetben aláhúzással és eltérő színnel jelölik.
A linkek állapotainak is különböző színe van:

- a meg nem látogatott link kék
- a már meglátogatott lila
- az aktív piros

A link állapota akkor aktív, ha a felhasználó rákattintott és folyamatban van az erőforrás betöltése. Ez az állapot sokszor fel sem tűnik, mert nagyon gyorsan töltődnek be az oldalak.
Ha például képet használunk hivatkozásként, a kurzor megváltoztatásával érdemes jelezni, hogy az adott kép az hivatkozás.

Az <a> tag használata

Linket a páros <a> taggel lehet létrehozni. Az a betű az angol anchor (horgony) szóra utal.
Szöveges hivatkozás esetén a link szövegét a nyitó és záró tagek között kell elhelyezni és a hivatkozott erőforrás webcímét a href paraméterrel kell megadni.

Példa a szöveges link készítésére

```
<a href="http://validator.w3.org/">HTML-validátor</a>
```

A link megnyitása új fülön

Előfordulhat, hogy a linket nem ugyanabban az ablakban szeretnénk betölteni, mint amelyben az aktuális oldal van. Ehhez a target paramétert kell használni _blank értékkel. Az alapértelmezett érték a _self.

Példa a külön ablakban megjelenő oldalak linkelésére

```
<a href="http://validator.w3.org/" target="_blank">HTML-
validátor</a>
```

Hivatkozás e-mailcímre

Ha e-mail címet jelenítünk meg az oldalon, akár hivatkozássá is tehetjük. Ezzel megkönnyítjük a felhasználók számára az e-mail küldést. Ha rákattintanak, a

felhasználó alapértelmezett levelezőprogramja elindul automatikusan kitöltött címzett mezővel.
E-mail esetén a href paramétert mailto: szöveggel kell kezdeni, majd ezt követi az e-mail cím.

Példa az e-mail-cím linkelésére

```
<a href="mailto:gipszjakab@gj.hu">E-mail küldése</a>
```

Hivatkozás telefonszámra

Telefonszám is alakítható linkké. A href paramétert a tel: szöveggel kell kezdeni, majd a telefonszámmal folytatni a nemzetközi előhívószámmal együtt. A linkre kattintás után az alapértelmezett hívóalkalmazás indul el, okostelefonról így egyetlen kattintással tárcsázható egy olyan szám, ami linkként szerepel egy weboldalon.

Példa a telefonszám linkelésére

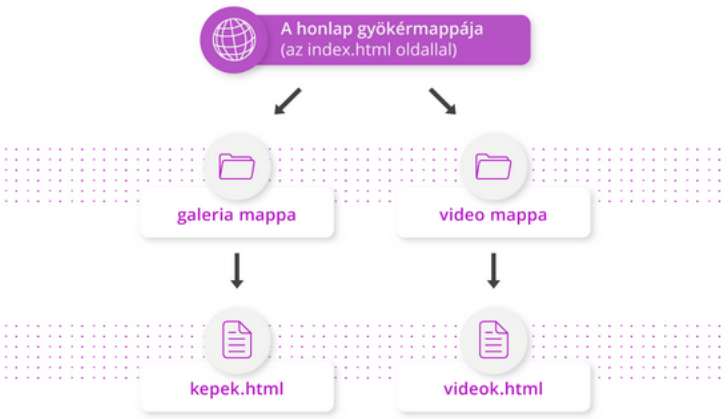
```
<a href="tel:+36209999999">Hívás indítása</a>
```

Hivatkozás a webhely oldalai között

Ha több oldalból áll az egész weboldal, meg kell oldani a navigációt, a kezdőlapról eljuthassanak a felhasználók a tartalmi oldalakra, azok között is navigálhassanak.
A holnap gyökerében a kezdőlapot érdemes index.html néven elmenteni.

Gyakorlati példa a webhelyen belüli hivatkozásokra

```
<!--Hivatkozás az index.html állományból-->

<a href="galeria/kepek.html">Képgaléria</a>
<a href="video/videok.html">Videók</a>

<!--Hivatkozás a kepek.html állományból-->

<a href="../index.html">Kezdőlap</a>
<a href="../video/videok.html">Videók</a>

<!--Hivatkozás az videok.html állományból-->

<a href="../index.html">Kezdőlap</a>
<a href="../galeria/kepek.html">Képgaléria</a>
```



A belső (oldalon belüli) hivatkozás

Ez a nagyobb tartalmi egységek tagolásánál lehet praktikus, így a felhasználó könnyen “ugrálhat” az oldalon belül.
Ilyen esetben ki kell alakítani egy csatlakozási pontot (horgonyt), ehhez kapcsolódik a hivatkozás. A csatlakozási pontot egyedi azonosítóval (id) kell ellátni. A href paraméter értéke pedig # és az azonosító.

Példa a belső hivatkozásra

A példában a kettes szintű címsor egy friss nevű egyedi azonosítóval van ellátva, majd a hivatkozás ennek segítségével van hozzárendelve.

```
<a href="#friss">Érdekel a legújabb munkám?</a>
<h2 id="friss">Legújabb munkám</h2>
```

Hivatkozás egy weben publikált honlap oldalrészére

Ehhez szükséges, hogy az adott oldalrésznek legyen azonosítója. Ha nincs, nem sokat tehetünk az ügy érdekében. Ezt az adott oldal forráskódjában ellenőrizhetjük. Ha rendelkezik ilyennel, akkor webcím után kettőskeresztet, majd az azonosítót kell megadni.

Példa egy weben publikált honlap oldalrészére történő hivatkozásra

A World Wide Webről szóló Wikipédia-szócikk forráskódjában a következő részlet olvasható:

<h2>History,

A History alcím egyedi azonosítóval van ellátva, vagyis a részre a következő hivatkozás készíthető el:

```
<a href="https://en.wikipedia.org/wiki/World_Wide_Web#History">
A világháló története (angolul)</a>
```

Magyarázat a hivatkozáshoz

A hivatkozások elláthatók rövid magyarázatokkal. Ez hasznos lehet, ha pl. kép tölti be a hivatkozás szerepét. A felhasználó alapesetben nem tudja mi történik kattintás után. A kép nagyobb változata, forrása vagy esetleg egy másik oldal jelenik meg? A title paraméterrel adható meg magyarázat.

Példa a hivatkozások magyarázattal való ellátására

```
<a href="galeria.html" title="Galéria az őszi élményekről">
<img src="https://i.ibb.co/7GJx2yz/esernyo.jpg"></a>
```

Gyakorlati feladat (Microsoft Teams 20230509_3.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/QWEjWRY
A bekezdésben található szövegből készíts egy hivatkozást.

- *A link mutasson a https://pixabay.com/hu/ URL-re.*
- *Oldd meg, hogy a hivatkozásra kattintva új böngészőfülön jelenjen meg a tartalom.*

A képre is készíts egy linket.

- *A hivatkozás egyezzen meg a kép elérési útvonalával.*
- *A hivatkozás új böngészőfülön jelenjen meg.*
- *A linkelt képhez a következő magyarázó szöveget állítsd be: "A kép megtekintése eredeti méretben"*

Gyakorlati feladat (Microsoft Teams 20230509_4.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/xxOwbqd
Az első bekezdésben sortöréssel elválasztva találod az oldalon lévő alcímek menüpontjait. Ezekből a szövegrészekből készíts három hivatkozást, amelyek az oldalon belül a megfelelő alcímekre mutatnak.
Az oldal alján található "Wikipedia" szóból készíts egy hivatkozást, ez mutasson a https://hu.wikipedia.org/wiki/Tűzhányó oldalra. A hivatkozás tartalmát jelenítsd meg új böngészőfülön.
Az utolsó bekezdésben található szövegből készíts egy linket, amelyre kattintva az oldal tetejére, a főcímre ugorhatsz vissza.

A linkek formázása stíluslappal

A pseudo classok (látszólagos osztályok) használata

Az egyes linkállapotokhoz ún. pseudo classok (látszólagos vagy álosztályok) tartoznak. Ezeket a class selectorokkal ellentétben kettőspont vezeti be:

- :link - a még meg nem nyitott link formázása állítható be
- :visited - a már meglátogatott linkek formázása állítható be
- :active - az aktív állapotban lévő link formázható vele
- :hover - beállítható, hogyan nézzen ki a link, amikor fölé viszik az egeret
- :focus - beállítható, hogyan nézzen ki a link, amikor a felhasználó tabulátorral (Tab billentyű) navigált rá a linkre

Példa a pseudo classok alkalmazására

```
<head>
<style>
   body {background-color: black;
   color: #ffffff}
   a:link {color: yellow}
   a:visited {color: orange}
   a:hover {color: white}
   a:focus {color: aqua}
   a:active {background-color: darkblue; color: lightgray}
</style>
</head>

<body>
   <p><a href="https://hu.wikipedia.org/">Wikipédia
kezdőlapja (magyar)</a></p>
   <p><a href="https://en.wikipedia.org/">Wikipédia
kezdőlapja (angol)</a></p>
   <p><a href="https://de.wikipedia.org/">Wikipédia
kezdőlapja (német)</a></p>
   <p><a href="https://fr.wikipedia.org/">Wikipédia
kezdőlapja (francia)</a></p>
   <p><a href="https://en.wikipedia.org/">Wikipédia
kezdőlapja (olasz)</a></p>
</body>
```

Az aláhúzás eltávolítása

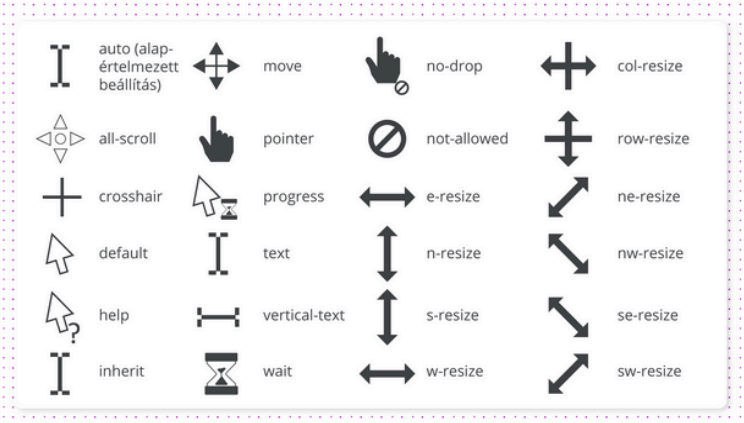
A linkek alapértelmezetten alá vannak húzva. Az aláhúzást eltávolítani a text-decoration: none segítségével lehet.

Példa a linkekhez tartozó aláhúzás eltávolítására

```
<head>
<style>
  #pelda {text-decoration: none}
</style>
</head>

<body>
  <p><a
  href="https://hu.wikipedia.org">
  Wikipédia kezdőlapja</a></p>

  <p><a id="pelda"
  href="https://hu.wikipedia.org" target="_blank">
  Wikipédia kezdőlapja</a></p>
</body>
```

A kurzor beállítása

A hivatkozás jellegének megfelelően a kurzort is be lehet állítani. A kurzor a cursor tulajdonsággal módosítható.
Ennek értéke lehet:



Példa a speciális cursor beállítására

```
<head>
<style>
  a {cursor: help}
</style>
</head>

<body>
  <a href="https://hu.wikipedia.org">
  <p>Wikipédia kezdőlapja</p></a>
</body>
```

Gyakorlati feladat (Microsoft Teams 20230509_5.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/RwRraxm
Hozz létre a stíluslapon egy stílust, amely a link alapértelmezett és látogatott állapotát formázza a következő beállításokkal:

- *A link háttere legyen kék (R: 58, G: 129, B: 187), míg betűi fehér színűek.*
- *A link szövege legyen félkövér stílusú, és ne legyen aláhúzva.*
- *A link függőleges belső margója legyen 15 px, míg a vízszintes 20 px nagyságú.*
- *Kerekítsd le a link sarkait 10 px-lel.*
- *Készíts a link köré egy 1 px vastag, folytonos, sötétkék (R: 10, G: 36, B: 75) színű szegélyt.*
- *A linkre helyezz fekete árnyékot. Az árnyék elmosódottsága 5 px legyen. Ne legyen eltolva.*

Ha a felhasználó a link fölé helyezi a kurzort, a link háttérszíne változzon világoszöldre (R: 213, G: 250, B: 145), míg betű színe egyezzen meg az alapértelmezett link háttérszínével.
A link aktív állapotához is készíts egy stílust. Ezzel állítsd be, hogy ha a felhasználó az egérrel rákattint a hivatkozásra, akkor a link fekete árnyéka vízszintes és függőleges irányba tolódjon el 2-2 px-lel, valamint az árnyék elmosódottsága ebben az esetben csak 3 px legyen.

Gyakorlati feladat (Microsoft Teams 20230509_6.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/eYzJZLy
Hozz létre a stíluslapon egy stílust, amely a link alapértelmezett és látogatott állapotát formázza a következő beállításokkal:

- *A link betűszíne legyen wheat, míg a mérete az alapértelmezetthez képest 20 százalékkal nagyobb. A szöveg sormagasságát állítsd 50 px nagyságúra.*
- *A link szövege legyen kiskapitális stílusú, és ne legyen aláhúzva.*
- *A link függőleges belső margóját 10 px nagyságúra állítsd be, a vízszintest 15 px-re.*
- *A linknek ne legyen külső margója függőleges irányba, míg a vízszintes margók nagysága 15 px legyen.*

Ha a felhasználó a link fölé helyezi a kurzort, a link tetején és alján jelenjen meg egy 3 px vastag pontozott, wheat színű szegély.
Ha egy hivatkozás fókuszba kerül, akkor cserélődjön fel a betűszín és a háttérszín, azaz a háttér legyen wheat, a betű sötétvörös színű. Az alsó és felső szegély színe szintén változzon sötétvörösre.

A listák beillesztése és formázása

A listák beillesztése

A listákról általában

A listák alkalmazása nagy segítséget jelent, ha a közölni kívánt információkat tagoltan és könnyen átláthatóan kívánjuk megjeleníteni.

A felsoroláslista

Ebben az esetben a listaelemek elé egy-egy listajel kerül, ez alapértelmezetten egy fekete kör.
A felsoroláslistákat a páros <ul> taggel hozhatjuk létre, az egyes listaelemeket a szintén páros <li> tagek közé kell tenni.

Példa a felsoroláslisták készítésére

```
<p>A táborba hozd magaddal a következőket:</p>
<ul>
  <li>hálózsák</li>
  <li>zseblámpa</li>
  <li>fürdőruha</li>
</ul>
```

A sorszámozott lista - növekvő számozás

Ebben az esetben az <ol> taget kell használni. Az elemek így kapnak egy automatikus sorszámozást.

A kezdőérték beállítása az ol tag start paraméterével lehetséges.

Példa a sorszámozott listák készítésére

```
<p>A verseny állása:</p>
<ol>
  <li>Nagy Gábor</li>
  <li>Németh Tamás</li>
  <li>Tóth Éva</li>
</ol>

<p>A verseny további helyezettjei:</p>
<ol start="4">
  <li>Kis Ágota</li>
  <li>Lenkei Zsolt</li>
  <li>Molnár Tamás</li>
  <li>Hegyi Rita</li>
</ol>
```

A sorszámozott lista - csökkenő sorszámozás

Az ol taget a reversed paraméterrel kell ellátni a csökkenő sorszámozáshoz.

Példa a csökkenő sorszámozásra

```
<p>A legnépszerűbb kutyafajták</p>
<ol reversed>
  <li>beagle</li>
  <li>bulldog</li>
  <li>golden retriever</li>
  <li>németjuhász</li>
  <li>labrador</li>
</ol>
```

A sorszámozott lista - a sorszámtípus beállítása

Alapesetben a sorszámozás tízes számrendszerbeli, arab számokkal történik.
Ennek módosításához az <ol> tag type paramétere használható.
A paraméter értékei lehetnek:

- 1 - tízes számrendszerbeli számok
- a - a latin ábécé kisbetűi

- A - a latin ábécé nagybetűi
- i - római számok kisbetűvel
- I - római számok nagybetűvel

Példa a különböző sorszámtípusok beállítására

```
<ol type="1">
  <li>listaelem</li>
  <li>listaelem</li>
  <li>listaelem</li>
</ol>
<ol type="a">
  <li>listaelem</li>
  <li>listaelem</li>
  <li>listaelem</li>
</ol>
<ol type="A">
  <li>listaelem</li>
  <li>listaelem</li>
  <li>listaelem</li>
</ol>
<ol type="i">
  <li>listaelem</li>
  <li>listaelem</li>
  <li>listaelem</li>
</ol>
<ol type="I">
  <li>listaelem</li>
  <li>listaelem</li>
  <li>listaelem</li>
</ol>
```

A definíciós lista

Ha fogalmakat és hozzájuk fűzött magyarázatokat közölnénk a weboldalunkon, akkor egy speciális listát, az úgynevezett definíciós listát érdemes használnunk. Ebben az esetben a teljes listát a páros <dl>, a fogalmakat a páros <dt>, a magyarázatokat a páros <dd> elemek közé kell elhelyezni.

Példa a definíciós listák készítésére

```
<dl>
  <dt>HTML</dt>
  <dd>HyperText Markup Language:
  Hiperszöveg-jelölő nyelv</dd>
  <dt>CSS</dt>
  <dd>Cascading Style Sheets:
  Rangsorolt stíluslapok</dd>
</dl>
```

Az egymásba ágyazott listák

Egy listaelem akár további listákat is tartalmazhat, vagyis lehetőség van többszintű listák kialakítására is. A beágyazott listákat az egyes listaelemekhez kell rendelni.

Példa a listák egymásba ágyazására

```
<p>Kedvelt reggeli italok:</p>
<ol>
  <li>gyümölcslé
    <ul>
      <li>narancslé</li>
      <li>almalé</li>
    </ul>
  </li>
  <li>tea
    <ul>
      <li>fekete tea</li>
      <li>gyümölcstea</li>
    </ul>
  </li>
</ol>
```

Gyakorlati feladat (Microsoft Teams 20230516_1.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/NWrNmXB
Készíts egy hét elemből álló felsoroláslistát a főcím alatt látható szövegből.

Gyakorlati feladat (Microsoft Teams 20230516_2.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/mdEPgQq
Készíts egy hét elemből álló sorszámozott listát a főcím alatt látható szövegből.

Gyakorlati feladat (Microsoft Teams 20230516_3.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/ZEOWZPz
Készíts egy öt fogalomból (Folyó, Légkör, Óceán, Tó, Természet) álló definíciós listát a főcím alatt látható szövegből.

Gyakorlati feladat (Microsoft Teams 20230516_4.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/zYBqXVr
Készíts egy kétszintű listát a főcím alatt látható szövegből.
Az első szintű listaelemek legyenek a következők: Az ókori világ hét csodája, A modern idők listája, Turistalátvány-csodák. Ezek az elemek alkossanak felsoroláslistát.
A felsoroláslista mindhárom elemébe ágyazz be egy-egy hét elemből álló sorszámozott listát.

A listák formázása

A listajelölő beállítása

A listák elemei előtt megjelenő listajelölők megjelenését stíluslapokkal formázni is lehet.
Ehhez a deklarációs blokkban a list-style-type tulajdonságot kell megadni. Ennek értékei a következő kulcsszavak lehetnek:

- disc (kitöltött kör)
- circle (üres kör)
- square (négyzet)
- decimal (arab számok)
- decimal-leading-zero (arab számok vezető nullával, 01, 02, 03, stb)
- upper-roman (nagybetűs római számok)
- lower-roman (kisbetűs római számok)
- upper-alpha (nagybetűs latin abc)
- lower-alpha (kisbetűs latin abc)
- none (nincs megjelenő listajel)

Példa a listajelölők formázására

```
<head>
<style>
   #pelda1 {list-style-type: circle}
   #pelda2 {list-style-type: square}
   #pelda3 {list-style-type: decimal-leading-zero}
   #pelda4 {list-style-type: upper-roman}
   #pelda5 {list-style-type: lower-alpha}
   #pelda6 {list-style-type: none}
</style>
</head>

<body>
   <ul id="pelda1"><li>listaelem</li><li>listaelem</li>
<li>listaelem</li></ul>
   <ul id="pelda2"><li>listaelem</li><li>listaelem</li>
<li>listaelem</li></ul>
   <ul id="pelda3"><li>listaelem</li><li>listaelem</li>
<li>listaelem</li></ul>
   <ul id="pelda4"><li>listaelem</li><li>listaelem</li>
<li>listaelem</li></ul>
   <ul id="pelda5"><li>listaelem</li><li>listaelem</li>
<li>listaelem</li></ul>
   <ul id="pelda6"><li>listaelem</li><li>listaelem</li>
<li>listaelem</li></ul>
</body>
```

A listajelölő pozicionálása

A stíluslapokkal a listajelölő pozíciója is módosítható. A deklarációs blokkon belül ehhez a list-style-position tulajdonságot szükséges használni, értékei a következő kulcsszavak lehetnek:

- outside - a listajel a listaelem dobozán kívülre kerül (ez az alapértelmezett érték)
- inside - a listajel a listaelem dobozán belülre kerül

Példa a listajelölők pozicionálására

```
<head>
<style>
  li {border: 1px dotted gray}
  #li1 {list-style-position: outside}
  #li2 {list-style-position: inside}
</style>
</head>

<body>
  <ul id="li1"><li>listaelem</li><li>listaelem</li>
<li>listaelem</li></ul>
  <ul id="li2"><li>listaelem</li><li>listaelem</li>
<li>listaelem</li></ul>
</body>
```

Képek használata listajelölőként

A szimbólumok, betűk és számok mellett lehetőség van arra is, hogy kép (ikon) jelenjen meg listajelölőként. A használni kívánt kép elérhetőségét a list-style-image tulajdonsággal adhatjuk meg.
A kép alapesetben fizikai méretben jelenik meg, ezért általában átméretezésre van szükség.

Példa a képek listajelölőként való alkalmazására

```
<head>
<style>
  ul {list-style-image: url('http://lorempixel.com/30/30/city')}
</style>
</head>

<body>
<ul>
  <li>listaelem</li>
  <li>listaelem</li>
  <li>listaelem</li>
</ul>
</body>
```

A tulajdonságok tömör megadása

A formázások a list-style tulajdonsággal akár tömörebben is megadhatók. Ebben az esetben értékként az előzőekben bemutatott kulcsszavakat szükséges feltüntetni egymás után, szóközökkel elválasztva.

Példa a listatulajdonságok tömör megadására

```
<head>
<style>
   ul {border: 1px dotted gray}
   #li1 {list-style: inside square}
   #li2 {list-style: url('http://lorempixel.com/30/30/city')
outside}
</style>
</head>

<body>
<ul id="li1">
  <li>listaelem</li>
  <li>listaelem</li>
  <li>listaelem</li>
</ul>
<ul id="li2">
  <li>listaelem</li>
  <li>listaelem</li>
  <li>listaelem</li>
</ul>
</body>
```

Gyakorlati feladat (Microsoft Teams 20230516_5.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/ZEOOzmL
Készíts egy kulso nevű class selectort, és segítségével állítsd be a következő formázásokat az első szintű listára.

- *A háttérszín legyen yellowgreen, a lista szélessége 400 px.*
- *A lista belső margóját állítsd be vízszintes irányban 30 px, míg függőlegesen 10 px nagyságúra.*
- *A listajel típusának csillagokat állíts be, amelyek kódja '\2728'.*
- *A lista szövege legyen kiskapitális, félkövér stílusú.*

Készíts egy belső nevű class selectort, és segítségével formázd a második szintű listát.

- *A háttérszín legyen világoszöld (R: 228, G: 247, B: 191).*
- *Állítsd be, hogy a második szintű listaelemek szövege ne legyen félkövér.*

Gyakorlati feladat (Microsoft Teams 20230516_6.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/ZEOOzNK
A definíciós lista minden HTML-eleméhez készíts egy-egy type selectort, és segítségükkel állítsd be a következő formázásokat:

- *A definíciós lista háttérszíne legyen #ddd szürke színű.*
- *A lista belső margóját függőleges irányban 5 px, vízszintesen 10 px nagyságúra állítsd be.*
- *A lista bal oldalán jelenjen meg egy 3 px vastag, dupla fekete szegély.*
- *A definíciós lista fogalmai legyenek nagybetűsek és félkövér stílusúak.*
- *A fogalmak felső margója legyen 15 px, míg alsó margója 5 px nagyságú.*
- *A fogalmak magyarázó szövegei jelenjenek meg dőlt stílussal.*

Az egyszerű táblázatok beillesztése és formázása

Az egyszerű táblázatok beillesztése

A táblázatokról általában

A táblázatok használatával az adatokat átláthatóan, sorokba és oszlopokba rendezve jeleníthetjük meg.
A táblázatok cellái alapvetően két csoportba sorolhatók:

- adatcella: ide kerülnek a tényleges információk, adatok
- fejléccella: az ide kerülő tartalmak határozzák meg a táblázat oszlopaiba és soraiba kerülő adatok jellegét

A táblázatoknak címet is adhatunk.

A táblázatok beillesztése

A táblázatokat minden esetben a páros <table> tagek közé kell helyezni.
Ezután következik a táblázat címe, amit a páros <caption> tagekkel lehet megadni.
A táblázat sorai a páros <tr> tagekkel jelölhetők ki.

Ha a táblázat első sorában fejléccellákat szeretnénk elhelyezni, akkor a <tr> tagekbe páros <th> tageket kell ágyazni. Ezekből annyit kell használni, ahány oszlopból álló táblázatot szeretnénk készíteni.
A páros <td> tagekkel kell ellátni azokat a sorokat, amelyekben adatcellákat szeretnénk kialakítani.

Példa a táblázatok létrehozására

```
<table>
<caption>A szerencsés nyertesek</caption>
<tr>
  <th>Név</th>
  <th>Lakóhely</th>
</tr>
<tr>
  <td>Kiss Ágnes</td>
  <td>Budapest</td>
</tr>
<tr>
  <td>Kővágó László</td>
  <td>Kecskemét</td>
</tr>
<tr>
  <td>Nagy Tibor</td>
  <td>Debrecen</td>
</tr>
</table>
```

A fejléccellák akadálymentes megadása

A képernyőolvasó programot használók számára a táblázatot érthetőbbé tehetjük, ha a fejléccellák kódjában jelezzük, hogy az adott fejléc sorra vagy oszlopra vonatkozik. Ilyenkor a képernyőolvasó programok a cellák szövegét a hozzájuk tartozó fejléc szövegével együtt olvassák fel. Ehhez a scope paramétert kell használni, ennek értéke oszlopra vonatkozó fejléccella esetén a col, sorra vonatkozó esetén a row.

Példa a fejléccellák akadálymentes megadására

```
<table>
<caption>A szerencsés nyertesek</caption>
<tr>
  <th scope="col">Név</th>
  <th scope="col">Lakóhely</th>
</tr>
<tr>
  <th>Kiss Ágnes</th>
  <th>Budapest</th>
</tr>
</table>
```

A táblázatok beillesztése a Visual Studio Code alkalmazásban

Mivel a táblázatok készítése alapvetően sok gépeléssel jár, a VS Code alkalmazás lehetőséget nyújt arra, hogy Emmet-parancsokkal egyszerűbben is létre lehessen hozni őket.
Az egymásba ágyazott, táblázatot alkotó tageket elegendő balra mutató, “nagyobb” relációs jelekkel (>) elválasztani egymástól. Például egy egyetlen cellából álló táblázat így rövidíthető le: <table><tr><td></td></tr></table> -> table>tr>td
Ha egy tagből többre lenne szükség, akkor a sokszorozni kívánt tag neve után egy * jellel állítható be a kívánt darabszám.

Példa a táblázatok Emmet-parancsokkal való megadására

```
table>tr*2>td*3

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

Gyakorlati feladat (Microsoft Teams 20230516_7.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/yLJJOWV

A főcím és a bekezdés alatt látható szövegből készíts egy 11 soros, 5 oszlopos táblázatot.
A táblázat első sora a fejléccellákat - “Nemzet (NOB-kód)”, “Nyári olimpiai részvételek száma”, “Arany”, “Ezüst”, “Bronz” - tartalmazza.
A további sorokban a sorfejlécek tartalmazzák a nemzetek nevét és azok NOB-kódját. A többi cella adatcella legyen, amelyek a részvételek számát és az érmek számát tartalmazzák.
Az első sor fejléccelláin jelöld, hogy oszlopra, míg a sorok első celláin, hogy sorra vonatkoznak.
A táblázat címe legyen “Nyári olimpiai játékok összesített adatai”.

A táblázatok formázása stíluslappal

A táblázatok szegélyezése

A táblázatokat a border tulajdonság használatával lehet szegélyezni. Ha ezt a tulajdonságot a <table> elemre helyezzük, akkor az egész táblázat külső szegélyt kap. Belső szegély létrehozásához a formázást a táblázat celláira is be kell állítani.

Példa a táblázatok külső szegélyezésére

```
<head>
<style>
  table {border: 1px solid black}
</style>
</head>

<body>
<table>
  <caption>A szerencsés nyertesek</caption>
  <tr>
    <th>Név</th>
    <th>Lakóhely</th>
  </tr>
  <tr>
    <td>Kiss Ágnes</td>
    <td>Budapest</td>
  </tr>
  <tr>
    <td>Kővágó László</td>
    <td>Kecskemét</td>
  </tr>
  <tr>
    <td>Nagy Tibor</td>
    <td>Debrecen</td>
  </tr>
</table>
</body>
```

Példa a táblázatok külső és belső szegélyezésére

```
<head>
<style>
  table, th, td {border: 1px solid black}
</style>
</head>

<body>
<table>
  <caption>A szerencsés nyertesek</caption>
  <tr>
    <th>Név</th>
    <th>Lakóhely</th>
  </tr>
  <tr>
    <td>Kiss Ágnes</td>
    <td>Budapest</td>
  </tr>
  <tr>
    <td>Kővágó László</td>
    <td>Kecskemét</td>
  </tr>
  <tr>
    <td>Nagy Tibor</td>
    <td>Debrecen</td>
  </tr>
</table>
</body>
```

A szegélyek közötti távolság beállítása

A cellaszegélyek közötti távolságot a deklarációs blokkban a border-spacing tulajdonsággal állíthatjuk be. Ha egy hosszúságértéket adunk meg, akkor az érték érvényesül mind vízszintes, mind függőleges irányban. Két megadott érték esetén az első érték a vízszintes, a második érték a függőleges távolság nagyságát határozza meg.

Példa a cellaszegélyek közötti távolság beállítására

```
<head>
<style>
  table, th, td {border: 1px solid black}
  table {border-spacing: 10px 20px}
</style>
</head>

<body>
<table>
  <caption>A szerencsés nyertesek</caption>
  <tr>
    <th>Név</th>
    <th>Lakóhely</th>
  </tr>
  <tr>
    <td>Kiss Ágnes</td>
    <td>Budapest</td>
  </tr>
  <tr>
    <td>Kővágó László</td>
    <td>Kecskemét</td>
  </tr>
  <tr>
    <td>Nagy Tibor</td>
    <td>Debrecen</td>
  </tr>
</table>
</body>
```

A szegélyek összevonása

Ha a szegélyek közötti távolságot 0 px-re állítjuk, akkor az egymás mellé kerülő szegélyek vastagsága összeadódik. Ha a szomszédos szegélyeket össze szeretnénk vonni, akkor a deklarációs blokkban a border-collapse tulajdonság értékét collapse kulcsszóval kell megadni. A tulajdonság alapértelmezett értéke separate.

Példa a táblázatban lévő szegélyek egymás mellé kerülésére

```
<head>
<style>
  table, th, td {border: 3px solid black}
  table {border-spacing: 0 10px}
</style>
</head>

<body>
<table>
  <caption>A szerencsés nyertesek</caption>
  <tr>
    <th>Név</th>
    <th>Lakóhely</th>
  </tr>
  <tr>
    <td>Kiss Ágnes</td>
    <td>Budapest</td>
  </tr>
  <tr>
    <td>Kővágó László</td>
    <td>Kecskemét</td>
  </tr>
  <tr>
    <td>Nagy Tibor</td>
    <td>Debrecen</td>
  </tr>
</table>
</body>
```

Példa a táblázatban egymás mellett lévő szegélyek összevonására

```
<head>
<style>
table, th, td {border: 3px solid black}
table {border-spacing: 0 10px;
border-collapse: collapse}
</style>
</head>

<body>
<table>
  <caption>A szerencsés nyertesek</caption>
  <tr>
    <th>Név</th>
    <th>Lakóhely</th>
  </tr>
  <tr>
    <td>Kiss Ágnes</td>
    <td>Budapest</td>
  </tr>
  <tr>
    <td>Kővágó László</td>
    <td>Kecskemét</td>
  </tr>
  <tr>
    <td>Nagy Tibor</td>
    <td>Debrecen</td>
  </tr>
</table>
</body>
```

A belső margó beállítása

A táblázat celláinál érdemes belső margót alkalmazni, máskülönben a szövegek túl közel kerülnek a szegélyekhez. Ehhez a deklarációs blokkban a padding tulajdonságot kell használni.
A szegélybeállításokhoz hasonlóan a belső margót nem csupán a <table>, hanem a <th> és a <td> elemekre is be kell állítani.

Példa a táblázatok belső margójának beállítására

```
<head>
<style>
  table, th, td {border: 3px solid black;
  padding: 10px}
  table {border-collapse: collapse;}
</style>
</head>

<body>
<table>
  <caption>A szerencsés nyertesek</caption>
  <tr>
    <th>Név</th>
    <th>Lakóhely</th>
  </tr>
  <tr>
    <td>Kiss Ágnes</td>
    <td>Budapest</td>
  </tr>
  <tr>
    <td>Kővágó László</td>
    <td>Kecskemét</td>
  </tr>
  <tr>
    <td>Nagy Tibor</td>
    <td>Debrecen</td>
  </tr>
</table>
</body>
```

A táblázatcím helyzete

A táblázatok címe alapértelmezett esetben a táblázatok felett jelenik meg. Ezt a pozíciót a caption-side tulajdonsággal bírálhatjuk felül.
Ha a tulajdonság értékeként a bottom kulcsszót állítjuk be, akkor a cím a táblázat alatt jelenik meg. Az alapértelmezett helyzetet a top kulcsszóval állíthatjuk vissza.

Példa a táblázatcím helyzetének módosítására

```
<head>
<style>
  table, th, td {border: 3px solid black;
  padding: 10px}
  table {border-collapse: collapse;
 caption-side: bottom}
</style>
</head>

<body>
<table>
  <caption>A szerencsés nyertesek</caption>
  <tr>
    <th>Név</th>
    <th>Lakóhely</th>
  </tr>
  <tr>
    <td>Kiss Ágnes</td>
    <td>Budapest</td>
  </tr>
  <tr>
    <td>Kővágó László</td>
    <td>Kecskemét</td>
  </tr>
  <tr>
    <td>Nagy Tibor</td>
    <td>Debrecen</td>
  </tr>
</table>
</body>
```

Gyakorlati feladat (Microsoft Teams 20230523_1.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/YzWWWPB
Vedd körbe a táblázatot egy 2px vastag folytonos fekete szegéllyel. A táblázat fejléc- és adatcelláira alul és felül készíts egy 2 px vastag pontozott stílusú, szürke (grey) színű szegélyt. Gondoskodj róla, hogy a táblázat és a cellák szegélyei egyesítve jelenjenek meg.
A táblázat címe legyen a táblázat alatt. A cím sötétkék (darkblue) színű és dőlt stílusú legyen. A cím tetejére helyezz el egy 6 px nagyságú külső margót.
A táblázatot igazítsd vízszintesen középre.
A táblázat belső magója legyen 5 px nagyságú.
A fejlécek háttérszíne legyen világosszürke (lightgrey).
Az adatcellák szövegét igazítsd vízszintesen középre.

Az összetett táblázatok beillesztése és formázása

Az összetett táblázatok beillesztése

A cellák összevonása

A táblázatot alkotó, szomszédos adat-, illetve fejléccellákat szükség esetén függőlegesen és vízszintesen is összevonhatjuk.
A cellák vízszintes összevonása technikailag oszlopok összevonását jelenti. Ehhez az első összevonni kívánt cellát jelölő taget a colspan paraméterrel kell ellátni. A kód neve az angol column és span (oszlop és összevonás) szavak rövidítése. A paraméter értékeként az összevonni kívánt cellák számát szükséges megadni. Az összevonásban résztvevő cellákat ezután nem kell megadni.
A függőleges összevonással sorokat egyesíthetünk. Ez az oszlopok összevonásához hasonlóan működik azzal a különbséggel, hogy itt a rowspan (row - sor) paramétert kell használni.
Akár egymással is kombinálhatjuk a colspan és rowspan paramétereket, azaz egy cellára egyszerre alkalmazhatunk függőleges és vízszintes összevonást is.

Példák a táblázatcellák függőleges, illetve vízszintes összevonására

```
<head>
<style>
   table, tr, th, td {border: 1px solid black;
   padding: 5px}
   table {border-collapse: collapse}
</style>
</head>
```

```
<body>
<table>
  <caption>A szerencsés nyertesek</caption>
  <tr>
    <th colspan="3" scope="col">Név</th>
    <th scope="col">Lakóhely</th>
  </tr>
  <tr>
    <th rowspan="4">Érmet szerzett</th>
    <td>Vezetéknév</td>
    <td>Keresztnév</td>
    <td></td>
  </tr>
  <tr>
    <td>Kiss</td>
    <td>Ágnes</td>
    <td>Budapest</td>
  </tr>
  <tr>
    <td>Kővágó</td>
    <td>László</td>
    <td>Kecskemét</td>
  </tr>
  <tr>
    <td>Nagy</td>
    <td>Tibor</td>
    <td>Debrecen</td>
  </tr>
</table>
</body>
```

Példák a táblázatcellák függőleges, illetve vízszintes összevonására

```
<head>
<style>
  table, tr, th, td {border: 1px solid black;
  padding: 5px}
  table {border-collapse: collapse}
</style>
</head>
```

```
<body>
<table>
  <caption>A verseny eredménye</caption>
  <tr>
    <th scope="col">Név</th>
    <th scope="col">Város</th>
    <th scope="col">Érem</th>
  </tr>
  <tr>
    <td>Kovács István</td>
    <td>Budapest</td>
    <td>arany</td>
  </tr>
  <tr>
    <td colspan="2" rowspan="2">
      <em>A zsűri idén nem osztott ki ezüst- és
bronzérmet</em>
    </td>
    <td>ezüst</td>
  </tr>
  <tr>
    <td>bronz</td>
  </tr>
</table>
</body>
```

A táblázat fej- és lábléce, illetve törzse

Az összetettebb táblázatok készítésekor érdemes tartalmi elemeket (fejléc, törzs, lábléc) kialakítani a táblázaton belül.
Ennek előnye elsősorban a hosszabb táblázatok nyomtatásakor tapasztalható, mivel a beállított fej- és láblécek valamennyi nyomtatott oldalra rákerülnek.
A fejlécet alkotó táblázatsorokat a páros <thead>, a táblázat törzsét a páros <tbody>, a láblécnek szánt sorokat a páros <tfoot> tagekbe kell beágyazni.
A láblécet az adatok forrásának jelölésére érdemes fenntartani.


```
<table>
  <thead>
  <tr>
    <th scope="col">Évszám</th>
    <th scope="col">Esemény</th>
    <th scope="col">Rendező város</th>
    <th scope="col">Rendező ország</th>
  </tr>
  </thead>
  <tbody>
  <tr>
    <td>1896</td>
    <td>I. nyári olimpiai játékok</td>
    <td>Athén</td>
    <td>Görögország</td>
  </tr>

  <!--További adatsorok-->

  </tbody>
  <tfoot>
  <tr>
    <td colspan="4">Forrás:
      <a href="#">Wikipedia</a>
    </td>
  </tr>
  </tfoot>
</table>
```

A tagek sorrendje a HTML5-szabványban

A hosszabb táblázatok esetén sem kötelező mindhárom tartalmi egységet megadni. Készíthetünk olyan táblázatot, amelynek nincs fej- vagy lábléce. Arra azonban

mindenképp figyelni kell, hogy a tartalmi elemeket csak meghatározott sorrendben lehet felhasználni a szerkesztés során.
A HTML5-szabványban a következő szabályokat kell betartani:

- <thead> - meg kell előznie a <tbody>, a <tfoot> és a <tr> tageket, míg előtte csak a <caption> tag helyezkedhet el
- <tbody> - a <caption>, illetve a <thead> tagek után lehet elhelyezni
- <tfoot> - a <caption>, a <thead>, illetve a <tbody> tagek után helyezkedhet el

Gyakorlati feladat (Microsoft Teams 20230523_2.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/QWEVYGb
Vond össze vízszintesen a táblázat első sorának celláit.
A "Más nevek" feliratot tartalmazó cellát függőleges irányban egyesítsd az alatta lévő két cellával.
Vond össze vízszintesen a táblázat hatodik sorának két celláját.
A "Megjelenés" feliratot tartalmazó cellát egyesítsd az alatta lévő üres cellával.

A táblázatok formázása

Az első gyermekelemek formázása

A :first-child pseudo class (látszólagos osztály) segítségével egy meghatározott elem első gyermekelemére állíthatunk be külön formázást.
Táblázat esetén a <table> tag első gyermekeleme a táblázat sorai, azaz a <tr> tagek. Ennek megfelelően a táblázat első sora a tr:first-child selector használatával formázható.
Ha a táblázat első fejléc- vagy adatcelláját jelöljük ki, akkor a táblázat első oszlopára alkalmazhatunk egységes beállításokat.

Példa a táblázat első sorának formázására

```
<head>
<style>
  table, tr, th, td {border: 1px solid black;
  padding: 5px}
  table {border-collapse: collapse}
  tr:first-child {background-color: gold}
</style>
</head>
```

```
<body>
<table>
  <caption>A szerencsés nyertesek</caption>
  <tr>
    <th>Vezetéknév</th>
    <th>Keresztnév</th>
    <th>Lakóhely</th>
  </tr>
  <tr>
    <td>Kiss</td>
    <td>Ágnes</td>
    <td>Budapest</td>
  </tr>
  <tr>
    <td>Kővágó</td>
    <td>László</td>
    <td>Kecskemét</td>
  </tr>
  <tr>
    <td>Nagy</td>
    <td>Tibor</td>
    <td>Debrecen</td>
  </tr>
</table>
</body>
```

Példa a táblázat első oszlopának formázására

```
<head>
<style>
  table, tr, th, td {border: 1px solid black;
  padding: 5px}
  table {border-collapse: collapse}
  td:first-child {background-color: gold}
</style>
</head>
```

```
<body>
<table>
  <caption>A szerencsés nyertesek</caption>
  <tr>
    <th>Vezetéknév</th>
    <th>Keresztnév</th>
    <th>Lakóhely</th>
  </tr>
  <tr>
    <td>Kiss</td>
    <td>Ágnes</td>
    <td>Budapest</td>
  </tr>
  <tr>
    <td>Kővágó</td>
    <td>László</td>
    <td>Kecskemét</td>
  </tr>
  <tr>
    <td>Nagy</td>
    <td>Tibor</td>
    <td>Debrecen</td>
  </tr>
</table>
</body>
```

Az utolsó gyermekelemek formázása

Az utolsó gyermekelemeket, azaz jelen esetben a táblázatok záró sorait és oszlopait szintén egy pseudo class-szal, a :last-child segítségével formázhatjuk egységesen.

Példa a táblázat utolsó sorának és oszlopának formázására

```
<head>
<style>
  table, tr, th, td {border: 1px solid black;
  padding: 5px}
  table {border-collapse: collapse}
  tr:last-child {background-color: lightblue}
  td:last-child {background-color: lightblue}
</style>
</head>
```

```
<body>
<table>
  <caption>A verseny eredménye</caption>
  <tr>
    <th>Név</th>
    <th>Város</th>
    <th>Érem</th>
  </tr>
  <tr>
    <td>Kovács István</td>
    <td>Budapest</td>
    <td>arany</td>
  </tr>
  <tr>
    <td>Nagy Ábel</td>
    <td>Monor</td>
    <td>ezüst</td>
  </tr>
  <tr>
    <td>Hevesi Ákos</td>
    <td>Gödöllő</td>
    <td>bronz</td>
  </tr>
  <tr>
    <td colspan="2">Gratulálunk a nyerteseknek!</td>
    <td></td>
  </tr>
</table>
</body>
```

A tetszőleges gyermekelemek formázása

Az :nth-child pseudo class segítségével lehetőségünk van tetszőleges gyermekelem kijelölésére, vagyis így a táblázat bármelyik sorát vagy oszlopát egységesen formázhatjuk. Ebben az esetben az :nth-child kulcsszavak után a kijelölni kívánt gyermekelem kódolásban elfoglalt sorszámát kell zárójelbe írni.
Ha a zárójelben az odd kulcsszót tüntetjük fel, akkor valamennyi páratlan elemre készíthetünk stílusbeállítást. Even kulcsszó használatával a páros sorszámú gyermekelemeket formázhatjuk egységesen.
A kulcsszavak és a konkrét értékek mellett különféle matematikai formulákat is megadhatunk a zárójelben. A 2n+1 formula segítségével szintén páratlan sorszámú elemekre hozhatunk létre kijelölést. A 3n+2 formula ennek megfelelően a 2, 5, 8, 11, stb. sorszámú elemek formázását teszi lehetővé.

Példa a tetszőleges táblázatsorok és -oszlopok formázására

```
<head>
<style>
  table, tr, th, td {border: 1px solid black; padding: 20px}
  table {border-collapse: collapse}
  tr:nth-child(9) {background-color: lightblue}
  tr:nth-child(even) {background-color: gold}
  tr:nth-child(6n+1) {background-color: salmon}
</style>
</head>
```

```
<body>
<table>
  <tr>
    <td></td><td></td><td></td><td></td>
  </tr>
  <tr>
    <td></td><td></td><td></td><td></td>
  </tr>
  <tr>
    <td></td><td></td><td></td><td></td>
  </tr>
  <tr>
    <td></td><td></td><td></td><td></td>
  </tr>
  <tr>
    <td></td><td></td><td></td><td></td>
  </tr>
  <tr>
    <td></td><td></td><td></td><td></td>
  </tr>
  <tr>
    <td></td><td></td><td></td><td></td>
  </tr>
  <tr>
    <td></td><td></td><td></td><td></td>
  </tr>
  <tr>
    <td></td><td></td><td></td><td></td>
  </tr>
  <tr>
    <td></td><td></td><td></td><td></td>
  </tr>
</table>
</body>
```

A cellatartalom függőleges igazítása

A cellák tartalmát a vertical-align tulajdonság alkalmazásával igazíthatjuk függőlegesen. A tulajdonság értékeként a következő kulcsszavak adhatók meg:

- top - felülre igazítás
- middle - középre igazítás (alapértelmezett érték)
- bottom - alulra igazítás

Példa a cellatartalom függőleges igazítására

```
<head>
<style>
  table, tr, th, td {border: 1px solid black; padding: 20px}
  table {border-collapse: collapse}
  #igazit1 {vertical-align: top}
  #igazit2 {vertical-align: bottom}
</style>
</head>
```

```
<body>
<table>
  <caption>Példa 1.</caption>
  <tr>
    <td rowspan="5" id="igazit1">Aktuális érték</td>
    <td>1</td>
  </tr>
  <tr><td>2</td></tr>
  <tr><td>3</td></tr>
  <tr><td>4</td></tr>
  <tr><td>5</td></tr>
</table>

<table>
  <caption>Példa 2.</caption>
  <tr>
    <td rowspan="5" id="igazit2">Aktuális érték</td>
    <td>1</td>
  </tr>
  <tr><td>2</td></tr>
  <tr><td>3</td></tr>
  <tr><td>4</td></tr>
  <tr><td>5</td></tr>
</table>
</body>
```

Gyakorlati feladat (Microsoft Teams 20230523_3.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/JjKaqEp
A táblázat első celláját vond össze függőlegesen a második sor első cellájával.
Alakítsd ki a táblázat fejlécét, törzsét és láblécét.

- *A táblázat első két sora legyen a táblázat fejléce.*
- *Az utolsó sor legyen a táblázat lábléce.*
- *A gyümölcsökre vonatkozó információkat a táblázat törzsében helyezd el.*

Gyakorlati feladat (Microsoft Teams 20230523_4.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/VwjGOdq
A táblázatot és celláit vedd körbe egy 1 px vastag folytonos szegéllyel. A szegély színe legyen fekete. Figyelj rá, hogy a szegély ne duplázódjon.
A cellák tartalmát igazítsd vízszintesen és függőlegesen is középre.
A táblázat cellamargója legyen 5 px nagyságú.
A táblázat felirata a táblázat alatt jelenjen meg. A felirat legyen dőlt stílusú. A felirat és a táblázat között állíts be 12 px nagyságú margót.
A további feladatokat a tanult pseudo classok felhasználásával oldd meg!
A táblázat első sorának háttérszíne legyen kék (R: 14, G: 83, B: 129), míg betűszíne fehér.
Az első oszlop háttérszíne legyen szürkéskék (R: 191, G: 212, B: 226). Az első oszlop szöveges tartalma legyen félkövér és dőlt stílusú.
A táblázat páros sorainak háttérszínét állítsd be világos szürkéskék színűre (R: 235, G: 242, B: 247).
A táblázat utolsó sorának háttérszínét állítsd be szürke színűre (R: 179, G: 182, B: 184), a cella tartalma pedig legyen félkövér stílusú.

A HTML5-oldalszerkezet elemei és a pozicionálási sémák

A HTML5-oldalszerkezet elemei

Az oldalszerkezet megadása

A HTML5-nyelvben az oldal a következő szerkezeti egységekre bontható:

- <header> - fejléc
- <main> - fő tartalmi egység
- <footer> - lábléc
- <nav> - navigációs elemek
- <article> - tartalmi egység (pl. cikk)
- <section> - szakasz
- <aside> - kapcsolódó, járulékos információk

A fejléc

Az oldalak fejléce a páros <header> taggel készíthető el. Az oldalaknak többnyire van fejlécük, de szükség esetén más szerkezeti elemeken (pl. article, section) is ki lehet alakítani. Viszont <header> tag nem ágyazható sem <footer> (lábléc), sem másik <header> tagbe.
A fejlécben érdemes bevezető információkat összegyűjteni. Általában a címsor, a logó, a keresési lehetőség és a navigációs elemek kerülnek ide.

A fő tartalom

A fő tartalmi egységet a páros <main> taggel érdemes jelölni. Ezt egy dokumentumon belül alapvetően csak egyszer lehet felhasználni. Több fő tartalom esetén egy kivételével az összeset el kell tüntetni a hidden paraméterrel. Erre a megoldásra akkor lehet szükség, ha cserélődik az oldal tartalma például egy gombra kattintás hatására.
A <main> taget érdemes egyedi azonosítóval ellátni, hogy az adott oldalrészt is be lehessen linkelni. A képernyőolvasó programok a link segítségével át tudják ugrani a lap tetején található blokkokat, és azonnal a tartalomra koncentrálhatnak.

A lábléc

Az oldalak lábléce a páros <footer> taggel alakítható ki. A fejléchez hasonlóan a lábléc nemcsak az oldalhoz, hanem a többi oldalszerkezeti elemhez is tartozhat, kivéve a fejlécet vagy egy másik láblécet.
A lábléc rendszerint a készítő nevét, a szerzői jogi információkat, elérhetőségeket, illetve a kapcsolódó oldalak linkjét tartalmazza.

A navigációs elemek

Az oldalon belüli, illetve az egyes oldalak közötti navigálásra szolgáló linkeket az oldalon belül a páros <nav> tagek közé kell elhelyezni.

Egy oldalon belül érdemes több ilyen blokkot is kialakítani, hogy a főmenü, az almenü és a tartalomjegyzék jól láthatóan elkülönüljön egymástól.

A tartalmi egység

Az összefüggő tartalmi egységek jelölésére a páros <article> tag szolgál. A tartalmi egység a gyakorlatban megfeleltethető egy cikknek, egy blog- vagy fórumbejegyzésnek például.
Egy oldalon tetszőleges számú tartalmi egység alakítható ki.

A szakasz (fejezet)

A weblapon a szakaszokat a páros <section> taggel alakíthatjuk ki. Az <article> tagen belül kisebb tartalmi egységeket különíthetünk el.
A szakaszokat többnyire címsor vezeti be.

A kapcsolódó információ

A páros <aside> taggel megjelölhetjük a tartalomhoz nem közvetlenül kapcsolódó tartalmi elemeket. Ide elsősorban a forrásjegyzéket, a szószedetet, a hozzászólásokat és más gyűjteményeket érdemes elhelyezni.

Gyakorlati feladat (Microsoft Teams 20230523_5.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/XWKPvVm
Bontsd az oldalt három tartalmi egységre, fejlécre, fő tartalomra és láblécre.

- *Az oldal fejléce tartalmazza az oldal tetején látható linkeket.*
- *A maradék szöveg alkossa a fő tartalmi egységet.*
- *Helyezz láblécet az oldal aljára, és írd be a következő szöveget: “Készítette: XY” (Az XY helyén a saját neved szerepeljen.)*

A fejlécen belül helyezd egy navigációs blokkba a hivatkozásokat.
A fő tartalmi egységet tagold további öt cikkre. A cikkeket úgy alakítsd ki, hogy a fő- vagy alcímet és a hozzá tartozó szöveges tartalmat tekintsd egy cikknek. A hármas szintű alcímet és a hozzá tartozó szövegrészt ne tekintsd külön egységnek.

A pozicionálási sémák, a méret kiszámítása és a mélységi sorrend

A pozicionálási sémák típusai

A weboldalak különböző elrendezéseket, layoutokat alkalmaznak. A blokkszintű elemek elhelyezkedését az úgynevezett pozicionálási sémákkal állíthatjuk be. Az egyes sémákat a deklarációs blokkban a position tulajdonsággal aktiválhatjuk, ennek értékeként a következő kulcsszavak adhatók meg:

- static (statikus)
- relative (relatív / viszonyított)
- absolute (abszolút)
- fixed (rögzített / fix)
- sticky (ragadós)

Az eltolás

A nem statikus pozicionálás esetén magunk állíthatjuk be, hogy egy adott elem milyen irányba és milyen mértékben legyen eltolva a böngészőben. Az eltolás nagysága mind abszolút, mind relatív mértékegységekkel megadható. Számértékek esetén akár negatív szám is megadható. Az eltolás mértéke a deklarációs blokkon belül az eltolás irányától függően a következő tulajdonságokkal adható meg:

- left (eltolás balról jobbra)
- right (eltolás jobbról balra)
- top (eltolás fentről lefelé)
- bottom (eltolás lentről felfelé)

A viszonyítási pontok jelentése eltér az egyes sémáknál.

A static (statikus) pozicionálás

A static (statikus) pozicionálás az alapértelmezett megjelenítési mód. Ebben az esetben a böngésző automatikusan kiszámítja a pozíciót a megjelenítő eszköz méreteinek és tulajdonságainak (szélesség, magasság, felbontás, ablakméret, stb.) figyelembevételével.
A static pozicionálású elemeknél nincs hatása a left, a right, a top és a bottom tulajdonságoknak, ezért a statikus értéket elsősorban akkor érdemes beállítani, ha egy elemnél az alapértelmezettre kell visszaállítani a megörökölt pozicionálási sémát.

A relative (viszonyított) pozicionálás

A relative (viszonyított) pozicionálás alkalmazásakor az elemek ugyanúgy sorban követik egymást, mint a static pozicionálásnál. Ebben az esetben azonban az elemek újrapozicionálhatók az eredeti helyükhöz képest a left, a right, a top és a bottom tulajdonságokkal. A relative pozicionálási sémában ezekhez a következő viszonyítási pontok kapcsolódnak:

- left - az eredeti helyzethez képest a balról jobbra való eltolás mértéke; pozitív értéknél a doboz jobbra tolódik, negatívnál balra
- right - az eredeti helyzethez képest a jobbról balra való eltolás mértéke; pozitív értéknél a doboz balra tolódik, negatívnál jobbra
- top - az eredeti helyzethez képest a felülről lefelé történő eltolás mértéke; pozitív értéknél a doboz lefelé tolódik, negatívnál felfelé

- bottom - az eredeti helyzethez képest az alulról felfelé történő eltolás mértéke; pozitív értéknél a doboz felfelé tolódik, negatívnál lefelé

Az absolute (abszolút) pozicionálás

Az absolute (abszolút) pozicionálás esetén a böngésző nem hagy ki helyet a formázott elemnek, hanem a pozicionálás azon tartalmazóelem (őselem) helyzetéhez képest történik, ahol static pozicionálástól eltérő (pl. relative) pozicionálás van beállítva.
Ha valamennyi tartalmazóelem static pozicionálású, akkor az eltolás a böngészőablak felső, bal, alsó és jobb oldalához képest történik.

A fixed (rögzített) pozicionálás

A fixed (rögzített) pozicionálással úgy lehet elemet elhelyezni a képernyőn, hogy az a dokumentum görgetése során is ugyanott marad.

A sticky (ragadós) pozicionálás

A sticky (ragadós) pozicionálás a relative és a fixed helyzetmegadás kombinációja. Az elem egy meghatározott pontig relative pozicionálásúként működik, majd ennek elérése után rögzítetté válik. A rögzítési pont a left, a right, a top és a bottom tulajdonságokkal határozható meg.

A szélesség és a magasság kiszámítása

A weblapok készítése során számos olyan helyzet adódhat, amikor a rendelkezésre álló hely alapján kell meghatározni egy doboz méretét. Ezek a számítások a calc() függvénnyel végezhetők el.
A függvény megadásakor a műveleti jelek (összeadás +, kivonás -, szorzás *, osztás /) előtt és mögött szóköznek kell lennie.
A width: calc(100% - 100px) tulajdonság-érték párral például megadhatjuk, hogy az elem szélessége a rendelkezésre álló hely mínusz 100 px legyen. A width: calc(100% / 2) deklarációval beállítható, hogy az elem szélessége a rendelkezésre álló hely fele legyen.

Példa a szélesség és a magasság kiszámítására

A következő kódolásban a doboz szélessége úgy van beállítva, hogy a rendelkezésre álló helynél 100 px-lel keskenyebb és középre igazított legyen.

```
<head>
<style>
  body {background-color: gray}

  div {
  background-color: white; /* Háttérszín */
  padding: 10px; /* Belső margó */
  margin: 0 auto; /* Középre igazítás */
  width: calc(100% - 100px); /*Szélesség kiszámítása */
  }
</style>
</head>

<body>
<div>
<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Ratione ducimus, assumenda tempore nemo aspernatur eaque dolore repellat beatae delectus ut voluptates maxime aliquid id, quod doloribus autem sint quis. Obcaecati.</p>
<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Ratione ducimus, assumenda tempore nemo aspernatur eaque dolore repellat beatae delectus ut voluptates maxime aliquid id, quod doloribus autem sint quis. Obcaecati.</p>
</div>
```

A mélységi sorrend

A pozicionálási sémák alkalmazása során előfordulhat, hogy egyes elemek egymást takarva jelennek meg. A mélységi sorrenddel beállítható, hogy az egymást fedő elemek közül melyik legyen a takaró- és melyik a takart elem. Ez a deklarációs blokkon belül a z-index tulajdonsággal adható meg, és azt fejezi ki, hogy az adott elem milyen koordinátát foglal el a képernyő síkjára merőlegesen vetített z tengelyen.
Mindig az az elem válik takaróelemmé, amelynek z-indexe nagyobb értékű. Ha az elemeknek azonos nagyságú a z-indexük, akkor a kódolásban később szereplő válik takaróelemmé. A z-index alapértelmezett értéke 0, míg az auto érték beállítása esetén a böngésző automatikusan határozza meg a mélységi sorrendet az elem kódolásban elfoglalt helye alapján.

A gyakorlatban a z-index értékét nem egyesével, hanem nagyobb (például tízes) értékekkel érdemes növelni, így szükség esetén két elem közé könnyedén helyezhetünk újabb elemeket. Máskülönben ilyenkor a többi elem z-indexét is módosítani kell.

Gyakorlati feladat (Microsoft Teams 20230525_1.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/abZRzBv
A kiinduló weboldalon tárolóelemeket láthatsz. A nagy tárolóelemben hat kisebb tárolóelem van, amelyekkel kipróbálhatod a különböző pozicionálási sémákat. Minden módosítás után vizsgáld meg az adott elem új pozicióját. Figyeld meg, hogy megmaradt-e a doboz eredeti helye, valamint takarásba került-e bármelyik elem.
A második doboz pozicionálása legyen relative és told el fentről -50 px-nyire, míg balról 70 px-nyire.
A harmadik dobozra a következő beállításokat alkalmazd:

- *Az elem legyen absolute pozicionálású.*
- *A felső és bal oldali viszonyítási pontját állítsd 0-ra.*
- *Ezután a bal oldali helyett a jobb oldali viszonyítási pontját állítsd 0-ra.*
- *A container osztályú tárolóelem helyzetmegadási módját állítsd át relative-ra, és figyeld meg mi történik a harmadik dobozzal.*

Az ötödik dobozt helyezd el a tárolóelem jobb alsó sarkába.
A negyedik doboz legyen az oldalon fixed pozicionálású, és told el balról 30 px-nyire, míg fentről 350 px-nyire.
A hatodik doboz helyzetmegadása legyen sticky. Állítsd be, hogy felülről 150 px-nél legyen a rögzítési pontja.

A túlnyúlás, a láthatóság és az átlátszatlanság

A túlnyúlással kapcsolatos beállítások

Weblapszerkesztés közben előfordulhat, hogy egy terjedelmesebb tartalom nem fér bele a kívánt méretű dobozba. Ilyenkor a deklarációs blokkon belül az overflow tulajdonsággal állíthatjuk be, hogyan jelenjen meg a túlnyúló tartalom. Ennek értékei a következő kulcsszavak lehetnek:

- visible - a tartalom mindenképp látható marad, adott esetben úgy, hogy kilóg a dobozból
- hidden - a túlnyúló tartalom nem jelenik meg
- auto - ha a tartalom nem fér a dobozba, egy gördítősáv jelenik meg
- scroll - a doboz mellett minden esetben gördítősáv jelenik meg, akkor is, ha nem lenne rá szükség

Példa a túlnyúlással kapcsolatos beállításokra

```
<head>
<style>
  .példa1 {
  width: 170px;
  height: 120px;
  overflow: hidden}

  .példa2 {position: absolute; top: 0px; left: 200px;
  width: 170px;
  height: 120px;
  overflow: visible}

  .példa3 {position: absolute; top: 250px;
  width: 170px;
  height: 120px;
  overflow: auto}

  .példa4 {position: absolute; top: 250px; left: 200px;
  width: 170px;
  height: 120px;
  overflow: scroll}
</style>
</head>
```

```
<body>
<p class="példa1">Lorem ipsum dolor sit amet consectetur adipisicing elit. Laudantium consequatur provident in reiciendis quae, fuga ea repellat mollitia commodi velit labore maxime exercitationem. Aliquam cupiditate consequuntur ab recusandae dolorum. Quidem!</p>
<p class="példa2">Lorem ipsum dolor sit amet consectetur adipisicing elit. Laudantium consequatur provident in reiciendis quae, fuga ea repellat mollitia commodi velit labore maxime exercitationem. Aliquam cupiditate consequuntur ab recusandae dolorum. Quidem!</p>
<p class="példa3">Lorem ipsum dolor sit amet consectetur adipisicing elit. Laudantium consequatur provident in reiciendis quae, fuga ea repellat mollitia commodi velit labore maxime exercitationem. Aliquam cupiditate consequuntur ab recusandae dolorum. Quidem!</p>
<p class="példa4">Lorem ipsum dolor sit amet consectetur adipisicing elit. Laudantium consequatur provident in reiciendis quae, fuga ea repellat mollitia commodi velit labore maxime exercitationem. Aliquam cupiditate consequuntur ab recusandae dolorum. Quidem!</p>
</body>
```

A láthatóság

A deklarációs blokkon belül a visibility tulajdonsággal rejthetők el, illetve jeleníthetők meg az egyes elemek. Ennek értékeként a következő kulcsszavak adhatók meg:

- visible - az elem látható
- hidden - a böngésző az elemet láthatatlanná teszi, de a helyét továbbra is kihagyja

Ha azt szeretnénk, hogy az elem teljesen eltűnjön akkor a deklarációs blokkban a korábban látott display: none tulajdonság-érték párt kell megadni.

Példa a láthatóság beállítására

```
<head>
<style>
   div {background-color: lightgray;
   border: 2px solid black;
   overflow: auto;}
   #példa1 {visibility: visible}
   #példa2 {visibility: hidden}
   #példa3 {display: none}
</style>
</head>

<body>
<div>
<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Ad vero molestiae at eveniet rem excepturi et aliquid error saepe similique! Sequi iusto dicta voluptate nisi dolorum adipisci distinctio, aut amet?</p>
<p id="példa1">Lorem ipsum dolor sit amet consectetur adipisicing elit. Ad vero molestiae at eveniet rem excepturi et aliquid error saepe similique! Sequi iusto dicta voluptate nisi dolorum adipisci distinctio, aut amet?</p>
</div>
```

```
<div>
<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Ad vero molestiae at eveniet rem excepturi et aliquid error saepe similique! Sequi iusto dicta voluptate nisi dolorum adipisci distinctio, aut amet?</p>
<p id="példa2">Lorem ipsum dolor sit amet consectetur adipisicing elit. Ad vero molestiae at eveniet rem excepturi et aliquid error saepe similique! Sequi iusto dicta voluptate nisi dolorum adipisci distinctio, aut amet?</p>
</div>

<div>
<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Ad vero molestiae at eveniet rem excepturi et aliquid error saepe similique! Sequi iusto dicta voluptate nisi dolorum adipisci distinctio, aut amet?</p>
<p id="példa3">Lorem ipsum dolor sit amet consectetur adipisicing elit. Ad vero molestiae at eveniet rem excepturi et aliquid error saepe similique! Sequi iusto dicta voluptate nisi dolorum adipisci distinctio, aut amet?</p>
</div>
</body>
```

Az átlátszatlanság

A deklarációs blokkon belül az opacity tulajdonság alkalmazásával átlátszatlanság állítható be.
Fontos, hogy ilyenkor nem átlátszóságról van szó, mivel átlátszóság esetén a nagyobb érték nagyobb átlátszóságot eredményez. Az opacity tulajdonság ezzel szemben fordítva működik. Minél nagyobb értéket adunk meg, annál kevésbé lesz átlátszó az elem.
Az opacity értékeként 0 és 1 közötti valós számokat adhatunk meg. Az 1-es érték a teljes átlátszatlanságot, míg a 0-s a teljes átlátszóságot jelenti.

Példa az átlátszatlanság beállítására

```
<style>
  body {background-color: yellow}

  div {background-color: lightgray;
  border: 2px solid black;
  overflow: auto;}

  .példa1 {height: 50px;
  opacity: 0.2}

  .példa2 {height: 50px;
  opacity: 0.4}

  .példa3 {height: 50px;
  opacity: 0.6}

  .példa4 {height: 50px;
  opacity: 0.8}
</style>
</head>
```

```
<body>
  <div class="példa1">
  </div>
  <div class="példa2">
  </div>
  <div class="példa3">
  </div>
  <div class="példa4">
  </div>
</body>
```

Gyakorlati feladat (Microsoft Teams 20230525_2.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/yLJRevd
A weboldalon négy tárolóelemet találsz, amelyekben egy-egy gyümölcs leírása olvasható. A tartalom terjedelme azonban mindenhol nagyobb, mint a dobozok mérete.
Készíts egy-egy ID selectort a tárolóelemekhez.
Rejtsd el az alma azonosítójú elem túlcsordulását.
Maradjon látható a korte azonosítójú elem túlcsordulása.

A cseresznye azonosítójú elem legyen mindig görgethető.
A szolo azonosítójú elem csak akkor legyen görgethető, ha szükséges.

Gyakorlati feladat (Microsoft Teams 20230525_3.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/gOMBPyd
A weboldalon a hat tárolóelem eltakar egy képet. A selectorok módosításával végezd el a feladatot.
Rejtsd el az első tárolóelem láthatóságát.
A második tárolóelem legyen 80%-ban látható.
A harmadik tárolóelem legyen 40%-ban áttetsző.
A negyedik tárolóelem legyen 40%-ban átlátszatlan.
Az ötödik tárolóelem legyen 80%-ban áttetsző.
A hatodik tárolóelem legyen átlátszatlan.

Az öröklődés és a CSS-rangsorolás

Az öröklődés és a CSS-reset

Az öröklődés

Az öröklődéssel kapcsolatos alapfogalmak

A szülő-, illetve őselem(ek) bizonyos tulajdonságai átöröklődnek a bennük tárolt elemekre, ezeket a gyermekelemeiknek szokás nevezni.

```
<body>
  <p>A<b>HTML5</b>-szabvány számos tulajdonságot
tartalmaz.</p>
</body>
```

A fenti kódolás alapján a következő állítások fogalmazhatók meg:

- A <p> elem szülője a <b> elemnek, azaz a <b> elem <p> gyermeke.
- A <body> elem szülője a <p> elemnek, azaz a <p> elem a <body> elem gyermeke.
- A <body> elem őse a <b> elemnek, azaz a <b> elem a <body> elem leszármazottja.

Példa az öröklődésre

A példában látható <b> tag tartalma kék színnel jelenik meg. Habár a <b> elemre nincs ilyen színmegadás külön beállítva a stíluslapban, a szülőelemre vonatkozó szabály érvényesül a gyermekén is. Ugyanez igaz a betűtípus beállítására is, ez a <body> elemtől öröklődik át.

```
<head>
<style>
  body {font-family: Arial, sans-serif}
  p {color: blue}
</style>
</head>

<body>
  <p>A <b>HTML5</b>-szabvány számos újdonságot
tartalmaz.</p>
</body>
```

Gyakran használt öröklődő és nem öröklődő tulajdonságok

Az egész oldalra vonatkozó tulajdonságok

Az egész oldalra vonatkozó tulajdonságokat a <body> elem tulajdonságaiként érdemes beállítani, hiszen ez az őse az összes, böngészőben megjelenített elemnek.
A font-familyt például ajánlott már a <body> elem formázásánál megadni, hogy az oldal valamennyi szöveges eleme ugyanazzal a betűtípussal jelenjen meg. Ha bizonyos elemeknél később mégis eltérő betűtípust használnánk, akkor az adott elemnél lehetőségünk van a beállítás egyedi felülbírálására.

Példa az egész oldalra vonatkozó tulajdonság beállítására és annak egyedi felülbírálására

```
<head>
<style>
  body {font-family: Arial, sans-serif}
  h1 {font-style: Georgia, serif}
</style>
</head>

<body>
<h1>Címsor</h1>
<p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Quaerat rem sapiente magnam! Ea, voluptate Reprehenderit molestias assumenda tenetur, dolorum beatae laboriosam laborum eveniet dolores vitae ullam soluta maiores molestiae suscipit</p>
</body>
```

A számított érték öröklődése

Az öröklődés másik fontos eleme, hogy a gyermekelemek nem a szülőknél megadott relatív értékeket öröklik, hanem azok számított értékét.
A korábban bemutatott text-indent tulajdonság, amellyel a szövegek első sora húzható be, például egy öröklődő tulajdonság. Ha a <body> elemre alkalmazzuk, akkor a tulajdonság valamennyi, böngészőben megjelenített elemre érvényesül. Ha ezt az értéket 3 em-re, azaz a betűméret háromszorosára állítjuk be, akkor a behúzások mértéke egységesen a <body> elemre érvényes betűméret háromszorosa lesz, függetlenül attól, hogy bizonyos elemek (pl. címsorok) előtte eltérő betűméretben jelennek meg.

Példa a számított értékek öröklődésére

Habár a címsor betűmérete 30 px, a behúzásnál beállított 3 em érték esetén nem ez, hanem a <body> tagnél megadott 15 px háromszorosa jut érvényre.

```
<head>
<style>
  body
  {border: 1px solid black; /*szegély*/
  padding: 10px; /*belső margó*/
  font-size: 15px; /*betűméret*/
  text-indent: 3em; /*első sor behúzása*/
  }

  h1 {
  font-size: 30px}
</style>
</head>

<body>
<h1>Címsor</h1>
<p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Vel ducimus sequi eius est! Sint, aut facilis. Saepe doloremque iure rerum assumenda atque, cupiditate nobis omnis laudantium iste, similique repellat at.</p>
</body>
```

A nem öröklődő tulajdonságok örökítése

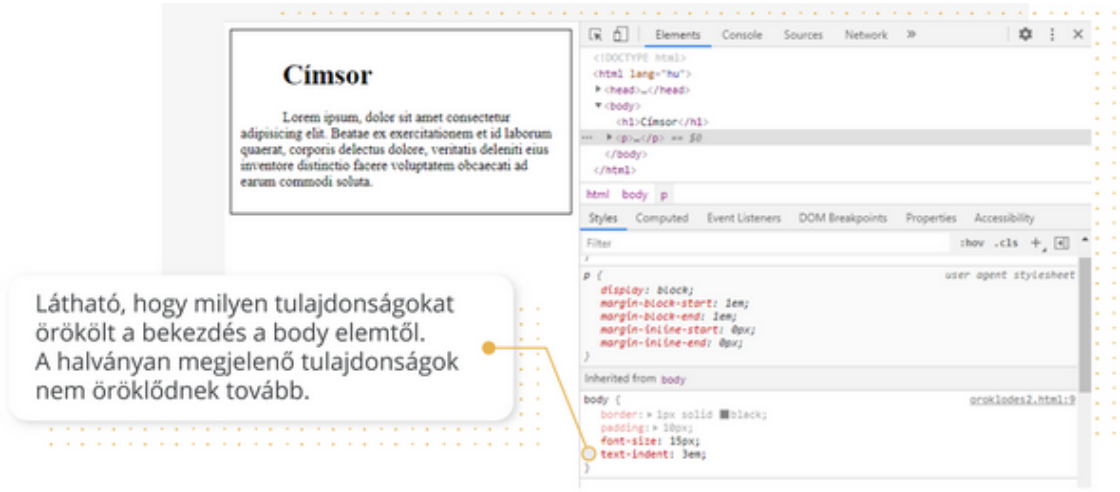
Weblapszerkesztés esetén előfordulhatnak olyan esetek, amikor az lenne a kedvező, ha egy nem öröklődő tulajdonság (pl. border) érvényesülne a szülőelem gyermekeire. Ilyenkor a gyermekelemre vonatkozó deklarációs blokkban az inherit kulcsszót szükséges megadni az örökíteni kívánt tulajdonság értékeként.

Példa a nem öröklődő tulajdonságok örökítésére

```
<head>
<style>
  body
  {border: 2px solid blue;
  padding: 10px;
  }

  p {
  border: inherit}
</style>
</head>

<body>
<h1>Címsor</h1>
<p>Lorem ipsum dolor sit amet consectetur, adipisicing
elit.</p>
<p>Eum magni doloribus amet laudantium reiciendis,
animi minima
explicabo ratione exceptiru sed volutatibus vel porro
placeat
sint expedita sunt eveniet earum atque.</p>
</body>
```

Az öröklődő tulajdonságok a böngésző fejlesztői eszközeiben

Amikor a böngészőprogramba beépített fejlesztői eszközöket használjuk, az adott elem vizsgálatánál megjelennek azok a tulajdonságok is, amelyeket a szülő- vagy őselemektől örököl.
Közülük halványabban jelennek meg azok a tulajdonságok, amelyek alapértelmezetten nem öröklődnek tovább az elemre.



Az em és a rem mértékegységek használata

Az értékek öröklődése olykor nem várt problémát okozhat. Közülük az egyik legtipikusabb eset a listaelemek egymásba ágyazása relatív betűméret megadása mellett. Ilyenkor például a listára beállított 2 em nagyságú betűméret a beágyazott elemek esetében már az alapértelmezett betűméret négyszerese lesz, mivel a böngésző az első szintű (már duplázott) elemek méretét kétszerezi meg.
Ez a probléma úgy előzhető meg, hogy az em helyett a rem mértékegységet használjuk. Ez minden esetben a <html> elemnél megadott értéket veszi alapul. Ezzel a módszerrel használhatjuk a relatív méretmegadást, és ez nem okoz megjelenítésbeli gondot még az elemek egymásba ágyazása esetén sem.

Példa az em mértékegység használatára

A <li> elemre beállított 2 em betűméret a beágyazott elemeknél már nem az alapértelmezett betűméret dupláját, hanem a négyszeresét eredményezi.

```
<head>
<style>
  html {font-size: 16px}
  body {font-size: 16px}
  li {font-size: 2em}
</style>
</head>

<body>
<ul>
  <li>Lorem ipsum</li>
  <li>Lorem ipsum
    <ul>
      <li>Lorem ipsum</li>
      <li>Lorem ipsum</li>
    </ul>
  </li>
  <li>Lorem ipsum</li>
</ul>
</body>
```

Példa a rem mértékegység használatára

A <li> elemre beállított 2 rem betűméret a beágyazott elemeknél is az alapértelmezett betűméret dupláját eredményezi.

```
<head>
<style>
  html {font-size: 16px}
  body {font-size: 16px}
  li {font-size: 2rem}
</style>
</head>

<body>
<ul>
  <li>Lorem ipsum</li>
  <li>Lorem ipsum
    <ul>
      <li>Lorem ipsum</li>
      <li>Lorem ipsum</li>
    </ul>
  </li>
  <li>Lorem ipsum</li>
</ul>
</body>
```

A CSS-reset

Mi a CSS-reset?

CSS-környezetben több olyan tulajdonság is van, amelyek alapértelmezett értéke nem rögzített a szabványban. Ezek megjelenése ezáltal a böngészőprogramba épített alapértelmezett stíluslapoktól függ.
A gyakorlatban ez például azt jelenti, hogy böngészőprogramonként változhat, mekkora margó van egy elem előtt és mögött, illetve milyen nagyságú belső margó található a tartalom és a szegély között.
A CSS-reset (CSS-alaphelyzet beállítása) arra szolgál, hogy valamennyi esetben az általunk használt stíluslapban alapértelmezett értékek jussanak érvényre. Ennek alkalmazásával elérhetjük, hogy az oldalunk az összes böngészőben egyformán jelenjen meg.

A CSS-reset univerzális selectorral

A CSS-reset egyszerűen beállítható az univerzális selector (*) alkalmazásával. Ez egy olyan speciális selector, amellyel tulajdonságokat csatolhatunk az állományban lévő összes HTML-taghez.
Ha szeretnénk, hogy a honlapunk minden böngészőben egyforma margó- és belső margóértékkel jelenjen meg, érdemes először az univerzális selectorral lenullázni az értékeket, majd elemenként a type selectorral beállítani az egyedi értékeket. Ezt követően a honlap a böngészőtől függetlenül egységesen jelenik meg.

Példa a CSS-reset alkalmazására

```
<head>
<style>
  * {
  margin: 0;
  padding: 0;
  }

  p {
  margin: 10px;
  padding: 5px;
  }
</style>
</head>

<body>
<p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Illum voluptates architecto consequatur suscipit similique possimus eligendi numquam dolor, id error rem quis exercitationem veritatis quidem nisi perferendis eos aliquam aut.</p>

<p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Illum voluptates architecto consequatur suscipit similique possimus eligendi numquam dolor, id error rem quis exercitationem veritatis quidem nisi perferendis eos aliquam aut.</p>
</body>
```

Gyakorlati feladat (Microsoft Teams 20230606_1.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/eYzPKjW
A feladat során megfigyelheted, hogy melyik CSS-tulajdonság öröklődik tovább a gyermekelemre.

Készíts egy univerzális selectort, amellyel megadhatod az oldalad alapbeállításait. A CSS-reset során állítsd be, hogy egyik elemnek se legyen belső és külső margója. Ellenőrizd, mit eredményezett ez a beállítás a kétszintű felsorolásnál.
Az oldal törzsére állítsd be a következőket:

- *A betűtípus legyen Georgia.*
- *A belső margó vízszintes irányban legyen 20 px, míg függőleges irányban 0.*

A címsor felső margója legyen 24 px, míg az alsó margója ennek fele.
A felsorolás bal oldali margóját állítsd 30 px nagyságúra.
A felsorolás betűit formázd a következők szerint:

- *A betűméret legyen a szülőelem másfélszerese (em).*
- *A betűméret legyen a gyökérelem alapértelmezett betűméretének másfélszerese (rem).*

A felsorolásokat formázd félkövér stílussal, majd az ul.belso{} selector segítségével állítsd be, hogy a belső felsorolás elemei ne legyenek félkövérek.
A tárolóelem szélessége legyen 600 px, a háttérszíne pedig kék (R: 157, G: 203, B: 228). A háttérszín legyen 70%-ban áttetsző.
A tárolóelemre állíts be egy 2 px vastag, folytonos sötétzöld szegélyt. A szegély nem öröklődő tulajdonság, de érd el, hogy a főcím megörökölje ezt a beállítást.
Zárásként vizsgáld meg a betűszínek öröklődését.

- *A tárolóelemre állítsd be, hogy a szöveg színe legyen sötétkék.*
- *A cím betűszíne legyen sötétzöld.*
- *A belső felsorolás betűszínének add az olivedrab értéket.*

A CSS-rangsorolás

A kiértékelési sorrend (rangsor)

Mivel a stíluslapok olykor ugyanazon elemre egyszerre többféle szabályt is tartalmazhatnak, szükség van egy úgynevezett rangsorolási szempontrendszerre. Ez meghatározza, hogy az egymást felülíró szabályok közül melyik jusson érvényre.

A következő példában a CSS3-szöveg egy <b> tagben van elhelyezve. Amint látod, a <b> elemre több szabály is vonatkozik:

```
<head>
<style>
p {color: gray}
b#fontos {color: red}
b {color: green}
b.narancs {color: orange}
</style>
</head>

<body>
<p>A <b class="narancs" id="fontos" style="color: purple">CSS3</b>
rengeteg újdonságot tartalmaz.</p>
</body>
```

Az alaptulajdonság, az örökölt érték és a selectorban megadott tulajdonság

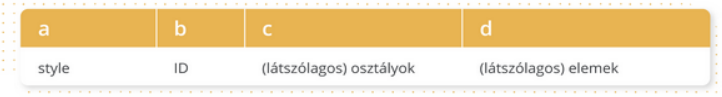
Az egyes elemek böngészőprogramokban való megjelenését sok tényező befolyásolja. Egy megjelenített elem színét például a következő tényezők határozzák meg:

- A böngészőprogramnak először a stíluslapban kell megkeresnie, hogy van-e az elem színét meghatározó szabály.
- Ha nincs a stíluslapban az elemre vonatkozó beállítás, akkor ki kell derítenie, hogy örökölhet-e színt az adott elem a szülő- vagy őselemektől.
- Ha nincs örökölt szín, akkor az alapértelmezett színt szükséges elővennie.

```
<style>
  p {color: blue}
/*A bekezdés kék lesz*/
</style>

<style>
  body {color: blue}
/*a bekezdés kék lesz az örökölt érték miatt*/
</style>

<style>
/*A bekezdés fekete lesz, mert ez az alapértelmezett szín*/
</style>
```

A selectorok egyedisége

A selectorok egyedisége sajátos rangsorolással jár. Minél egyedibb egy selector, annál inkább érvényre jutnak a hozzá tartozó deklarációs blokkban megadott szabályok.
A selectorok egyedisége számértékekben adható meg. Ezeket a rendszer a következő táblázat alapján generálja:



a	b	c	d
style	ID	(látszólagos) osztályok	(látszólagos) elemek

A táblázatot a következő szabályok szerint kell kitölteni:

- Az “a” oszlopba akkor kerül egy 1-es számjegy, ha a tulajdonság-érték pár a style paraméter értékeként szerepel.
- A “b” oszlopba az ID selectorok számát kell beírni.
- A “c” oszlopba a class selector és a pseudo classok számát kell beírni.
- A “d” oszlopba a type selector és a látszólagos elemek (pseudo element) számát kell beírni.

A táblázat mezőibe csak egy számjegy írható, így hosszabb, akár tíz tagból álló, úgynevezett selectorláncok esetén tízesnél nagyobb számrendszerben (pl. 16-os) szükséges gondolkodni, hiszen csak ezzel oldható meg, hogy a 9-nél nagyobb értékeknél is csak egy számjegy legyen a mezőkbe írva.

Példa az egyes selectorok egyediségének kiszámítására

selector	a	b	c	d	egyediség
*	0	0	0	0	0
li { . . . }	0	0	0	1	1
.szegely { . . . }	0	0	1	0	10
a:link { . . . }	0	0	1	1	11
li.fontos { . . . }	0	0	1	1	11
#doboz1 { . . . }	0	1	0	0	100
style= ” ”	1	0	0	0	1000

A meghatározás sorrendje

A selectorok egyedisége mellett egy másik rangsorolási szempont a meghatározás sorrendje. Ez azt jelenti, hogy a később megadott deklaráció nagyobb súllyal bír a korábban megadottaknál.

A meghatározás sorrendje akkor érvényesül, ha a szabályokhoz tartozó selectorok egyedisége azonos, vagyis ezzel a módszerrel nem lehet felülbírálni a kisebb egyediségű selectort.
Az importált stíluslapok használatakor úgy kell tekinteni a kódot, mintha az importált stíluslap tartalma az adott helyre lenne beágyazva.
A belinkelt külső stíluslapokat pedig úgy kell vizsgálni, mintha azok tartalma egymás után lenne fűzve.

Példa a meghatározás sorrendjének érvényesülésére

```
<head>
<style>
  p {color: green}
  p {color: orange}
</style>
</head>

<body>
  <p>A HTML5 számos újdonságot tartalmaz.</p>
</body>
```

Az explicit súly szerinti rendezés

Az egyes deklarációknak nagyobb súlyt lehet adni az !important attribútummal. Ez a gyakorlatban kiiktatja a meghatározás sorrendjét, azaz a később megadott szabály nem hat majd a korábbira.

Példa az explicit súly szerinti rendezésre

Az !important paraméter használata miatt a későbbi deklaráció nem írja felül a korábbi hatását.

```
<head>
<style>
  p {color: green !important}
  p {color: orange}
</style>
</head>

<body>
  <p>A HTML5 számos újdonságot tartalmaz.</p>
</body>
```

Gyakorlati feladat (Microsoft Teams 20230613_1.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/RwReBer
A feladat megoldása során előfordulhat, hogy egy adott elemre ugyanaz a tulajdonság többször is érvényesül, csak más-más értékekkel. Vizsgáld meg, hogy ilyen esetekben melyik beállítás írja felül a másikat.
Az oldal törzs betűszínét állítsd sötétkékre.
A bekezdések betűszíne legyen darkslategray színű, a szöveg pedig sorkizárt.
A HTML-fájlban néhány elemre a kiemel osztály van elhelyezve. Készíts ezekhez az elemekhez egy kiemel nevű class selectort és formázd a szöveget a következő beállításokkal:

- *A szöveg színe legyen fekete, és növeld a betűméretet az alapértelmezetthez képest 10%-kal.*
- *A szöveg legyen félkövér és kiskapitális stílusú.*

Készíts egy p.kiemel{} selectort, és állítsd be vele, hogy a szöveg színe sötétvörös, betűmérete az alapértelmezettnél 30%-kal nagyobb legyen.
A főcímre sötétvörös betűszínt állíts be egy type selectorral.
A HTML-fájlban, a cikken belüli egyes szintű címsorra készíts egy inline (sorközi) formázást, és a betűszínt állítsd darkolivegreen zöld színűre. Vizsgáld meg, milyen változás történt.
Használd az !important attribútumot, és változtasd az összes egyes szintű címsort sötétvörös színűre.

A speciális selectorok és hibakeresési praktikák CSS-ben

A speciális selectorok használata

Az öröklődés

Az egyes elemek között szülő-gyermek, illetve ős-leszármazott kapcsolat áll fenn. A honlapszerkesztés során gyakran előfordulnak olyan helyzetek, amikor az a legelőnyösebb, ha az elemeket az alapján tudjuk kijelölni, hogy milyen más elemekbe vannak beágyazva. Ezt a gyakorlatban az úgynevezett speciális selectorok segítségével valósíthatjuk meg.

A leszármazott elemek kijelölése

Ha egy beágyazott elemnek egyedi tulajdonságot szeretnénk beállítani, azt megtehetjük a rámutató selectorok szóközökkel való összekapcsolásával. A szóköz ebben az esetben kombinátorként működik, azaz a selectorok közötti kapcsolatot (leszármazotti viszonyt) állíthatjuk be vele.

Ezzel a módszerrel tetszőleges hosszúságú selectorláncokat alkothatunk.

Példa a leszármazott elemek kijelölésére

A következő példában a címsorokra és a hangsúlyos elemekre is a kék betűszín van beállítva. Ahhoz, hogy a címsorban lévő hangsúlyos elemek színét is módosíthasd, egy speciális selectort kell alkalmaznod.

Jelen esetben a h1 em selectorlánc két type selectort tartalmaz, az egyedisége 2, míg az em selectoré csupán 1. Emlékezz vissza, hogy mindig a speciálisabb szabály érvényesül, ezért változik meg a címsorban lévő tagekbe foglalt szöveg színe.

```
<head>
<style>
  h1 {color: blue}
  em {color:blue}
  h1 em {color: purple}
</style>
</head>

<body>
<h1>Ismerkedés a <em>CSS3</em>-szabvánnyal</h1>
<p>A <em>CSS3</em> nem más, mint egy szabvány a
weboldalak megjelenésének leírására.</p>
</body>
```

A gyermekelemek kijelölése

Selectorként nemcsak leszármazási viszonyt, hanem közvetlen leszármazást, vagyis gyermekviszonyt is beállíthatunk. Ilyenkor kombinátorként a balra mutató, “nagyobb” relációs jelet (>) kell alkalmazni.
Ha például a div > p > span speciális selectort használjuk, akkor ez azokra a elemekre lesz hatással, amelyek elemekbe ágyazott bekezdésekben helyezkednek el.

Példa a gyermekelemek kijelölésére

A következő példában a deklaráció csak azokra a bekezdésekre érvényesül, amelyek közvetlenül a <body> elem gyermekei.

```
<head>
<style>
  body > p {
  font-weight:bold;
  border: 1px solid black}
</style>
</head>

<body>
  <p>Első bekezdés</p>
  <div>
  <p>Második bekezdés</p>
  <p>Harmadik bekezdés</p>
  </div>
  <p>Negyedik bekezdés</p>
</body>
```

Az elemek kijelölése paraméter alapján

Az elemeket az alapján is kijelölhetjük, hogy van-e adott paraméterük (attribútumuk). A selectoroknak ezt a típusát nevezik attribútum selectornak. Ebben az esetben az elem kódja után szögletes zárójelben ([ ]) kell feltüntetni a kijelölni kívánt paraméter kódját.
Szükség esetén a selectorban egy konkrét paraméter-érték párra is készíthetünk kijelölőt. Ilyenkor a szögletes zárójelben a paraméter kódja mellett meg kell adnunk a hozzá tartozó értéket is.

Példa az elemek paraméter alapján való kijelölésére

A stíluslapban a lang paraméterrel ellátott elemek vannak kijelölve. A francia nyelvű szöveg mégis eltér, mivel rajta a p[lang="fr"]speciális selector hatása érvényesül.

```
<head>
<style>
  p[lang] {
  margin-left: 2em;
  color: slateblue;
  }

  p[lang="fr"] {
  color: teal;
  }
</style>
</head>

<body>
  <p>Viszontlátásra!</p>
  <p lang="en">Good-bye!</p>
  <p lang="de">Auf Wiedersehen!</p>
  <p lang="fr">Au revoir !</p>
</body>
```

Gyakorlati feladat (Microsoft Teams 20230613_2.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/bGeXRjL
Készíts egy speciális selectort és jelöld ki vele a szakaszban található képet (a szakasz közvetlen gyermekeleme a kép). Formázd a következő beállításokkal a kijelölt képet.

- *Igazítsd jobb oldalra.*
- *A kódolásban a képek általánosan úgy vannak beállítva, hogy 300 pixel szélességűek legyenek. A kijelölt elem szélességét formázd úgy, hogy a böngésző automatikusan méretezze.*
- *Mivel ezt a képet jobbra igazítottad, töröld a jobb oldali margóját, míg a bal oldalit állítsd 20 px nagyságúra.*

Az oldalon több elemet is láthatsz, ezek félkövér stílusúak. A szakaszban található elemek stílusát úgy módosítsd egy speciális selectorral, hogy az elemek a félkövér helyett dőlt stílusúak legyenek.

Az oldal bizonyos elemei a kiemel osztályba vannak besorolva. Készíts olyan speciális seletort, amely csak a kiemel osztályú elemeket jelöli ki, és állítsd be ezen elemek szövegszínét sötétbordóra (R: 121, G: 0, B: 0).
Az alt attribútummal ellátott képeket vedd körbe 3 px vastag, folytonos szegéllyel, ennek színe egyezzen meg a szakasz háttérszínével.

Gyakorlati feladat (Microsoft Teams 20230613_3.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/WNxVZGQ
A számozatlan felsoroláslista elemeire állítsd be a következőket:

- *A listaelem jele legyen decimális szám.*
- *A szöveg legyen félkövér stílusú.*
- *A betűszín legyen olivedrab.*
- *A gyökérelem betűméretéhez képest a betűméretet növeld 20%-kal.*

Vizsgáld meg, mely elemekre voltak hatással a formázások. Miért épp ezek az elemek változtak meg? Mivel a kétszintű felsorolás két egymásba ágyazott számozatlan felsorolásból áll, minden elemre érvényes a szabály.
Kombinátor felhasználásával készíts olyan stílusszabályt, amely csak a második szintű listaelemekre érvényesül.

- *Legyenek négyzetek a második szintű felsoroláselemek listajelölői.*
- *A listaelemek ne legyenek félkövéren szedve.*
- *A listaelemek színét és méretét állítsd alapértelmezettre.*

A hibakeresési praktikák CSS-ben

A hibakeresés

Ha a stíluslapban megadott tulajdonságok nem érvényesülnek, általában a következő okok állnak a háttérben:

- Nincs elmentve a módosított állomány.
- A tulajdonság nem azt a fájlt módosítja, amelyet kellene.
- A publikált honlap stíluslapállománya nincs felülírva a módosított állománnyal.
- A CSS-állomány szintaktikai hibát tartalmaz.
- A CSS-állományban van egy nagyobb rangsorral bíró szabály, és emiatt nem jut érvényre az éppen módosított szabály.

Forrás: Abonyi-Tóth Andor: Webszerkesztési alapok és Weboldalak formázása e-learning tananyag

Űrlapok

Űrlapok

A HTML űrlapokkal adatokat kérhetünk be a felhasználóktól. Például regisztráció, bejelentkezés, rendelés, kérdőív, stb.
A felhasználói adatokat leggyakrabban egy szervernek továbbítjuk feldolgozásra.

A <form> elem

A <form> elem egy tároló különböző típusú űrlap elemekhez, mint például a beviteli mező, rádiógomb, jelölőnégyzet, gomb, stb.
Szintaktikája:
<form>
    tartalom
</form>

Az <input> elem

Az <input> elem a leggyakrabban használt űrlap elem. Önmagában azonban semmit nem ér, szükséges megadni hozzá a type paramétert, amelynek segítségével megmondjuk, hogy milyen típusú űrlap elemet szeretnénk létrehozni.
Néhány példa:

- text - egysoros beviteli mező
- radio - rádiógomb (ha több lehetőség közül csak egyet lehet kiválasztani)
- checkbox - jelölőnégyzet (ha több lehetőség közül többet is lehet választani)
- submit - az űrlap elküldéséhez használható gomb
- button - kattintható gomb

A beviteli mező

Egysoros beviteli mező szöveges adatok bekéréséhez.
A <label> elem segítségével címkét adhatunk a beviteli mezőhöz. A for paraméter értéke annak az elemnek az azonosítója (id), amelyhez kapcsolódik a címke. A címkét a képernyőolvasó programok felolvassák, amikor a felhasználó ráteszi a fókuszt a beviteli mezőre.
Az id mellett egy name paramétert is használunk, bizonyos esetekben arra hivatkozunk.
A value segítségével alapértelmezett értéket adhatunk a mezőhöz. Egyébként a value értéke továbbítódik az űrlap elküldésekor.

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe">
</form>
```

A rádiógomb

A rádiógombok esetén a felhasználó egyet választhat több lehetőség közül.
Ha a rádiógombok egymás alatt helyezkednek el, érdemes a címkét az <input> elem után tenni, hogy függőlegesen egy vonalban legyenek.
Az egy csoportba tartozó rádiógombok name paraméterének értéke meg kell, hogy egyezzen. Ellenkező esetben mindegyik rádiógomb választható lesz egyszerre.
Ha szeretnénk, hogy az egyik rádiógomb alapértelmezetten be legyen jelölve, el kell látni a checked paraméterrel.

```
<p>Choose your favorite Web language:</p>

<form>
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language" value="JavaScript">
  <label for="javascript">JavaScript</label>
</form>
```

A jelölőnégyzet

A jelölőnégyzetek esetén a felhasználó többet is választhat a több lehetőség közül.
Ha a jelölőnégyzetek egymás alatt helyezkednek el, érdemes a címkét az <input> elem után tenni, hogy függőlegesen egy vonalban legyenek.
A rádiógombokkal ellentétben a jelölőnégyzeteket nem kell csoportosítani, így a name paramétereik értéke különböző.
Ha szeretnénk, hogy egy vagy jelölőnégyzet be legyen jelölve, el kell látni őket a checked paraméterrel.

```
<form>
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> I have a car</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label for="vehicle3"> I have a boat</label>
</form>
```

A submit gomb

Az űrlap elküldéséhez használható gomb. Az elküldéshez szükséges, hogy a <form> elemnek legyen action paramétere, amelynek értéke a feldolgozást végző fájl elérési útja és neve. Ez a fájl általában szerveroldali szkriptet tartalmaz, amely kezeli az űrlap adatokat.
A gomb feliratát a value paraméter értékeként adhatjuk meg.

Ha a beviteli mezők name paraméterét elhagyjuk, a beviteli mezők értéke nem kerül elküldésre.

```
<form action="action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

A <form> elem további paraméterei

A target paraméter

A paraméter határozza meg, hogy hol jelenjen meg az űrlap elküldése után kapott válasz. _blank érték esetén a válasz új ablakban vagy böngészőfülön jelenik meg, _self érték esetén az aktuális ablakban vagy böngészőfülön. Utóbbi az alapértelmezett.

```
<form action="action_page.php" target="_blank">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

A method paraméter

A paraméter az űrlap adatok elküldésekor használandó módszert határozza meg. Az adatok elküldhetők URL segítségével, ebben az esetben a get metódust használjuk és elküldhetők post metódusként. Előbbi az alapértelmezett módszer.
A get metódus tehát az URL-hez fűzi az űrlap adatokat név=érték párokban. Soha ne használjuk ezt a metódust érzékeny adatok küldésére, mivel az URL-ben láthatók és így az előzmények között is megmaradnak! Az URL hossza korlátozott (2048 karakter).
A post metódus hozzáfűzi az űrlap adatokat a HTTP-kérés törzséhez és nem az URL-ben jelennek meg. Nincs méretkorlátozás, nagy mennyiségű adat küldésére is használható.

```
<form action="action_page.php" target="_blank" method="get">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

Az autocomplete paraméter

Ha a paraméter értéke on, a böngésző automatikusan kitölti a beviteli mezőket a felhasználó által korábban beírt adatokkal, ha újratöltjük az oldalt, off esetén üresen hagyja.

```
<form action="action_page.php" autocomplete="on">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="email">Email:</label>
  <input type="text" id="email" name="email"><br><br>
  <input type="submit">
</form>
```

A novalidate paraméter

A novalidate paraméter egy logikai paraméter. Ha használjuk, az űrlap adatokat nem kell validálni az űrlap elküldésekor.

```
<form action="action_page.php" novalidate>
  <label for="email">Enter your email:</label>
  <input type="email" id="email" name="email"><br><br>
  <input type="submit">
</form>
```

Az enctype paraméter

Az űrlap adatok kódolásához használjuk. Lehetséges értékei:

- application/x-www-form-urlencoded (alapértelmezett)
- multipart/form-data (fájlfeltöltést tartalmazó űrlapoknál ezt kell használni)
- text/plain (levélküldéskor)

<u>További űrlapelemek</u>

A <select> elem

A <select> elem segítségével egy legördülő listát lehet megvalósítani.
Az <option> elemek segítségével adhatók meg a választható elemek.
Alapértelmezés szerint a legördülő lista első eleme van kiválasztva. Egy adott elem kiválasztásához a selected paramétert kell megadni.

```
<form action="action_page.php">
  <label for="cars">Choose a car:</label><br>
  <select id="cars" name="cars">
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
    <option value="fiat">Fiat</option>
    <option value="audi">Audi</option>
  </select><br>
  <input type="submit">
</form>
```

A látható elemek számának megadásához a size paramétert használhatjuk.

```
<form action="action_page.php">
  <label for="cars">Choose a car:</label><br>
  <select id="cars" name="cars" size="3">
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
    <option value="fiat">Fiat</option>
    <option value="audi">Audi</option>
  </select><br><br>
  <input type="submit">
</form>
```

Ha engedélyezni szeretnénk egynél több elem kiválasztását, akkor a multiple paramétert kell használnunk.

```
<form action="action_page.php">
  <label for="cars">Choose a car:</label>
  <select id="cars" name="cars" size="4" multiple>
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
    <option value="fiat">Fiat</option>
    <option value="audi">Audi</option>
  </select><br><br>
  <input type="submit">
</form>
```

Az elemeket csoportosítani lehet az <optgroup> elem segítségével.

```
<form action="action_page.php">
  <label for="cars">Choose a car:</label><br>
  <select name="cars" id="cars">
    <optgroup label="Swedish Cars">
      <option value="volvo">Volvo</option>
      <option value="saab">Saab</option>
    </optgroup>
    <optgroup label="German Cars">
      <option value="mercedes">Mercedes</option>
      <option value="audi">Audi</option>
    </optgroup>
  </select><br><br>
  <input type="submit" value="Submit">
</form>
```

Egy csoportot letiltani a disabled paraméterrel lehet. Ha lista elején letiltott csoport vagy elem van, akkor az első olyan elem jelenik meg alapértelmezetten, amelyik nincs letiltva.

A <textarea> elem

A <textarea> elem segítségével egy többsoros beviteli mezőt (szövegdobozt) lehet megvalósítani.
A rows paraméter segítségével adhatjuk meg a szövegdoboz látható sorainak számát.
A cols paraméter segítségével a szövegdoboz szélességét adhatjuk meg.

```
<form action="action_page.php">
  <textarea name="message" rows="10" cols="30">The cat was playing in the garden.
</textarea><br><br>
  <input type="submit">
</form>
```

A szövegdoboz méretét stíluslappal is megadhatjuk.

```
<form action="action_page.php">
  <textarea name="message" style="width: 400px; height: 300px;">The cat was playing in
the garden.</textarea><br><br>
  <input type="submit">
</form>
```

A <button> elem

A <button> elem segítségével egy kattintható gombot lehet megvalósítani.
Mindig meg kell adni a type paramétert a <button> elemhez, a különböző böngészők különböző alapértelmezett típusokat használnak.

```
<button type="button" onclick="alert('Hello World!');">Click Me!</button>
```

A <fieldset> és a <legend> elemek

A <fieldset> elem a kapcsolódó adatok űrlapon belül történő csoportosítására használható.
A <legend> elem segítségével egy felirat adható meg a <fieldset> elemhez.

```
<form action="action_page.php">
  <fieldset>
    <legend>Name:</legend>
    <label for="fname">First name:</label><br>
    <input type="text" id="fname" name="fname" value="John"><br>
    <label for="lname">Last name:</label><br>
    <input type="text" id="lname" name="lname" value="Doe"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

A <datalist> elem

A <datalist> elem előre meghatározott elemek listáját adja egy <input> elemhez. A felhasználó az adatbevitel során választani tud az előre meghatározott elemek legördülő listájából.
Az <input> elem list paraméterének értéke a <datalist> elem id paraméterének értéke kell, hogy legyen.

```
<form action="action_page.php">
  <input list="browsers" name="browser">
  <datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist><br><br>
  <input type="submit">
</form>
```

<u>További input típusok</u>

A password típus

A password típus segítségével egy jelszó beviteli mezőt lehet megvalósítani.

```
<form action="action_page.php">
  <label for="username">Username:</label><br>
  <input type="text" id="username" name="username"><br>
  <label for="pwd">Password:</label><br>
  <input type="password" id="pwd" name="pwd"><br><br>
  <input type="submit" value="Submit">
</form>
```

A reset típus

A reset típus segítségével egy gomb valósítható meg, amellyel törölhetők az űrlap mezők (visszaállítja az alapértelmezett értékeket).

```
<form action="action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
  <input type="reset">
</form>
```

A color típus

Szín kiválasztásához használható beviteli mező. Böngészőtől függően egy színválasztó ablak jelenik meg.

```
<form action="action_page.php">
  <label for="favcolor">Select your favorite color:</label><br>
  <input type="color" id="favcolor" name="favcolor" value="#ff0000"><br>
  <input type="submit" value="Submit">
</form>
```

A date típus

Dátum kiválasztásához használható beviteli mező. Böngészőtől függően egy dátumválasztó ablak jelenik meg.

```
<form action="action_page.php">
  <label for="birthday">Birthday:</label><br>
  <input type="date" id="birthday" name="birthday"><br>
  <input type="submit" value="Submit">
</form>
```

A min és max paraméterek használatával korlátozásokat is beállíthatunk a dátumokhoz.

```
<form action="action_page.php">
  <label for="datemin">Enter a date after 2000-01-01:</label>
  <input type="date" id="datemin" name="datemin" min="2000-01-02"><br><br>

  <label for="datemax">Enter a date before 1980-01-01:</label>
  <input type="date" id="datemax" name="datemax" max="1979-12-31"><br><br>

  <input type="submit" value="Submit">
</form>
```

Az e-mail típus

E-mail cím bekéréséhez használható beviteli mező. A böngésző támogatásától függően az e-mail elküldésekor automatikusan validálható.

```
<form action="action_page.php">
  <label for="email">Enter your email:</label><br>
  <input type="email" id="email" name="email"><br><br>
  <input type="submit" value="Submit">
</form>
```

Az image típus

Kép használata submit gombként. A kép elérési útvonalát és nevét az src paraméter értékeként kell megadni.

```
<form action="action_page.php">
  <label for="fname">First name: </label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name: </label><br>
  <input type="text" id="lname" name="lname"><br>
  <input type="image" src="submit.png" alt="Submit" width="50" height="50">
</form>
```

A file típus

Fájl kiválasztásához használható beviteli mező. Egy Tallózás (Browse) gomb jelenik, amellyel a fájl kiválasztható.

```
<form action="action_page.php">
  <label for="myfile">Select a file:</label><br>
  <input type="file" id="myfile" name="myfile"><br><br>
  <input type="submit" value="Submit">
</form>
```

A hidden típus

A hidden típus segítségével rejtett beviteli mezőt valósíthatunk meg, amely a felhasználó számára nem látható.
A rejtett mező lehetővé teszi a webfejlesztők számára, hogy olyan adatokat adjanak meg, amelyeket a felhasználók nem láthatnak vagy nem módosíthatnak az űrlap elküldésekor.

```
<form action="action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br><br>
  <input type="hidden" id="userId" name="userId" value="3487">
  <input type="submit" value="Submit">
</form>
```

A number típus

A number típus segítségével egy szám bevitelére alkalmas mezőt valósíthatunk meg. Korlátozásokat is beállíthatunk arra vonatkozóan, hogy milyen számok fogadhatók el.

```
<form action="action_page.php">
  <label for="quantity">Quantity (between 1 and 5):</label><br>
  <input type="number" id="quantity" name="quantity" min="1" max="5"><br><br>
  <input type="submit" value="Submit">
</form>
```

A step paraméter segítségével lépésköz is beállítható.

```
<form action="action_page.php">
  <label for="quantity">Quantity:</label><br>
  <input type="number" id="quantity" name="quantity" min="0" max="100" step="10"
value="30"><br><br>
  <input type="submit" value="Submit">
</form>
```

A range típus

A range típus segítségével csúszka valósítható meg. Az alapértelmezett tartomány 0 és 100 között van. Korlátozások állíthatók be arra vonatkozóan, hogy milyen számok fogadhatók el a min, a max és a step paraméterekkel.

```
<form action="action_page.php" method="get">
  <label for="vol">Volume (between 0 and 50):</label><br>
  <input type="range" id="vol" name="vol" min="0" max="50" step="10"><br><br>
  <input type="submit" value="Submit">
</form>
```

A tel típus

Telefonszám bekéréséhez használható beviteli mező.
A placeholder paraméter segítségével helyőrző jeleníthető meg a beviteli mezőben.
A pattern paraméter segítségével a formátum adható meg.
A required paraméter biztosítja, hogy a mezőt kötelező legyen kitölteni.

```
<form action="action_page.php">
  <label for="phone">Enter a phone number:</label><br><br>
  <input type="tel" id="phone" name="phone" placeholder="123-45-678" pattern="[0-9]{3}-
[0-9]{2}-[0-9]{3}" required><br><br>
  <input type="submit" value="Submit">
</form>
```

A time típus

Idő kiválasztásához használható beviteli mező. Böngészőtől függően egy időválasztó ablak jelenik meg.

```
<form action="action_page.php">
  <label for="time">Select a time:</label><br>
  <input type="time" id="time" name="time"><br><br>
  <input type="submit" value="Submit">
</form>
```

Az url típus

URL bekéréséhez használható beviteli mező. A böngésző támogatásától függően az url mező az űrlap elküldésekor automatikusan validálható.

```
<form action="action_page.php">
  <label for="homepage">Add your homepage:</label><br>
  <input type="url" id="homepage" name="homepage"><br><br>
  <input type="submit" value="Submit">
</form>
```

<u>További input paraméterek</u>

A readonly paraméter

A readonly paraméter használata esetén a beviteli mező csak olvasható lesz. A csak olvasható beviteli mező nem módosítható. A felhasználó azonban kimásolhatja belőle a szöveget. Az űrlap elküldésekor az írásvédett beviteli mező értéke elküldésre kerül.

```
<form action="action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John" readonly><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

A disabled paraméter

A disabled paraméter használata esetén a beviteli mező letiltásra kerül és nem használható. A letiltott beviteli mező értéke az űrlap elküldésekor nem kerül elküldésre.

```
<form action="action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="John" disabled><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>
```

A size paraméter

A size paraméter egy beviteli mező látható szélességét adja meg karakterben. Az alapértelmezett méret 20. A size paraméter a következő beviteli mező típusokkal működik: text, password, tel, url, email.

```
<form action="action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" size="50"><br><br>
  <label for="pin">PIN:</label><br>
  <input type="text" id="pin" name="pin" size="4"><br><br>
  <input type="submit" value="Submit">
</form>
```

A maxlength paraméter

A maxlength paraméter a beviteli mezőben megengedett karakterek maximális számát adja meg.

```
<form action="action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" size="50"><br><br>
  <label for="pin">PIN:</label><br>
  <input type="text" id="pin" name="pin" maxlength="4" size="4"><br><br>
  <input type="submit" value="Submit">
</form>
```

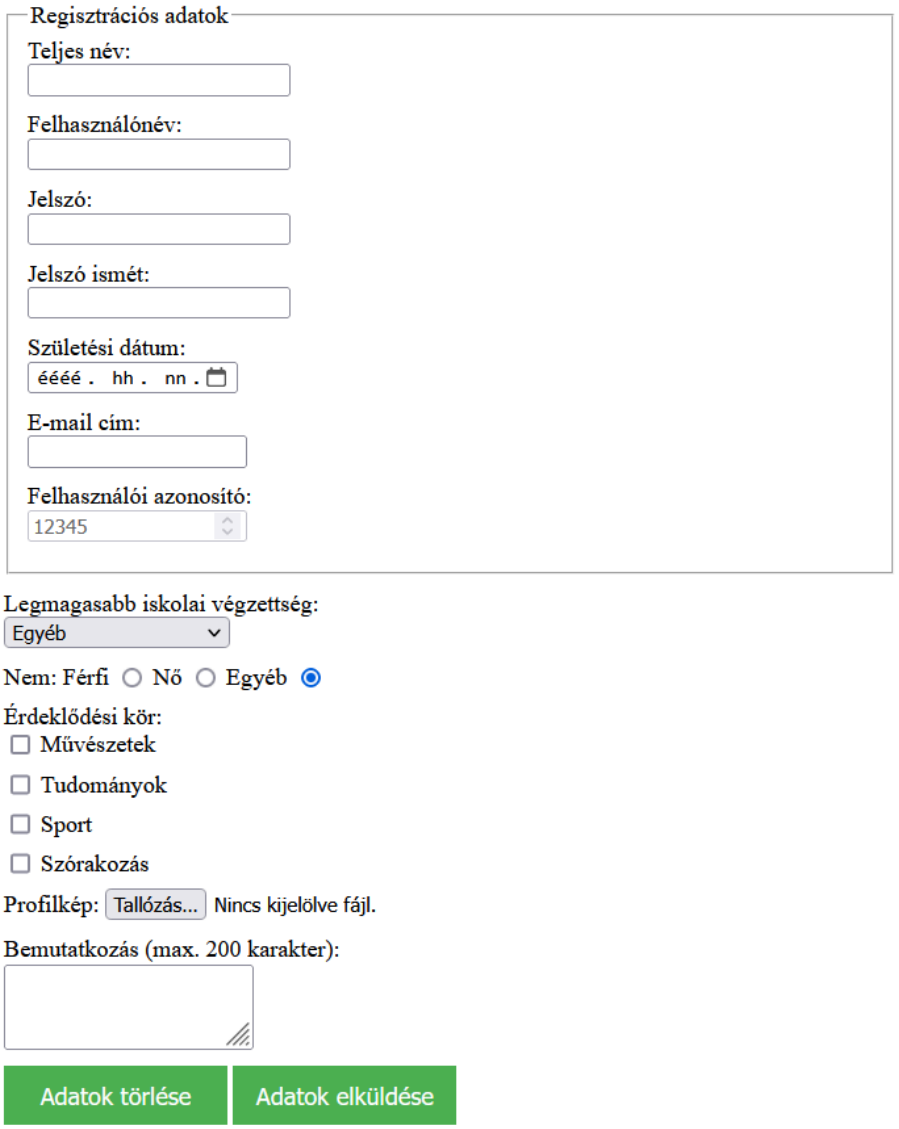
Az autofocus paraméter

Az autofocus paraméter használatával a beviteli mező automatikusan fókuszba kerül az oldal betöltésekor.

```
<form action="action_page.php">
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" autofocus><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname"><br><br>
  <input type="submit" value="Submit">
</form>
```

Gyakorlati feladat (Microsoft Teams 20230704_1.pdf)

Készítsük el az alábbi képen látható regisztrációs űrlapot!



CSS Flexbox

Mi az a Flexbox?

A CSS Flexbox egy nagyon praktikus eleme a CSS-nek, segítségével könnyedén lehet pozicionálni a képernyőn a különféle elemeket. Modern lehetőség az elemek elrendezésének módosításához. Tulajdonképpen egy display érték, hasonlóan az inline-hoz vagy a block-hoz.
Segítségével az elemeket sorba vagy oszlopba rendezhetjük, meghatározhatjuk a fő irányokat, az elemek térközeit, illetve azt is, hogy az elemek új sorba csúszannak-e, amennyiben már nem férnének ki, vagy pedig az elemek mérete csökkenjen, de azok maradjanak egy sorban.

A CSS Flexbox-os elrendezés nagy mértékben megkönnyíti a reszponzív weblapok készítését.

CSS Flexbox használatakor egy szülőelemnek a display:flex tulajdonságot adjuk. Ekkor minden egyes elem, ami közvetlenül ezen belül található, ennek megfelelően fog rendeződni.
A display: flex beállítással létrehozhatunk egy rugalmas konténert, amelynek mérete a rendelkezésre álló képernyőmérethez igazodik.
A gyermekelemei:

- elrendezhetők bármilyen irányban (jobbra, balra, lefelé, felfelé)
- elhelyezhetők egy vagy több sorban (oszlopban)
- a rugalmas konténerhez és egymáshoz képest is többféleképpen igazíthatók
- a rugalmas konténer méretének változásakor automatikusan változtathatják méretüket.

```html
<div class="flex-container">
    <div>Elem 1</div>
    <div>Elem 2</div>
    <div>Elem 3</div>
</div>
```

```css
*{
    margin: 0;
    padding: 0;
}

.flex-container{
    background-color: orange;
    display: flex;
}

.flex-container div{
    background-color: lightgray;
    margin: 10px;
    padding: 20px;
}
```

Flex-direction

A flex-direction tulajdonság azt adja meg, hogy az elemeknek a fő rendezési iránya mi legyen. Ez lehet sor vagy oszlop, pontosan négy különböző értéket definiálhatunk:

- row
- row-reverse
- column
- column-reverse

A row érték az alapértelmezett, ha nem definiáljuk ezt a tulajdonságot, akkor automatikusan ez fog érvényesülni. Az elemek sorrendje a HTML kód szerinti lesz. A row-reverse érték esetén az elemek sorrendje a HTML kódban megadott sorrend fordítottja lesz és az elemek rendezése is jobbról kezdődik.
A column érték esetén az elemeket oszlopba rendezzük, az elemek sorrendje a HTML kód szerinti lesz. Amennyiben a column-reverse értéket használjuk, akkor az elemek sorrendje a HTML kódban megadott sorrend fordítottja lesz és az elemek rendezése is alulról kezdődik.

```
.flex-container{
    background-color: orange;
    display: flex;
    flex-direction: row;
}
```

Flex-wrap

A flex-wrap tulajdonság azt hivatott meghatározni, hogy az elemeink abban az esetben, ha az aktuális képernyőméret nem engedné, új sorba / oszlopba rendeződjenek-e vagy sem. Alapvetően három értéket adhatunk neki:

- wrap
- nowrap
- wrap-reverse

A wrap azt jelenti, hogy az elemek rendeződjenek új sorba / oszlopba, amennyiben azok eredeti mérete szerint nem férnének el. A nowrap ezzel ellentétben nem töri meg az elemeket, azok összecsúszni igyekeznek a padding és margin értékek csökkentésével, egy idő után pedig még ki is lóghatnak a szülőelemből. Ez az alapértelmezett beállítás.
A wrap-reverse szintén engedi az elemeket új sorba / oszlopba rendeződni, de az ellentétes oldalra rendezi azokat, sor esetén az alapsor felé, oszlop esetén pedig bal oldalra.

```
<div class="flex-container">
    <div>Elem 1</div>
    <div>Elem 2</div>
    <div>Elem 3</div>
    <div>Elem 4</div>
    <div>Elem 5</div>
    <div>Elem 6</div>
</div>
```

```
.flex-container{
    background-color: orange;
    display: flex;
    flex-direction: row;
    flex-wrap: wrap;
}
```

Az előző két tulajdonságot, a flex-directiont és a flex-wrap-et egyetlen tulajdonságként is definiálhatjuk a flex-flow segítségével. Ha például a flex-direction értéke row, a flex-wrap értéke pedig wrap, akkor így is megadhatjuk: flex-flow: row wrap;

```
.flex-container{
    background-color: orange;
    display: flex;
    flex-flow: row wrap;
}
```

Justify-content

A justify-content tulajdonság arra való, hogy meghatározzuk, hogy az elemeket hogyan szeretnénk a flex-direction-ön (fő irányon) belül rendezni. Az alábbi értékeket adhatjuk neki:

- center
- flex-start
- flex-end
- space-between
- space-around
- space-evenly

A center értékkel középre igazítjuk az elemeket. A flex-start az alapértelmezett, míg a flex-end a fő irány végétől rendezi az elemeket. A space-between az elemeket a lehető legtávolabb rendezi egymástól, míg a space-around a térközt egyenlően osztja el az elemek mindkét oldalán. A space-evenly egyforma méretű térközöket eredményez.

```
.flex-container{
    background-color: orange;
    display: flex;
    flex-flow: row wrap;
    justify-content: center;
}
```

Align-items

Az align-items tulajdonság segítségével a második tengely mentén igazíthatjuk az elemeket. Amennyiben a flex-direction értéke row, akkor függőlegesen, amennyiben a flex-direction értéke column, akkor vízszintesen. Az alábbi értékeket adhatjuk neki:

- stretch
- center
- flex-start
- flex-end

A stretch az alapértelmezett, amennyiben nem adunk meg az elemek számára méretbeli kiterjedést (szélesség, magasság), akkor automatikusan maximális méretűre állítja be ezeket.
A többi érték esetén az elemeket minimális méretűre állítja be, a flex-start az alapértelmezett, a center középre igazítja, a flex-end pedig a végére igazítja az elemeket.

```
.flex-container{
    background-color: orange;
    height: 400px;
    display: flex;
    flex-flow: row wrap;
    align-items: center;
}
```

Align-content

Amennyiben az elemeink több sorba rendeződnének, akkor is beállíthatjuk a megfelelő térközöket. Tulajdonképpen a justify-content tulajdonság értékeit használhatjuk, valamint a stretch értéket, ami az alapértelmezett. A stretch érték esetén, ha nem adtuk meg az elemek méretét, akkor azok a maximális méretet igyekeznek felvenni, kitöltve a teret.
Ez a beállítás csak akkor értelmezhető, ha a flex-wrap:wrap tulajdonság be van állítva.

```
<div class="flex-container">
    <div>Elem 1</div>
    <div>Elem 2</div>
    <div>Elem 3</div>
    <div>Elem 4</div>
    <div>Elem 5</div>
    <div>Elem 6</div>
    <div>Elem 7</div>
    <div>Elem 8</div>
    <div>Elem 9</div>
    <div>Elem 10</div>
    <div>Elem 11</div>
    <div>Elem 12</div>
</div>
```

```css
.flex-container{
    background-color: orange;
    height: 400px;
    display: flex;
    flex-flow: row wrap;
    align-content: center;
}
```

Gap

A gap tulajdonság szabályozza a rugalmas elemek közötti teret. Ez a távolság csak az elemek között érvényes, a külső széleken nem.
Ha egy értéket adunk meg, akkor vízszintesen és függőleges is érvényesül. Két érték esetén először a sorok közötti távolságot adjuk meg, utána pedig az oszlopok közötti távolságot. Ezen értékeket a row-gap és a column-gap tulajdonságok segítségével is megadhatjuk.
Ha például egy justify-content: space-between beállítás miatt a távolság nagyobb, mint a gap értéke, akkor nem érvényesül, csak akkor ha kisebb, tulajdonképpen egy minimális távolságot adunk meg vele.
Nem kizárólag Flexboxhoz használható, más elrendezéseknél is működik.

```css
.flex-container{
    background-color: orange;
    height: 350px;
    display: flex;
    flex-flow: row wrap;
    align-items: flex-start;
    align-content: flex-start;
    gap: 10px 20px;

}
```

Az eddigi tulajdonságok a flex-container-re, vagyis a szülőelemre vonatkoztak. Az egyes elemeknek is lehet megadni speciális tulajdonságokat.

Flex-grow

Ennek a tulajdonságnak a segítségével állíthatjuk be, hogy az elemeink méretaránya hogyan alakuljon. A megadott számok arányában osztja fel a teljes szélességet, valamint a kijelzőméret növekedésekor az egyes elemek a többihez képest ilyen mértékben nyúlnak meg. Ha egy gyermekelemet nagyobb mértékben szeretnénk nyújtani a többinél, akkor nagyobb számértéket adunk meg, mint a többinél. Ha kisebb mértékben, akkor kisebbet. Nincs egyenes arányosság a számérték és a nyújtás mértéke között. A böngésző figyelembe veszi a gyermekelemek tartalmát is.

```html
<div class="flex-container">
    <div class="elem_1">Elem 1</div>
    <div class="elem_2">Elem 2</div>
    <div class="elem_3">Elem 3</div>
</div>
```

```css
.elem_1{
    flex-grow: 1;
}

.elem_2{
    flex-grow: 2;
}

.elem_3{
    flex-grow: 1;
}
```

Flex-shrink

A flex-shrink tulajdonság segítségével megadhatjuk, hogy egy rugalmas elem mennyivel zsugorodik a többi rugalmas elemhez képest. Az alapértelmezett érték 1.

```html
<div class="flex-container">
    <div class="elem_1">Elem 1</div>
    <div class="elem_2">Elem 2</div>
    <div class="elem_3">Elem 3</div>
    <div class="elem_4">Elem 4</div>
    <div class="elem_5">Elem 5</div>
    <div class="elem_6">Elem 6</div>
    <div class="elem_7">Elem 7</div>
    <div class="elem_8">Elem 8</div>
    <div class="elem_9">Elem 9</div>
    <div class="elem_10">Elem 10</div>
    <div class="elem_11">Elem 11</div>
    <div class="elem_12">Elem 12</div>
</div>
```

```css
.flex-container{
    background-color: orange;
    display: flex;
}

.flex-container div{
    background-color: lightgray;
    margin: 10px;
    padding: 20px;
}

.elem_3{
    flex-shrink: 0;
}
```

Flex-basis

A flex-basis tulajdonság egy rugalmas elem kezdeti hosszát adja meg.

```css
.elem_3{
    flex-basis: 200px;
}
```

Flex

A flex tulajdonság a flex-grow, a flex-shrink és a flex-basis tulajdonságok összevont (shorthand) megadási módja.

```css
.elem_3{
    flex: 0 0 200px;
}
```

Order

Az order segítségével tetszőleges sorrendbe állíthatjuk az elemeket. A flex-direction és a HTML kód határozza meg az alapvető sorrendet, ezt írhatjuk felül. Minél kisebb értéket adunk meg, annál előrébb, minél nagyobbat, annál hátrébb kerül az adott elem. Nem szükséges folyamatosan növekvő számokat használni.

```css
.elem_1{
    order: 2;
}

.elem_2{
    order: 3;
}

.elem_3{
    order: 1;
}
```

Align-self

Az align-self tulajdonság segítségével egy adott elem igazítását adhatjuk meg. Felülírja a tároló align-items tulajdonsága által beállított alapértelmezett igazítást.

```html
<div class="flex-container">
    <div class="elem_1">Elem 1</div>
    <div class="elem_2">Elem 2</div>
    <div class="elem_3">Elem 3</div>
    <div class="elem_4">Elem 4</div>
    <div class="elem_5">Elem 5</div>
    <div class="elem_6">Elem 6</div>
</div>
```

```css
.flex-container{
    background-color: orange;
    height: 200px;
    display: flex;
    align-items: center;
}

.flex-container div{
    background-color: lightgray;
    margin: 10px;
    padding: 20px;
}

.elem_2{
    align-self: flex-start;
}

.elem_5{
    align-self: flex-end;
}
```

Reszponzív Flexbox

Médialekérdezések segítségével különböző elrendezéseket hozhatunk létre a különböző képernyőméretekhez és eszközökhöz.

Például kétoszlopos elrendezést hozunk létre a nagyobb képernyőméretekhez és egyoszlopos elrendezést kis képernyőméretekhez (telefon, tablet) és módosítjuk az irányt sorról oszlopra. A töréspont legyen 800 pixel.

```html
<div class="flex-container">
    <div class="elem_1">Elem 1</div>
    <div class="elem_2">Elem 2</div>
</div>
```

```css
.flex-container{
    display: flex;
    flex-direction: row;
    text-align: center;
}

.elem_1{
    background-color: aqua;
    padding: 10px;
    flex: 50%;
}

.elem_2{
    background-color: aquamarine;
    padding: 10px;
    flex: 50%;
}

@media (max-width: 800px){
    .flex-container{
        flex-direction: column;
    }
}
```

Reszponzív képgaléria Flexbox használatával

```
<div class="row">
    <div class="column">
        <img src="butterfly.jpg">
        <img src="boat.jpg">
        <img src="animals.png">
        <img src="silver-gull.jpg">
        <img src="imperial-coat.jpg">
    </div>
    <div class="column">
        <img src="island.jpg">
        <img src="large-leaved-lupine.jpg">
        <img src="meadow.jpg">
        <img src="meadowsweet.jpg">
        <img src="desert.jpg">
    </div>
    <div class="column">
        <img src="mountain.jpg">
        <img src="mountain2.jpg">
        <img src="naxos.jpg">
        <img src="paraglider.jpg">
        <img src="flower.jpg">
    </div>
```

```
    <div class="column">
        <img src="st-stephens-basilica.jpg">
        <img src="theme-park.jpg">
        <img src="town-hall-tower.jpg">
        <img src="trees.jpg">
        <img src="windmills.jpg">
    </div>
</div>
```

```css
*{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

img{
    width: 100%;
}

.row{
    display: flex;
    flex-wrap: wrap;
    padding: 0 5px;
}

.column{
    flex: 25%;
    max-width: 25%;
    padding: 0 5px;
}
```

```css
.column img{
    margin-top: 10px;
    vertical-align: middle;
}

@media (max-width: 1000px){
    .column{
        flex: 50%;
        max-width: 50%;
    }
}

@media (max-width: 500px){
    .column{
        flex: 100%;
        max-width: 100%;
    }
}
```

Menü (nagyobb kijelzőn jobbra igazított, kisebb kijelzőn középre igazított, kis kijelzőn pedig egyoszlopos)

```html
<ul class="navigation">
    <li><a href="#">Home</a></li>
    <li><a href="#">About</a></li>
    <li><a href="#">Products</a></li>
    <li><a href="#">Contact</a></li>
</ul>
```

```css
*{
    margin: 0;
    padding: 0;
}

.navigation {
    display: flex;
    flex-flow: row wrap;
    justify-content: flex-end;
    list-style: none;
    margin: 0;
    background: deepskyblue;
}

.navigation a {
    text-decoration: none;
    display: block;
    padding: 1em;
    color: white;
}

.navigation a:hover {
    background: #1565C0;
}
```

```css
@media all and (max-width: 800px) {
    .navigation {
        justify-content: space-around;
    }
}

@media all and (max-width: 600px) {
    .navigation {
        flex-flow: column wrap;
        padding: 0;
    }
    .navigation a {
        text-align: center;
        padding: 10px;
        border-top: 1px solid rgba(255, 255, 255,0.3);
        border-bottom: 1px solid rgba(0, 0, 0, 0.1);
    }
    .navigation li:last-of-type a {
        border-bottom: none;
    }
}
```

Oldalszerkezet

```html
<div class="container">
    <header class="header">Header</header>
    <article class="main">
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Odio
        quos debitis quaerat, deserunt non tempore at repudiandae, excepturi
        quod minus quisquam fugit distinctio veritatis velit doloremque
        exercitationem officia, quam delectus!</p>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Odio
        quos debitis quaerat, deserunt non tempore at repudiandae, excepturi
        quod minus quisquam fugit distinctio veritatis velit doloremque
        exercitationem officia, quam delectus!</p>
        <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Odio
        quos debitis quaerat, deserunt non tempore at repudiandae, excepturi
        quod minus quisquam fugit distinctio veritatis velit doloremque
        exercitationem officia, quam delectus!</p>
    </article>
    <aside class="aside aside-1">Aside 1</aside>
    <aside class="aside aside-2">Aside 2</aside>
    <footer class="footer">Footer</footer>
</div>
```

```css
*{
    margin: 0;
    padding: 0;
}

.container {
    display: flex;
    flex-flow: row wrap;
    font-weight: bold;
    text-align: center;
}

.container > * {
    padding: 10px;
    flex: 1 100%;
}
```

```css
.header {
    background: tomato;
}

.footer {
    background: lightgreen;
}

.main {
    background: deepskyblue;
}

.aside-1 {
    background: gold;
}

.aside-2 {
    background: hotpink;
}
```

```css
@media all and (min-width: 600px) {
    .aside { flex: 1 0 0; }
}

@media all and (min-width: 800px) {
    .main    { flex: 3 0px; }
    .aside-1 { order: 1; }
    .main    { order: 2; }
    .aside-2 { order: 3; }
    .footer  { order: 4; }
}
```

Linkek:
https://css-tricks.com/snippets/css/a-guide-to-flexbox/
https://www.w3schools.com/css/css3_flexbox.asp
https://flexbox.help/
https://flexboxfroggy.com/#hu

CSS Grid

Mi az a CSS Grid?

A weboldalon megjelenő elemek elrendezésére egy megoldás a CSS Grid. Segítségével egy tetszőleges rácsszerkezetet hozhatunk létre a weboldalunkon, amely sorokból és oszlopokból áll. A sorok és oszlopok által alkotott cellákba tehetjük a különböző elemeket, amelyeknek tetszőleges egyéni stílust adhatunk. A CSS Grid a reszponzív weboldal létrehozásában is hasznos eszköz.

A CSS Grid úgy jön létre, hogy egy szülőelem (pl. egy div) számára beállítjuk a display: grid tulajdonságot. A rácsszerkezet létrejöttéhez egy másik tulajdonságra is szükség van, ez a grid-template-columns, amely megadja, hogy az oszlopok (és így a cellák) milyen szélesek legyenek.
A Grid container tehát egy tároló, az összes rácselem közvetlen szülőeleme. A Grid elemek a container gyermekelemei, vagyis közvetlen leszármazottai, tehát a bennük lévő elemek már nem számítanak annak.
A Grid vonalak vagy elválasztó vonalak lehetnek függőlegesek (oszlop rácsvonalak) és vízszintesek (sor rácsvonalak), és egy oszlop vagy sor mindkét oldalán megtalálhatók. Tehát egy 3 oszlopos rács esetén 4 oszlop rácsvonal van.
Két szomszédos sor rácsvonal és két szomszédos oszlop rácsvonal közötti térköz a Grid cella. Például az első sor második cellája az 1. és 2. sor rácsvonal és a 2. és 3. oszlop rácsvonal között helyezkedik el.
Grid tracknek (sávnak) két szomszédos rácsvonal közötti teret nevezünk, ezek a rács oszlopai és sorai.
Grid area (terület) a négy rácsvonallal körülvett terület, amely tetszőleges számú cellából állhat.
A Grid tulajdonságok kapcsolódhatnak a szülőelemhez (container) és a gyermekelemekhez.

```html
<div class="container-grid">
    <div class="elem_1">Elem 1</div>
    <div class="elem_2">Elem 2</div>
    <div class="elem_3">Elem 3</div>
    <div class="elem_4">Elem 4</div>
</div>
```

```css
.container-grid{
    display: grid;
    grid-template-columns: 20% 20% 20% 20%;
    height: 500px;
    justify-content: center;
    align-content: center;
}

.container-grid div{
    padding: 20px;
}
```

```css
.elem_1{
    background-color: aqua;
}

.elem_2{
    background-color: aquamarine;
}

.elem_3{
    background-color: bisque;
}

.elem_4{
    background-color: blanchedalmond;
}
```

Grid-template-columns

A tulajdonság azt határozza meg, hogy milyen szélesek legyenek a rácsunk oszlopai és hány oszlop kerülhet egy sorba. Ha egy elem már nem fér ki az adott sorba, akkor új sorba kerül. Az értékeket szóközökkel kell elválasztani egymástól.

```html
<div class="container-grid">
    <div class="elem_1">Elem 1</div>
    <div class="elem_2">Elem 2</div>
    <div class="elem_3">Elem 3</div>
    <div class="elem_4">Elem 4</div>
    <div class="elem_5">Elem 5</div>
    <div class="elem_6">Elem 6</div>
</div>
```

```css
.elem_5{
    background-color: cadetblue;
}

.elem_6{
    background-color: darksalmon;
}
```

Az oszlop szélesség megadható:

- százalékban
- pixelben
- vw-ben (a képernyő szélességének századrésze)

Ismételt megadás: repeat(oszlopok darabszáma, oszlop szélessége)

```css
.container-grid{
    display: grid;
    grid-template-columns: repeat(4, 25%);
    height: 500px;
    justify-content: center;
    align-content: center;
}
```

Kitöltéses megadás: repeat (auto-fill, oszlop szélessége) A megadott méretű oszlopból annyit tesz egy sorba, amennyi elfér.

```css
.container-grid{
    display: grid;
    grid-template-columns: repeat(auto-fill, 400px);
    height: 500px;
    justify-content: center;
    align-content: center;
}
```

Arányos felosztás: a mértékegység az fr (fraction). Minden fr egység a rendelkezésre álló hely egy részét foglalja le. Például két, 1fr és 3fr méretűre állított elem esetén a rendelkezésre álló hely négy egyforma részre lesz osztva, ahol az első elem 1/4, a második elem pedig 3/4 részre terjed ki.

```css
.container-grid{
    display: grid;
    grid-template-columns: 2fr 1fr;
    height: 500px;
    justify-content: center;
    align-content: center;
}
```

A rácsvonalakhoz oszlopok esetén balról jobbra haladva 1-től kezdve növekvő számok, jobbról balra haladva -1-től kezdődő csökkenő számok vannak rendelve. Sorok esetén fentről lefelé haladva 1-től kezdődően növekvő számok, lentről felfelé haladva -1-től kezdődően csökkenő számok vannak rendelve.
A rácsvonalakra névvel is hivatkozhatunk. Oszlopok esetén például: first, line2, line3, col4-start, five, end. Sorok esetén például: row1-start, row1-end, third-line, last-line. Egy vonalnak több neve is lehet, például, a row1-end és a row2-start ugyanarra a vonalra utal. A neveket szögletes zárójelbe kell tenni a kódban.

```html
<div class="container-grid">
    <div class="elem_1">Elem 1</div>
    <div class="elem_2">Elem 2</div>
    <div class="elem_3">Elem 3</div>
    <div class="elem_4">Elem 4</div>
    <div class="elem_5">Elem 5</div>
    <div class="elem_6">Elem 6</div>
    <div class="elem_7">Elem 7</div>
    <div class="elem_8">Elem 8</div>
    <div class="elem_9">Elem 9</div>
    <div class="elem_10">Elem 10</div>
    <div class="elem_11">Elem 11</div>
    <div class="elem_12">Elem 12</div>
</div>
```

```css
.elem_7{
    background-color: goldenrod;
}

.elem_8{
    background-color: greenyellow;
}

.elem_9{
    background-color: lightblue;
}
```

```css
.elem_10{
    background-color: lightgreen;
}

.elem_11{
    background-color: orchid;
}

.elem_12{
    background-color: teal;
}
```

```css
.container-grid{
    display: grid;
    grid-template-columns: [first] 100px [line2] 200px [col3-start] 200px [four]
    100px [end];
    grid-template-rows: [row1-start] 25% [row1-end] 100px [second-line] 200px
    [last-line];
    height: 500px;
    justify-content: center;
    align-content: center;
}
```

Grid-template-rows

A tulajdonság azt határozza meg, hogy milyen magasak legyenek az egyes sorok.

```
.container-grid{
    display: grid;
    grid-template-columns: 20% 20% 20% 20%;
    grid-template-rows: 100px 200px;
    height: 500px;
    justify-content: center;
    align-content: center;
}
```

Justify-content

A justify-content tulajdonság segítségével igazíthatjuk a teljes rácsszerkezetet a containeren belül. Értékei lehetnek:

- start (a rácsszerkezetet a tároló bal oldalára igazítja)
- end (a rácsszerkezetet a tároló jobb oldalára igazítja)
- center (a rácsszerkezetet a tároló közepére igazítja)
- space-between (egyenletesen osztja el a helyet az oszlopok között, az első oszlop előtt és az utolsó oszlop után nem hagy ki helyet)
- space-around (egyforma méretű helyet hagy ki az oszlopok előtt és után)
- space-evenly (egyforma méretű helyet hagy ki mindenütt)

Align-content

Az align-content tulajdonság segítségével igazíthatjuk függőlegesen a teljes rácsszerkezetet a containeren belül. Értékei megegyeznek a justify-content értékeivel.

Place-content

Az align-content és a justify-content beállítás összevonható a place-content tulajdonság segítségével. Az első érték az align-content, a második pedig a justify-content. Ha egyforma értéket szeretnénk mindkettőnek beállítani, elég egyszer megadni.

```
.container-grid{
    display: grid;
    grid-template-columns: 20% 20% 20% 20%;
    grid-template-rows: 100px 100px;
    height: 500px;
    place-content: center;
}
```

Justify-items

A cellán belül igazíthatjuk az elemeket a justify-items tulajdonság segítségével. Értékei lehetnek:

- start (az elemeket a cella bal oldalához igazítja)
- end (az elemeket a cella jobb oldalához igazítja)
- center (az elemeket a cella közepéhez igazítja)
- stretch (az elem a cella teljes szélességét kitölti, ez az alapértelmezett)

```
.container-grid{
    display: grid;
    grid-template-columns: 20% 20% 20% 20%;
    justify-items: center;
    height: 500px;
    justify-content: center;
    align-content: center;
}
```

Align-items

A cellán belül függőlegesen igazíthatjuk az elemeket az align-items tulajdonság segítségével. Értékei lehetnek:

- stretch (az elem a cella teljes magasságát kitölti, ez az alapértelmezett)
- start (az elemeket a cella felső oldalához igazítja)
- end (az elemeket a cella alsó oldalához igazítja)
- center (az elemeket a cella közepéhez igazítja függőlegesen)

```
.container-grid{
    display: grid;
    grid-template-columns: 20% 20% 20% 20%;
    grid-template-rows: 100px 100px;
    align-items: center;
    height: 500px;
    justify-content: center;
    align-content: center;
}
```

Place-items

Az align-items és a justify-items beállítás összevonható a place-items tulajdonság segítségével. Az első érték az align-items, a második pedig a justify-items. Ha egyforma értéket szeretnénk mindkettőnek beállítani, elég egyszer megadni.

```css
.container-grid{
    display: grid;
    grid-template-columns: 20% 20% 20% 20%;
    grid-template-rows: 100px 100px;
    place-items: center;
    height: 500px;
    justify-content: center;
    align-content: center;
}
```

Grid-column-gap, grid-row-gap

Ezen tulajdonságok segítségével az oszlopok (grid-column-gap vagy column-gap) és sorok (grid-row-gap vagy row-gap) közötti távolságot adhatjuk meg vagyis a rácsvonalak méretét. Használhatjuk bármelyik CSS mértékegységet. Amennyiben egyforma sor és oszlopközt szeretnénk beállítani vagy összevontan szeretnénk megadni, megtehetjük a grid-gap (gap) tulajdonsággal.
Ezek a távolságok csak a sorok és oszlopok között jelennek meg, a külső széleken nem.

```css
.container-grid{
    display: grid;
    grid-template-columns: 20% 20% 20% 20%;
    grid-column-gap: 20px;
    grid-row-gap: 10px;
    height: 500px;
    justify-content: center;
    align-content: center;
}
```

Cellák tartalmának igazítása

A cellatartalmakat igazíthatjuk és azok méretét a minimálisra vehetjük, ha a Grid elemek align-self és justify-self tulajdonságait állítjuk be. Mindkét tulajdonság értékei lehetnek:

- start
- end
- center
- stretch

Összevontan is megadható a két beállítás a place-self tulajdonság segítségével. Az első érték az align-self, a második pedig a justify-self.

```css
.container-grid div{
    padding: 20px;
    align-self: center;
    justify-self: center;
}
```

Cellák egyesítése

A cellák egyesíthetők a grid-column-start és grid-column-end, valamint a grid-row-start és grid-row-end tulajdonságokkal. Ezeket a tulajdonságokat nem a Grid contaneirnek hanem a Grid elemeknek kell megadni. A számozás úgy értendő, hogy a rácsszerkezet bal oldali határvonala és a felső határvonala számít egynek.

```
.elem_1{
    background-color: aqua;
    grid-column-start: 1;
    grid-column-end: 3;
    grid-row-start: 1;
    grid-row-end: 3;
}
```

Összevont megadásra a grid-column és a grid-row tulajdonságok szolgálnak.

```
.elem_1{
    background-color: aqua;
    grid-column: 1 / 3;
    grid-row: 1 / 3;
}
```

Szintén összevont megadásra szolgál a grid-area tulajdonság (sor, oszlop, sor, oszlop).

```
.elem_1{
    background-color: aqua;
    grid-area: 1 / 1 / 3 / 3;
}
```

Összevont megadás esetén a függőleges és vízszintes második rácsvonal helyett az összevont cellák darabszáma is megadható (pl. span 2).

Grid-template-areas

A rácsszerkezeten belül az elemeket tudjuk elrendezni ezen tulajdonság segítségével. Idézőjelek segítségével sztringeket kell megadnunk.
Az egyes elemeknél a grid-area tulajdonsággal megadjuk, hogy az adott elemből mi lesz (pl. header). Értéke lehet:

- a grid-area tulajdonsággal megadott terület neve
- . (pont) - üres cellát jelöl
- none - nincs meghatározva rácsterület

Fontos, hogy minden sorban ugyanannyi cellát adjunk meg.

```css
.container-grid{
    display: grid;
    grid-template-columns: 20% 20% 20% 20%;
    grid-template-areas:
        "header header header header"
        "main-left main-left main-left main-right"
        "second-left second-right second-right second-right"
        "footer footer footer footer";
    column-gap: 20px;
    row-gap: 10px;
    height: 500px;
    justify-content: center;
    align-content: center;
}
```

```css
.elem_1{
    background-color: aqua;
    grid-area: header;
}

.elem_2{
    background-color: aquamarine;
    grid-area: main-left;
}

.elem_3{
    background-color: bisque;
    grid-area: main-right;
}
```

```css
.elem_4{
    background-color: blanchedalmond;
    grid-area: second-left;
}

.elem_5{
    background-color: cadetblue;
    grid-area: second-right;
}

.elem_6{
    background-color: darksalmon;
    grid-area: footer;
}
```

Üres cella

```
.container-grid{
    display: grid;
    grid-template-columns: 20% 20% 20% 20%;
    grid-template-areas:
        "header header header header"
        "main-left main-left main-left main-right"
        "second-left . second-right second-right"
        "footer footer footer footer";
    height: 500px;
    justify-content: center;
    align-content: center;
}
```

Reszponzív rácsszerkezet

Médialekérdezések segítségével különböző elrendezéseket hozhatunk létre a különböző képernyőméretekhez és eszközökhöz.

```
@media (max-width: 800px){
    .container-grid{
        grid-template-columns: 100%;
        grid-template-areas:
            "header" "header" "header" "header"
            "main-left" "main-left" "main-left" "main-right"
            "second-left" "second-right" "second-right" "second-right"
            "footer" "footer" "footer" "footer";
    }
}
```

Az elemek rendezése

A grid-area tulajdonság segítségével az elemeket áthelyezhetjük, ahova szeretnénk.

```
.container-grid{
    display: grid;
    grid-template-columns: 20% 20% 20%;
    height: 500px;
    justify-content: center;
    align-content: center;
}
```

```css
.elem_1{
    background-color: aqua;
    grid-area: 1 / 3 / 2 / 4;
}

.elem_2{
    background-color: aquamarine;
    grid-area: 2 / 3 / 3 / 4;
}

.elem_3{
    background-color: bisque;
    grid-area: 1 / 2 / 2 / 3;
}
```

```css
.elem_4{
    background-color: blanchedalmond;
    grid-area: 2 / 2 / 3 / 3;
}

.elem_5{
    background-color: cadetblue;
    grid-area: 1 / 1 / 2 / 2;
}

.elem_6{
    background-color: darksalmon;
    grid-area: 2 / 1 / 3 / 2;
}
```

Reszponzív CSS Grid

```html
<div class="container">
    <header>Header</header>
    <nav>Navbar</nav>
    <main>Main</main>
    <aside>Sidebar</aside>
    <div id="content1">Content1</div>
    <div id="content2">Content2</div>
    <div id="content3">Content3</div>
    <footer>Footer</footer>
</div>
```

```css
.container{
    display: grid;
    height: 100vh;
    grid-template-columns: 1fr 1fr 1fr 1fr;
    grid-template-rows: 1fr 0.5fr 3fr 3fr 1fr;
    grid-template-areas:
        "header header header header"
        "nav nav nav nav"
        "sidebar main main main"
        "sidebar content1 content2 content3"
        "footer footer footer footer";
    gap: 0.5em;
}
```

```css
.container *{
    border-radius: 5px;
    padding: 10px;
    font-weight: bold;
    text-transform: uppercase;
    font-size: 14px;
    color: #004d40;
    text-align: center;
}
```

```css
header{
    background-color: #a7ffeb;
    grid-area: header;
}

nav{
    background-color: #84ffff;
    grid-area: nav;
}

main{
    background-color: #18ffff;
    grid-area: main;
}

aside{
    background-color: #6fffd2;
    grid-area: sidebar;
}
```

```css
#content1{
    background-color: #64ffda;
    grid-area: content1;
}

#content2{
    background-color: #73ffba;
    grid-area: content2;
}

#content3{
    background-color: #1de9b6;
    grid-area: content3;
}

footer{
    background-color: #18ffd2;
    grid-area: footer;
}
```

```css
@media (max-width: 600px){
    .container{
        grid-template-columns: 1fr;
        grid-template-rows: 1fr 0.5fr 3fr 2fr 2fr 2fr 1fr 1fr;
        grid-template-areas:
            "header"
            "nav"
            "main"
            "content1"
            "content2"
            "content3"
            "sidebar"
            "footer";
    }
}
```

Linkek:
https://css-tricks.com/snippets/css/complete-guide-grid/
https://www.w3schools.com/css/css_grid.asp
https://learncssgrid.com/
https://cssgridgarden.com/#hu

CSS3 transzformációk

Elforgatás

A CSS3 lehetővé teszi a HTML elemek középpontjuk körül történő elforgatását a transform tulajdonság segítségével. A tulajdonság értékeként a rotate kulcsszót és az elforgatás szögét kell megadni, mértékegysége a deg.

```html
<div id="rotate_1" class="tarolo">
    <div class="elem">elforgatás</div>
</div>
<div id="rotate_2" class="tarolo">
    <div class="elem">elforgatás</div>
</div>
<div id="rotate_3" class="tarolo">
    <div class="elem">elforgatás</div>
</div>
```

```css
*{
    margin: 0;
    padding: 0;
}

.tarolo{
    margin-top: 50px;
    margin-left: 50px;
    width: 200px;
    height: 200px;
    border: 1px solid red;
}

.elem{
    width: 200px;
    height: 200px;
    background-color: lightgreen;
    border: 1px solid darkgreen;
    opacity: 0.8;
}
```

```css
#rotate_1 .elem{
    transform: rotate(45deg);
}

#rotate_2 .elem{
    transform: rotate(-45deg);
}

#rotate_3 .elem{
    transform: rotate(120deg);
}
```

Elforgatás az X tengely mentén

A transform tulajdonság segítségével a HTML elemek az X tengely körül is elforgathatók. A tulajdonság értékeként a rotateX kulcsszót és az elforgatás szögét kell megadni, mértékegysége a deg.

```css
.tarolo{
    margin-top: 50px;
    margin-left: 50px;
    width: 200px;
    height: 200px;
    border: 1px solid red;
    perspective: 500px;
    transform-style: preserve-3d;
}
```

```css
#rotate_1 .elem{
    transform: rotateX(45deg);
}

#rotate_2 .elem{
    transform: rotateX(-45deg);
}

#rotate_3 .elem{
    transform: rotateX(60deg);
}
```

Elforgatás az Y tengely mentén

A transform tulajdonság segítségével a HTML elemek az Y tengely körül is elforgathatók. A tulajdonság értékeként a rotateY kulcsszót és az elforgatás szögét kell megadni, mértékegysége a deg.

```css
#rotate_1 .elem{
    transform: rotateY(45deg);
}

#rotate_2 .elem{
    transform: rotateY(-45deg);
}

#rotate_3 .elem{
    transform: rotateY(60deg);
}
```

Elforgatás a 3D tengely mentén

A transform tulajdonság segítségével a HTML elemek a térben megadott tengely körül is elforgathatók. Értékei:

- x: a forgástengelyt jelölő vektor X koordinátája, amely lehet pozitív vagy negatív szám
- y: a forgástengelyt jelölő vektor Y koordinátája, amely lehet pozitív vagy negatív szám

- z: a forgástengelyt jelölő vektor Z koordinátája, amely lehet pozitív vagy negatív szám
- a: az elforgatás szöge, a pozitív szög az óramutató járásával megegyező, a negatív szög az óramutató járásával ellentétes forgást jelent

```
#rotate_1 .elem{
    transform: rotate3d(30, 30, 30, 30deg);
}
```

Vízszintes eltolás

A CSS3 lehetővé teszi a HTML elemek eltolását (helyzetének módosítását) x irányba (vízszintesen) a transform: translateX() tulajdonság segítségével. Pozitív érték esetén jobbra, negatív érték esetén balra tolja az elemet.

```
<div id="translate_1" class="tarolo">
    <div class="elem">eltolás</div>
</div>
<div id="translate_2" class="tarolo">
    <div class="elem">eltolás</div>
</div>
<div id="translate_3" class="tarolo">
    <div class="elem">eltolás</div>
</div>
```

```
#translate_1 .elem{
    transform: translateX(0);
}

#translate_2 .elem{
    transform: translateX(50px);
}

#translate_3 .elem{
    transform: translateX(-50px);
}
```

Függőleges eltolás

A CSS3 lehetővé teszi a HTML elemek eltolását (helyzetének módosítását) y irányba (függőlegesen) a transform: translateY() tulajdonság segítségével. Pozitív érték esetén lefelé, negatív érték esetén felfelé tolja az elemet.

```
#translate_1 .elem{
    transform: translateY(0);
}

#translate_2 .elem{
    transform: translateY(50px);
}

#translate_3 .elem{
    transform: translateY(-50px);
}
```

Közelítés-távolítás

A CSS3 lehetővé teszi a HTML elemek eltolását (helyzetének módosítását) z irányba (közelítés-távolítás) a transform: translateZ() tulajdonság segítségével. Pozitív érték esetén közelíti, negatív érték esetén távolítja az elemet.

```
#translate_1 .elem{
    transform: translateZ(0);
}

#translate_2 .elem{
    transform: translateZ(-100px);
}

#translate_3 .elem{
    transform: translateZ(100px);
}
```

3D eltolás

A translate3d() függvény segítségével egy lépésben megadhatók az eltolás x, y és z irányú összetevőinek értékei.

```
#translate_1 .elem{
    transform: translate3d(0);
}

#translate_2 .elem{
    transform: translate3d(50px, 50px, 100px);
}

#translate_3 .elem{
    transform: translate3d(-50px, -50px, -100px);
}
```

Vízszintes átméretezés

A CSS3 lehetővé teszi a HTML elemek átméretezését (skálázását). A transform: scaleX() tulajdonság segítségével vízszintesen nyújthatjuk vagy zsugoríthatjuk a HTML elemet. Az 1-nél kisebb érték csökkenti, az 1-nél nagyobb érték növeli az elem méretét.

```
<div id="scale_1" class="tarolo">
    <div class="elem">átméretezés</div>
</div>
<div id="scale_2" class="tarolo">
    <div class="elem">átméretezés</div>
</div>
<div id="scale_3" class="tarolo">
    <div class="elem">átméretezés</div>
</div>
```

```
#scale_1 .elem{
    transform: scaleX(0);
}

#scale_2 .elem{
    transform: scaleX(0.8);
}

#scale_3 .elem{
    transform: scaleX(1.2);
}
```

Függőleges átméretezés

A CSS3 lehetővé teszi a HTML elemek átméretezését (skálázását). A transform: scaleY() tulajdonság segítségével függőlegesen nyújthatjuk vagy zsugoríthatjuk a HTML elemet. Az 1-nél kisebb érték csökkenti, az 1-nél nagyobb érték növeli az elem méretét.

```
#scale_1 .elem{
    transform: scaleY(0);
}

#scale_2 .elem{
    transform: scaleY(0.8);
}

#scale_3 .elem{
    transform: scaleY(1.2);
}
```

Vízszintes és függőleges átméretezés

A scale() függvény segítségével vízszintesen és függőlegesen is nyújthatjuk vagy zsugoríthatjuk a HTML elemet. Az 1-nél kisebb érték csökkenti, az 1-nél nagyobb érték növeli az elem méretét. Az első érték a vízszintes átméretezés, a második érték a függőleges átméretezés.

```
#scale_1 .elem{
    transform: scale(0);
}

#scale_2 .elem{
    transform: scale(1.2, 0.8);
}

#scale_3 .elem{
    transform: scale(0.8, 1.2);
}
```

Vízszintes döntés

A CSS3 lehetővé teszi a HTML elemek megdöntését. A transform: skewX() tulajdonság segítségével vízszintesen dönthetjük a HTML elemet. Értékként a döntés szögét kell megadni, mértékegység a deg.

```
<div id="skew_1" class="tarolo">
    <div class="elem">döntés</div>
</div>
<div id="skew_2" class="tarolo">
    <div class="elem">döntés</div>
</div>
<div id="skew_3" class="tarolo">
    <div class="elem">döntés</div>
</div>
```

```
#skew_1 .elem{
    transform: skewX(0);
}

#skew_2 .elem{
    transform: skewX(30deg);
}

#skew_3 .elem{
    transform: skewX(-30deg);
}
```

Függőleges döntés

A CSS3 lehetővé teszi a HTML elemek megdöntését. A transform: skewY() tulajdonság segítségével függőlegesen dönthetjük a HTML elemet. Értékként a döntés szögét kell megadni, mértékegység a deg.

```
#skew_1 .elem{
    transform: skewY(0);
}

#skew_2 .elem{
    transform: skewY(30deg);
}

#skew_3 .elem{
    transform: skewY(-30deg);
}
```

Vízszintes és függőleges döntés

A vízszintes és függőleges irányú megdöntés szögei egy lépésben megadhatók a skew() függvénnyel deg-ben. Az első érték a vízszintes, a második érték a függőleges döntés szöge.

```
#skew_1 .elem{
    transform: skew(0);
}

#skew_2 .elem{
    transform: skew(30deg, 30deg);
}

#skew_3 .elem{
    transform: skew(-30deg, -30deg);
}
```

A transzformáció középpontja

A transzformáció középpontja alapértelmezetten a HTML elem középpontjával esik egybe, de ettől eltérhetünk a transform-origin tulajdonság segítségével.
A helyzet megadható százalékos értékben és hosszmértékben is.
Az értékek sorrendje: x és y.
Az alapértelmezett érték: transform-origin(50%, 50%)
Használható kulcsszavak:

- vízszintes irányban: left, center, right
- függőleges irányban: top, center, bottom

A térbeli transzformációknál meg kell adni a z értékét is, de ebben az esetben kulcsszavak nem használhatók.

```html
<div id="rotate_1" class="tarolo">
    <div class="elem">elforgatás</div>
</div>
<div id="rotate_2" class="tarolo">
    <div class="elem">elforgatás</div>
</div>
<div id="rotate_3" class="tarolo">
    <div class="elem">elforgatás</div>
</div>
```

```css
.elem{
    width: 200px;
    height: 200px;
    background-color: lightgreen;
    border: 1px solid darkgreen;
    opacity: 0.8;
    transform: rotate(45deg);
}

#rotate_1 .elem{
    transform-origin: right top;
}

#rotate_2 .elem{
    transform-origin: 20% 40%;
}

#rotate_3 .elem{
    transform-origin: left bottom;
}
```

CSS3 animációk

A CSS stílusbeállítások módosítása

A :hover és :active álosztály kijelölők felhasználásával módosíthatjuk a HTML elemek CSS beállításait abban az esetben, ha az egérkurzor az elem felett van, vagy a felhasználó a bal egérgombot megnyomja az egérkurzorral az elem felett állva.
A változás ilyenkor azonnal és minden átmenet nélkül életbe lép.
Ezeket az álosztályokat tipikusan beviteli elemeknél és linkeknél szokás használni, de akár képtárak esetén a kiválasztott kép kiemelésére vagy egy általános tároló elem tulajdonságainak módosítására is használhatjuk.

```html
<img src="boat.jpg" alt="boat" width="200">
```

```css
*{
    margin: 0;
    padding: 0;
}

img{
    margin: 20px;
}

img:hover{
    width: 400px;
}
```

A CSS3 animációs lehetőségei

Átmenetek

A CSS3 lehetővé teszi, hogy a stílusbeállítások változtatása fokozatosan menjen végbe. Korábban Flash animációk vagy JavaScript segítségével lehetett ilyen hatásokat megvalósítani. A CSS3-ban ennek eléréséhez használhatjuk a transition tulajdonságot. Ez lehetőséget ad átmenetek készítésére, ahol egy elem kezdeti CSS beállításai fokozatosan változnak az új értékekre. Az átmenet indulhat az oldal betöltődését követően is, de egy-egy esemény bekövetkezésekor is (pl. :hover).

```css
img{
    margin: 20px;
    transform: rotate(0);
    transition: 2s;

}

img:hover{
    width: 400px;
    transform: rotate(360deg);
}
```

Az egyes tulajdonság változásokhoz is megadható időtartam.

```html
<div></div>
```

```css
div{
    margin: 20px;
    width: 100px;
    height: 100px;
    background-color: red;
    transition: width 2s, height 4s;
}

div:hover{
    width: 300px;
    height: 300px;
}
```

Transition-property

A transition-property tulajdonság segítségével megadható, hogy mely CSS tulajdonság(ok) változásait kívánjuk folyamatos átmenetté alakítani.
Az átmenettel érintett tulajdonságok neveinek megadása:
kijelölő{
transition-property: tulajdonság, tulajdonság, …;
}
Vesszővel elválasztva több tulajdonság is megadható. Az all kulcsszó segítségével valamennyi jellemző változása folyamatossá tehető.
Az átmenet időtartamának beállítása:
kijelölő{
transition-duration: érték, érték, …;
}
Vesszővel elválasztva több érték is megadható. Az időtartam megadása kötelező, mértékegysége lehet s vagy ms.

```html
<div id="palya">
    <div class="labda1"><img src="labda.png" width="100" alt=""></div>
    <div class="labda2"><img src="labda.png" width="100" alt=""></div>
    <div class="labda3"><img src="labda.png" width="100" alt=""></div>
</div>
```

```css
#palya{
    width: 500px;
    background-color: darkgreen;
}

#palya div{
    padding: 20px 0;
}

#palya:hover img{
    transform: translateX(400px) rotate(360deg);
}
```

```css
.labda1 img{
    transition-property: transform;
    transition-duration: 1s;
}

.labda2 img{
    transition-property: transform;
    transition-duration: 3s;
}

.labda3 img{
    transition-property: transform;
    transition-duration: 5s;
}
```

CSS3 átmenetek változó sebességgel

A transition-timing-function tulajdonság segítségével adható meg a CSS tulajdonság(ok) változásainak karakterisztikája.
Az átmenet karakterisztikájának beállítása:
kjelölő{
    transition-timing-function: kulcsszó;
}
Vesszővel elválasztva több tulajdonság is megadható.
A választható karakterisztikák:

- linear: állandó sebesség
- ease: a változás eleje és vége lassú, közepén gyors
- ease-in: a változás eleje lassú, majd gyorsul
- ease-out: gyors változás, majd a végén lassul
- ease-in-out: a változás eleje és vége lassú, közepén gyors

```html
<div id="palya">
    <div class="labda1"><img src="labda.png" width="100" alt=""></div>
    <div class="labda2"><img src="labda.png" width="100" alt=""></div>
    <div class="labda3"><img src="labda.png" width="100" alt=""></div>
    <div class="labda4"><img src="labda.png" width="100" alt=""></div>
    <div class="labda5"><img src="labda.png" width="100" alt=""></div>
</div>
```

```css
.labda1 img{
    transition-property: transform;
    transition-duration: 3s;
    transition-timing-function: linear;

}

.labda2 img{
    transition-property: transform;
    transition-duration: 3s;
    transition-timing-function: ease;
}

.labda3 img{
    transition-property: transform;
    transition-duration: 3s;
    transition-timing-function: ease-in;
}
```

```css
.labda4 img{
    transition-property: transform;
    transition-duration: 3s;
    transition-timing-function: ease-out;
}

.labda5 img{
    transition-property: transform;
    transition-duration: 3s;
    transition-timing-function: ease-in-out;
}
```

A CSS3 átmenetek késleltetése

A transition-delay tulajdonság segítségével adható meg a CSS tulajdonság(ok) változásainak késleltetése (hány másodperccel kezdődjön később az átmenet).
Az átmenet késleltetésének beállítása:
kijelölő{
transition-delay: érték;
}

```html
<div id="palya">
    <div class="labda1"><img src="labda.png" width="100" alt=""></div>
    <div class="labda2"><img src="labda.png" width="100" alt=""></div>
    <div class="labda3"><img src="labda.png" width="100" alt=""></div>
</div>
```

```
.labda1 img{
    transition-property: transform;
    transition-duration: 3s;
    transition-delay: 1s;

}

.labda2 img{
    transition-property: transform;
    transition-duration: 3s;
    transition-delay: 2s;
}

.labda3 img{
    transition-property: transform;
    transition-duration: 3s;
    transition-delay: 3s;
}
```

A CSS3 animációk kulcskockái

A CSS3 animációk esetében kötelező a változó CSS tulajdonságok kezdeti és utolsó értékének a megadása. Az animáció ezen értékek folyamatos változása révén jön létre.
A kulcskockák beállítása: @keyframes deklarációban meg kell adni az animáció nevét, valamint a módosítandó jellemző(k)nek minimum a kezdő és végértékét. Szóközzel elválasztva több tulajdonság is megadható.
Az első kulcskocka jelölésére használható a from kulcsszó és a 0% jelölés is, az utolsó kulcskocka jelölésére pedig a to kulcsszó és a 100% jelölés is. A közbenső állapotok esetében a %-os jelölést használjuk, például az animáció időtartamának a felét 50%-al jelöljük.

CSS3 animáció létrehozása

Az animáció nevének beállítása:
kijelölő{
    animation-name: animacio_neve;
}
Az animáció neve meg kell egyezzen a @keyframes deklarációban megadott névvel. Ez teszi lehetővé a kulcskockák és az animáció összerendelését.
Az animáció időtartamának beállítása:
kijelölő{
    animation-duration: érték;
}
Az animation-duration tulajdonság segítségével meg kell adni, hogy az animáció az első kulcskockától indulva mennyi idő alatt jut el az utolsóig. Az időtartam megadása kötelező, mértékegysége lehet s vagy ms.

```css
@keyframes labda{
    0% { transform: translateX(1px) rotate(1deg); }
    50% { transform: translateX(100px) rotate(90deg); }
    100% { transform: translateX(400px) rotate(360deg); }

}

#palya .labda1 img{
    animation-name: labda;
    animation-duration: 3s;
}

#palya .labda2 img{
    animation-name: labda;
    animation-duration: 5s;
}

#palya .labda3 img{
    animation-name: labda;
    animation-duration: 7s;
}
```

Az animáció ismétlése

Az animation-iteration-count tulajdonság segítségével megadható, hogy hányszor ismétlődjön meg az animáció. Az infinite kulcsszó esetén végtelen számú ismétlést írunk elő, tehát amíg látszik az adott elem a képernyőn, addig az animáció is újra és újra indul.
Az animáció ismétlésének beállítása:
kijelölő{
        animation-iteration-count: érték;
}

```css
#palya .labda1 img{
    animation-name: labda;
    animation-duration: 3s;
    animation-iteration-count: 3;
}

#palya .labda2 img{
    animation-name: labda;
    animation-duration: 5s;
    animation-iteration-count: 5;
}

#palya .labda3 img{
    animation-name: labda;
    animation-duration: 7s;
    animation-iteration-count: infinite;
}
```

A CSS3 animáció késleltetése

Az animation-delay tulajdonság segítségével adott idővel késleltethető az animáció elindulása.
Megadható másodpercben és ezredmásodpercben. Negatív érték esetén az animáció nem az elejéről indul, hanem onnan, ahol a megadott idővel később tartana.
Az animáció késleltetésének beállítása:
kijelölő{
    animation-delay: érték;
}

```
#palya .labda1 img{
    animation-name: labda;
    animation-duration: 3s;
    animation-delay: 3s;
}

#palya .labda2 img{
    animation-name: labda;
    animation-duration: 5s;
    animation-delay: 5s;
}

#palya .labda3 img{
    animation-name: labda;
    animation-duration: 7s;
    animation-delay: -3s;
}
```

A CSS3 animáció iránya

Az animation-direction tulajdonság segítségével megadható, hogy ismételt lejátszás esetén honnan kezdődjön és milyen irányba haladjon a lejátszás.
Lehetséges értékei:

- normal: mindig az elején kezdődik és a vége felé halad
- alternate: a végét elérve visszafordul, tehát minden második alkalommal a végétől az eleje felé halad
- alternate-reverse: a végéről indul, majd az elejét elérve visszafordul, tehát minden második alkalommal az elejétől a vége felé halad

Az animáció irányának beállítása:
kijelölő{
    animation-direction: kulcsszó;
}

```css
#palya .labda1 img{
    animation-name: labda;
    animation-duration: 3s;
    animation-iteration-count: 3;
    animation-direction: normal;
}

#palya .labda2 img{
    animation-name: labda;
    animation-duration: 5s;
    animation-iteration-count: 5;
    animation-direction: alternate;
}

#palya .labda3 img{
    animation-name: labda;
    animation-duration: 7s;
    animation-iteration-count: infinite;
    animation-direction: alternate-reverse;
}
```

A CSS3 animáció időkarakterisztikája

Az animation-timing-function tulajdonság segítségével adható meg a CSS tulajdonság(ok) változásának karakterisztikája.
Lehetséges értékei:

- linear: állandó sebességű
- ease: a változás eleje és vége lassú, közepén gyors
- ease-in: a változás eleje lassú, majd gyorsul
- ease-out: gyors változás, a végén lassul
- ease-in-out: a változás eleje és vége lassú, közepén gyors

Az időkarakterisztika beállítása:
kijelölő{
    animation-timing-function: kulcsszó;
}

```html
<div id="palya">
    <div class="labda1"><img src="labda.png" width="100" alt=""></div>
    <div class="labda2"><img src="labda.png" width="100" alt=""></div>
    <div class="labda3"><img src="labda.png" width="100" alt=""></div>
    <div class="labda4"><img src="labda.png" width="100" alt=""></div>
    <div class="labda5"><img src="labda.png" width="100" alt=""></div>
</div>
```

```css
@keyframes labda{
    0% { transform: translateX(1px) rotate(1deg); }
    100% { transform: translateX(400px) rotate(360deg); }

}

#palya .labda1 img{
    animation-name: labda;
    animation-duration: 5s;
    animation-iteration-count: infinite;
    animation-timing-function: linear;
}

#palya .labda2 img{
    animation-name: labda;
    animation-duration: 5s;
    animation-iteration-count: infinite;
    animation-timing-function: ease;
}
```

```css
#palya .labda3 img{
    animation-name: labda;
    animation-duration: 5s;
    animation-iteration-count: infinite;
    animation-timing-function: ease-in;
}

#palya .labda4 img{
    animation-name: labda;
    animation-duration: 5s;
    animation-iteration-count: infinite;
    animation-timing-function: ease-out;
}

#palya .labda5 img{
    animation-name: labda;
    animation-duration: 5s;
    animation-iteration-count: infinite;
    animation-timing-function: ease-in-out;
}
```

A színátmenet (gradient)

Két szín használata esetén a kezdő és végérték között a böngésző egyenletes átmenetet biztosít. Megadhatók közbenső értékek is, és azok pozíciói is százalékban vagy hosszmértékben. Ha a közbenső értékekhez nem rendelünk pozíciót, akkor arányosan lesznek elhelyezve (pl. egy közbenső érték középre kerül).
Lineáris átmenetnél egy egyenes (gradient-line) mentén jön létre az átmenet, sugárirányú átmenetnél egy pontból kiindulva ellipszis alakban jön létre az átmenet.

A lineáris átmenet iránya

A gradiens-vonal iránya megadható a függőleges tengellyel bezárt szögével:

- 0: felfelé
- 90: jobbra
- 180: lefelé
- 270: balra

Használható kulcsszavak:

- to right: balról jobbra
- to left: jobbról balra
- to bottom: fentről lefelé
- to top: lentről felfelé
- to bottom right: bal felső sarokból a jobb alsó sarok felé
- to bottom left: jobb felső sarokból a bal alsó sarok felé
- to top right: bal alsó sarokból a jobb felső sarok felé
- to top left: jobb alsó sarokból a bal felső sarok felé

A lineáris átmenet beállítása:

kijelölő{
    background-image: linear-gradient(irány, szín1, szín2);
}

```html
<div class="doboz1">balról jobbra</div>
<div class="doboz2">90 fok</div>
<div class="doboz3">bal alsó sarokból jobb felső sarok felé</div>
<div class="doboz4">45 fok</div>
```

```css
div{
    width: 200px;
    height: 100px;
    margin: 20px;
    text-align: center;
}

.doboz1{
    background-image: linear-gradient(to right, #000000, #ffffff);
}

.doboz2{
    background-image: linear-gradient(90deg, #000000, #ffffff);
}

.doboz3{
    background-image: linear-gradient(to top right, #000000, #ffffff);
}

.doboz4{
    background-image: linear-gradient(45deg, #000000, #ffffff);
}
```

Több lineáris színátmenet

Kettőnél több színt is megadhatunk.
Ha a közbenső értékekhez nem adunk meg pozíciót, akkor arányosan lesznek elhelyezve.
Megadhatjuk az egyes színek gradiens-vonalon elfoglalt helyzetét hosszértékkel vagy százalékban.
Több lineáris színátmenet beállítása:
kijelölő{
background-image: linear-gradient(irány, szín1 pozíció, szín2 pozíció, …, színN pozíció);
}

```html
<div class="doboz1"></div>
<div class="doboz2"></div>
<div class="doboz3"></div>
```

```css
.doboz1{
    background-image: linear-gradient(to right, blue, gray, green);
}

.doboz2{
    background-image: linear-gradient(to right, blue, gray 20%, green);
}

.doboz3{
    background-image: linear-gradient(to right, blue, gray 40px, green);
}
```

Lineáris színátmenet és átlátszóság

Ha a színátmenet definiálásakor a színek megadására az rgba() függvényt használjuk, akkor a háttér egyes részei áttetszővé tehetők. Az átlátszóság különböző értékei között ekkor folyamatos átmenet jön létre.
Lineáris színátmenetben átlátszóság beállítása:
kijelölő{
background-image: linear-gradient(irány, rgba(...), rgba(...));
}

```css
div{
    width: 200px;
    height: 100px;
    margin: 20px;
    text-align: center;
    background-color: #ff0000;
}
```

```css
.doboz1{
    background-image: linear-gradient(to right, rgba(0, 0, 255, 1), rgba(0, 0
    255, 0));
}

.doboz2{
    background-image: linear-gradient(to right, rgba(0, 0, 255, 0.7), rgba(0,
    0, 255, 0.3));
}

.doboz3{
    background-image: linear-gradient(to right, rgba(0, 255, 0, 0.7), rgba(0,
    0, 255, 0.3));
}
```

Lineáris színátmenet ismétlése

A repeating-linear-gradient() függvény használatával az átmenetek periodikusan ismétlődnek.
A lineáris színátmenet ismétlésének beállítása:
kijelölő{
    background-image: repeating-linear-gradient(irány, szín1 pozíció, szín2 pozíció, …, színN pozíció);
}

```html
<div class="doboz1"></div>
<div class="doboz2"></div>
```

```css
.doboz1{
    background-image: repeating-linear-gradient(45deg, #000066, #0000ff 20px)
}

.doboz2{
    background-image: repeating-linear-gradient(90deg, #000066, #0000ff 20%);
}
```

Sugárirányú színátmenetek

Sugárirányú színátmenetnél egy pontból kiindulva kör (circle) vagy ellipszis (ellipse) alakban jön létre az átmenet.
Sugárirányú színátmenet beállítása:
kijelölő{
    background-image: radial-gradient(forma, méret at pozíció, szín1 pozíció, szín2 pozíció, …, színN pozíció);
}

```css
div{
    width: 400px;
    height: 200px;
    margin: 20px;
    text-align: center;
}
```

```css
.doboz1{
    background-image: radial-gradient(circle closest-side at center, #4444ff,
    #000044);
}

.doboz2{
    background-image: radial-gradient(ellipse closest-side at center, #4444ff,
    #000044);
}
```

A sugárirányú színátmenetek pozíciója

A sugárirányú színátmenet pozíciója megadható kulcsszóval, hosszméretben vagy százalékban.
Használható kulcsszavak:

- left top: bal felső sarok
- right top: jobb felső sarok
- left bottom: bal alsó sarok
- right bottom: jobb alsó sarok
- left center: baloldalon, függőlegesen középen
- right center: jobboldalon, függőlegesen középen
- center top: fent, vízszintesen középen
- center bottom: lent, vízszintesen középen
- center: vízszintesen és függőlegesen középen

```html
<div class="doboz1"></div>
<div class="doboz2"></div>
<div class="doboz3"></div>
```

```css
.doboz1{
    background-image: radial-gradient(circle at left top, #4444ff, #000044);
}

.doboz2{
    background-image: radial-gradient(circle closest-side at 20% 60%, #4444ff,
    #000044);
}

.doboz3{
    background-image: radial-gradient(circle closest-side at 80px 120px,
    #4444ff, #000044);
}
```

A sugárirányú színátmenetek mérete

A befoglaló kör vagy ellipszis méretét kulcsszavakkal adhatjuk meg:

- closest-side: az alakzat széle érinti a befoglaló elem legközelebbi oldalát
- farthest-side: az alakzat széle érinti a befoglaló elem legtávolabbi oldalát
- closest-corner: az alakzat széle érinti a befoglaló elem legközelebbi sarkát
- farthest-corner: az alakzat széle érinti a befoglaló elem legtávolabbi sarkát

```html
<div class="doboz1"></div>
<div class="doboz2"></div>
<div class="doboz3"></div>
<div class="doboz4"></div>
```

```css
.doboz1{
    background-image: radial-gradient(circle closest-side at 20% 60%, #4444ff,
    #000044);
}

.doboz2{
    background-image: radial-gradient(circle farthest-side at 20% 60%, #4444ff,
    #000044);
}

.doboz3{
    background-image: radial-gradient(circle closest-corner at 20% 60%,
    #4444ff, #000044);
}

.doboz4{
    background-image: radial-gradient(circle farthest-corner at 20% 60%,
    #4444ff, #000044);
}
```

Reszponzív weboldalak készítése

A reszponzív weboldalakkal kapcsolatos alapfogalmak

A viewport fogalma

Mi a viewport?

A mobileszközök böngészőprogramjai az oldalakat virtuális ablakban, úgynevezett viewportban (nézetablak) jelenítik meg.
Ha a viewport szélesebb a mobileszköz képernyőjénél, akkor a felhasználók az oldalt több irányba mozgatva, illetve az egyes területekre ráközelítve nézhetik meg.

Ennek elkerülésére be kell állítani, hogy a közzétett tartalom minden esetben a kijelzőre optimalizáltan jelenjen meg.

A kijelzők mérete

Egy adott eszköz képernyőméretét a viewportsizer.com weboldalon keresztül kérdezhetjük le.
A Devices menüben további eszközök adatait is megtekinthetjük egy táblázatban.
A táblázatban eszközönként két-két szélesség- és magasságadat szerepel, mivel a mobileszközök fizikai méretei különbözhetnek a bennük használt viewportokétól. A mobileszközök pixelsűrűsége jóval nagyobb, mint monitoroké.
A CSS-szabvány ezért ebben az esetben nem a fizikai pixellel számol, hanem úgynevezett referenciapixelt használ. A táblázat css width és css height oszlopaiban tehát a referenciapixellel átszámolt értékek láthatók. A honlapszerkesztés során ezek az átszámolt értékek fontosak a méretek beállításánál.

A viewport tulajdonságai

A viewport tulajdonságai a <meta> taggel állíthatók be. A name paraméter értéke a viewport, a content esetében pedig tulajdonság-érték párokat lehet megadni egymástól vesszővel elválasztva.

Példa a viewport tulajdonságainak megadására

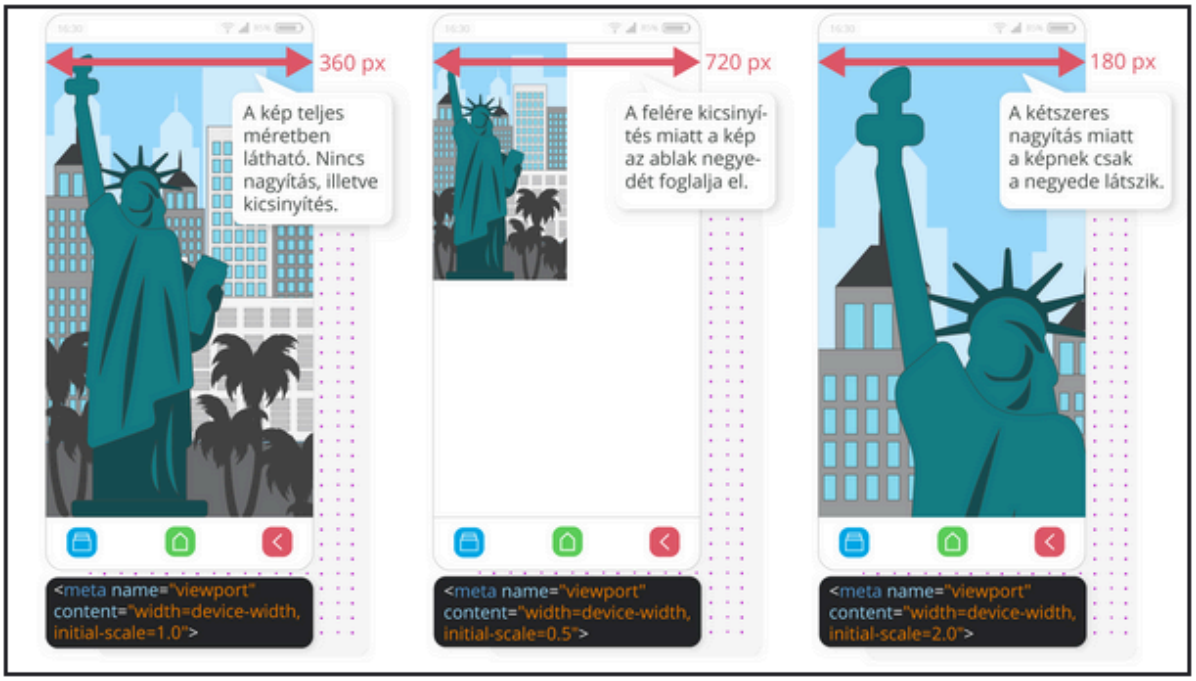
```
<meta name="viewport"
content="width=device-width, initial-scale=1.0">
```

A viewport szélessége, magassága és a nagyítási szint

A content paraméterrel a következő tulajdonságok állíthatók be:

- initial-scale: alapértelmezett nagyítási szint, az oldal első betöltésekor érvényesül (pl. 1.0)
- width: a viewport szélessége
- height: a viewport magassága

Mind a szélesség, mind a magasság esetében megadhatók konkrét számértékek és olyan speciális értékek (device-width, device-height), amelyekkel az aktuális eszköz kijelzőjének szélességét és magasságát lehet beállítani (normál nagyítási szintre vonatkozólag).



A content paraméter segítségével lehetőség van korlátok közé szorítani a nagyítás-kicsinyítés mértékét. Ehhez használhatjuk a maximum-scale és a minimum-scale kulcsszavakat.

Példa a maximális és a minimális nagyítási szint beállítására

```
<meta name="viewport"
content="width=device-width, initial-scale=1, maximum-scale=1.5, minimum-scale=0.7">
```

A nagyíthatóság engedélyezése

A content paraméternél a user-scalable kulcsszó segítségével engedélyezhetjük (yes), illetve letilthatjuk (no) az oldal nagyíthatóságát.
A legtöbb oldal természetesen lehetővé teszi a nagyítást a felhasználók számára, de akad egy-két kivétel. Ilyen például a Google térképe. Ezekre a user-scalable=no van beállítva. Ennek az az oka, hogy a térképek pixelessé válnának, ha a felhasználók az eszköz nagyítás funkciójával közelítenének rá. Így a Google térképoldalán nagyítás helyett részletesebb térképet tölt be az alkalmazás a felhasználók nagyító mozdulatára.

Példa a nagyíthatóság tiltására

```
<meta name="viewport"
content="width=device-width, initial-scale=1, user-
scalable=no">
```

A viewport beállításához használható mértékegységek

A CSS3-szabványban használhatunk olyan mértékegységeket, amelyek a nézetablak méretéhez történő méretmegadást teszik lehetővé:

- vw: a viewport szélességéhez igazodik, az 1 vw a viewport szélesség 1%-ának felel meg
- vh: a viewport magasságához igazodik, az 1 vh a viewport magasság 1%-ának felel meg
- vmin: a vw és a vh közül a kisebbnek felel meg
- vmax: a vw és a vh közül a nagyobbnak felel meg

A vmin és vmax mértékegységeknek akkor vehetjük hasznát, ha az álló/fekvő üzemmód miatt a ténylegesen kisebb/nagyobb oldal méretéhez kell viszonyítani az elemek méretét.

Az RWD

Mi az RWD?

Az RWD, azaz a Responsive (reszponzív) Web Design célja, hogy valamennyi eszközön (a mobiltelefonoktól a nagy felbontású monitorokig) optimálisan jelenítse meg a tartalmat.
Az optimális megjelenés magába foglalja a jó olvashatóságot, a könnyű kezelhetőséget és az egyszerű navigálhatóságot.

Az RWD főbb összetevői

Ahhoz, hogy egy weboldal valóban alkalmazkodjon a megjelenítő eszköz tulajdonságaihoz, az alábbi összetevők szükségesek:

- folyékony rács
- folyékonyan átméretezhető képek és médiaelemek
- médialekérdezések

A folyékony rács

A rács egy olyan struktúra, amely függőleges és vízszintes tengely mentén teszi lehetővé a tartalom elhelyezését.

A rács akkor válik “folyékonnyá”, ha pixelben rögzített szélességek helyett relatív mértékegységeket használunk. A relatív méretmegadások lehetővé teszik, hogy rugalmasan változzon a megjelenítendő tartalom.

A folyékonyan átméretezhető képek és médiaelemek

A folyékony rács mellett szükség van folyékonyan átméretezhető képekre és médiaelemekre. Ez azt jelenti, hogy az egyes elemeknek a rendelkezésre álló hely függvényében kell változtatniuk a méretüket.

A médialekérdezések

Ahhoz, hogy a különböző tulajdonságú eszközökre más-más stílusokat lehessen beállítani, úgynevezett médialekérdezések szükségesek.
A médialekérdezés tulajdonképpen logikai kifejezés, amely igaz vagy hamis értéket vehet fel. Ha a megadott médiaelem típusa megegyezik az eszköz médiatípusával és / vagy egyéb feltételek (képernyőszélesség és -magasság) is teljesülnek, akkor a lekérdezés igaz.

Médialekérdezés a <link> taggel

Médialekérdezés készíthető a <link> taggel is. A media paraméter értékeként állítható be, hogy melyik eszközre vonatkozzon az adott beállítás:

- screen: képernyő
- print: nyomtatás
- projection: kivetítő
- tv: televízió
- all: összes eszköz

A media paraméter értékeként nem csupán a médiatípus, hanem feltételek is megadhatók. A feltételekben használhatók logikai műveletek (and, not) és olyan gyakran használt tulajdonságok, mint a width, min-width, max-width, height, min-height vagy max-height.
A tájolás szintén a feltételek részeként állítható be. Ehhez az orientation kulcsszó szükséges, értékei lehetnek:

- portrait (álló)
- landscape (fekvő)

Példa a <link> taggel való médialekérdezésre

Az első példában láthatod, hogy a 800felett.css nevű külső stíluslap tartalma akkor érvényesül, ha az oldal minimum 800 px széles képernyőn jelenik meg.

A második példában a beállítások akkor érvényesülnek, ha a megjelenítő eszköz egy fekvő tájolású képernyő.

```
<!--Első példa-->

<link rel="stylesheet" type="text/css"
media="screen and (min-width: 800px)"
href="800felett.css">

<!--Második példa-->

<link rel="stylesheet" type="text/css"
media="screen and (orientation: landscape)"
href="fekvo.css">
```

Médialekérdezés a @media szabály használatával

A @media szabállyal a stíluslapon belül lehet médialekérdezéseket megvalósítani. Ebben az esetben a médiatípusok nevét és szükség esetén hozzájuk kapcsolt feltételeket kell selectorként használni, majd a hozzájuk tartozó deklarációs blokkban állíthatók be az egyes elemekre vonatkozó formázások.

Példa a @media szabállyal való médialekérdezésre

Az első példában a nyomtatáshoz tartozik egy lekérdezés. A szabály itt kimondja, hogy az oldal betűtípusa legyen serif (talpas).

A második példában a feltételek miatt a 400 és 700 px közötti szélességű képernyőkön a fontos azonosítójú div elem jobbra fog lebegni.

```
/*Első példa*/

@media print {
   body {font-family: serif;}
   }

/*Második példa*/

@media screen and (min-width: 400px) and (max-width: 700px) {
   div#fontos {float: right;}
   }
```

A mobile first tervezés

A mobile first tervezés lényege, hogy a honlapot kifejezetten a mobileszközökre optimalizáltan szükséges megtervezni, majd ezt alapul véve kell a nagyobb felbontású eszközök (tablet, notebook, asztali számítógép) felületeit kialakítani.
A mobileszközökön a felhasználó ujja tölti be a kurzor szerepét. Az ujj jóval vastagabb, mint a kurzor, így könnyen előfordulhat, hogy a felhasználó rossz elemet választ ki. A tervezés során nagy figyelmet kell fordítani arra, hogy optimális legyen a kiválasztható elemek mérete és elegendő térköz legyen köztük.
A méret mellett az elemeknek az oldalon betöltött pozíciójára is oda kell figyelni. A felhasználók többsége egykezes navigáció során a hüvelykujját használja, így igen körülményes a képernyő bal felső sarkába kattintani (főleg ha jobbkezes a felhasználó).
A mobile first tervezés fontos előnye, hogy azok az oldalak, amelyek optimálisak a mobileszközökre, jellemzően előkelőbb helyet foglalnak el a Google-keresések találati listáján.

Gyakorlati feladat (Microsoft Teams 20230801_1.pdf)

A feladatot a Codepen vagy a VS Code segítségével készítsd el.
Forrás: https://codepen.io/webalap/pen/yLaBJwR
Tedd reszponzívvá az oldalt.
A container osztályú elemnek fix szélesség van beállítva. Ezt módosítsd úgy, hogy a maximális szélessége 720 px, az aktuális szélessége pedig 100%-os legyen.
Az egyik bekezdés elején találsz egy main tárolóelembe helyezett fix szélességű képet. Módosítsd a stílusdefiníciót úgy, hogy a kép szélessége a szülőelemhez képest 50%-os legyen. A minimum szélességet állítsd 200 px-re.
Készíts a stíluslapban médialekérdezést a legfeljebb 600 px szélességű kijelzőkre. A lekérdezett elemekre a következő beállítások legyenek érvényesek:

- *A main tárolóelem bekezdésében található kép szélessége a szülőelemhez képest legyen 90%-os. Igazítsd középre a képet, és állítsd be, hogy alatta jelenjen meg a szöveg. Add meg, hogy a kép függőleges margója legyen 10 px nagyságú.*
- *Az images osztályú szakaszon belüli képre állítsd be, hogy középre legyen igazítva, és alatta jelenjen meg a hozzá tartozó szöveg. A kép függőleges irányú margója itt is legyen 10 px.*

Készíts a legfeljebb 400 px szélességű kijelzőkhöz egy médialekérdezést. Ez tartalmazza a következő beállításokat:

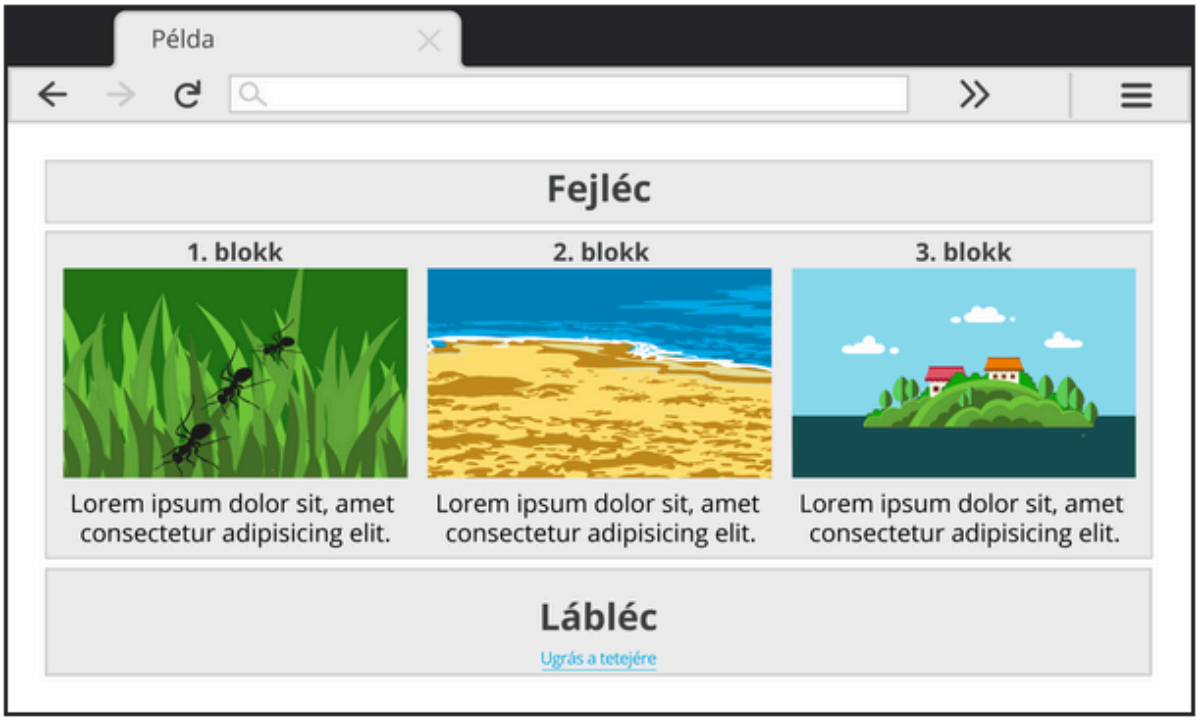
- *A képek ne jelenjenek meg az oldalon.*
- *A fejlécnek ne legyen háttérképe, hanem helyette rózsaszín (R: 216, G: 179, B: 180) háttérszín jelenjen meg. A fejléc magassága igazodjon a tartalomhoz.*
- *A fejlécben található egyes szintű címsor az úsztatás helyett legyen középre igazítva. Töröld a jobb oldali margót. A betűméret egyezzen meg az alapbeállítások felével.*

Esettanulmány

Gyakorlati példa a reszponzív weboldalak kialakításához szükséges módszerek megismeréséhez.

Az esettanulmány bevezetése

A következőkben egy fejlécre, tartalmi blokkokra és láblécre tagolt oldal különböző eszközökön való nézeteit készítjük el. Láthatjuk, hogy alapesetben a tartalmi blokk három oszlopból áll. Mobilnézetben ezt egyoszlopúvá alakítjuk. A nyomtatáshoz pedig készítünk egy olyan stíluslapot, amely eltünteti a szegélyeket és módosítja a háttérszínt és a betűtípust.



Az alapmegjelenés beállítása

Az oldal alapmegjelenéséhez a következő HTML-kód szükséges:

```
<body>
  <header>
    <h1>Fejléc</h1>
  </header>
  <main>
    <section>
      <h2>1. blokk</h2>
      <img src="https://lorempixel.com/300/200/nature/1/" alt="">
      <p>Lorem ipsum dolor sit, amet consectetur adipisicing elit.</p>
    </section>
    <section>
      <h2>2. blokk</h2>
      <img src="https://lorempixel.com/300/200/nature/2/" alt="">
      <p>Lorem ipsum dolor sit, amet consectetur adipisicing elit.</p>
    </section>
    <section>
      <h2>3. blokk</h2>
      <img src="https://lorempixel.com/300/200/nature/3/" alt="">
      <p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. </p>
    </section>
  </main>
  <footer>
    <h1>Lábléc</h1>
    <a href="#">Ugrás a tetejére</a>
  </footer>
</body>
```

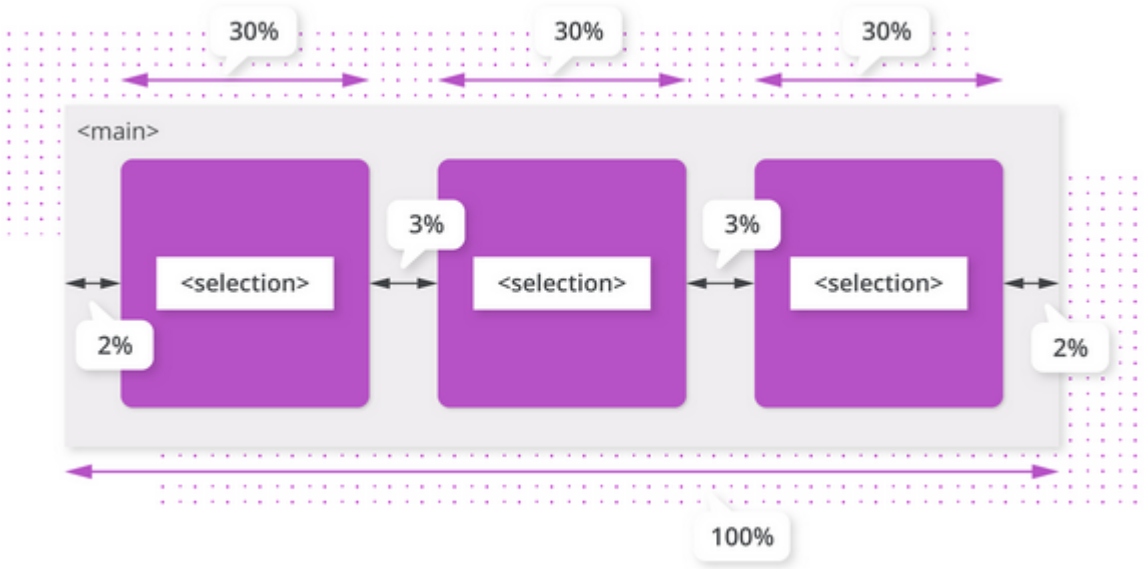
A folyékony arculatért felelős stílusbeállításokat helyezzük el egy külön CSS-állományban (alap.css).

```
<head>
  <meta charset="UTF-8">

  <meta name="viewport"
  content="width=device-width, initial-scale=1">

  <link stylesheet="stylesheet" href="alap.css">
</head>
```

Az alap.css állományban először az oldal betűtípusát és az egyes szerkezeti elemek világosszürke háttérszínét, valamint a szülőelemhez viszonyított 95%-os szélességét állítjuk be. A szegély szélességét a viewporthoz képest adjuk meg. A szövegeket és a blokkokat középre igazítjuk.

```css
body {font-family: sans-serif;}

main,
header,
footer {
  background-color: rgb(241,241,241);
  width: 95%;
  border: 0.5vw solid darkgray;
  text-align: center;
  margin: 10px auto;
  }
```

A három oszlopból álló elrendezést folyékonyan kell kialakítani, ezért a méreteit valamelyik relatív mértékegységgel (pl %) szükséges megadni.
A section elemek szélessége legyen 30%, az első elem előtti és az utolsó elem utáni margó legyen 2-2%, a közbenső részeken a margó legyen 3%. Ez megoldható úgy, hogy az elemek bal oldalán 2 px, a jobb oldalán pedig 1 px széles margót használunk. Az elemek között összeadódik a 2 px és az 1 px széles margó. Az utolsó section elemnél pedig a last-child pseudo class segítségével beállítjuk, hogy az utolsó gyermekelem jobb oldali margója 2% legyen.
Az elemek egymás mellé helyezéséhez lebegtessük őket balra a float tulajdonság segítségével.



```
section {
  float: left;
  margin-bottom: 10px;
  margin-top: 10px;
  padding: 0;
  width: 30%;
  margin-left: 2%;
  margin-right: 1%;
  }

section:last-child {
  margin-right: 2%;
  }
```

A fejléc és a tartalom között megjelent egy felesleges vonal, a lábléc körül pedig eltűnt a korábban beállított szegély.
Mivel a section elemek lebegtetve vannak, az alapértelmezett megjelenés miatt kilógnak a main elemből. Ennek megoldására használjuk a main elem esetében az overflow: hidden tulajdonságot.

Végül szükséges beállítani, hogy a médiaelemek a rendelkezésre álló hely függvényében változtassák a méretüket.
Ehhez a section elemekben elhelyezett képek szélességét kell 100%-ra állítani. Így ha a böngészőablak átméretezése miatt megváltozik az oszlopok szélessége, akkor ezt a képek mérete aránytartóan követi. Ez a megjelenés azonban még nem megfelelő a mobileszközök esetén.

A mobilnézet kialakítása

A mobileszközökön való megjelenés módosításához először a viewport tulajdonságait kell beállítani. A viewport szélessége az eszköz képernyő szélességével legyen megegyező, az alapértelmezett nagyítási szint pedig legyen 1.

```
<head>
  <meta charset="UTF-8">

  <meta name="viewport"
  content="width=device-width, initial scale=1.0">
</head>
```

A kisebb felbontású eszközöknél használjunk újabb stíluslapot (mobil.css), ezt csatoljuk a <link> taggel a HTML állományhoz. A media paraméter segítségével

állítsuk be, hogy a stíluslapban lévő szabályok olyan képernyőkön érvényesüljenek, amelyek maximális szélessége 500 px.

```
<head>
   <meta charset="UTF-8">
   <meta name="viewport"
   content="width=device-width,
   initial-scale=1.0">

   <link stylesheet="stylesheet" href="alap.css">

<!--500px-nél kisebb képernyőszélességre vonatkozó
stíluslap-->

   <link stylesheet="stylesheet" href="mobil.css"
   media="screen and (max-width: 500px)"
</head>
```

A létrehozott mobil.css állományban először a section elem szélességét állítsuk be 100%-ra és kapcsoljuk ki a float funkciót, hogy a tartalom egy oszlopba rendezve jelenjen meg.
Az alsó és felső margó legyen 20 px, a bal és jobb oldali margó pedig legyen 0 px. Végül az elemnek készítsünk egy alsó szegélyt, hogy a blokkok vizuálisan is elkülönüljenek egymástól.

```
section {
   width: 100%;
   float: none;
   margin: 20px 0;
   border-bottom: 2px solid gray;
   }

section:last-child {border-bottom: none;}
```

A nyomtatási beállítások

A háttérszínek (és akár a szegélyek) eltávolításával tintát lehet spórolni. A serif betűtípus beállítása javítaná a szöveges tartalmak olvashatóságát. A felesleges elemeket (pl. Ugrás a tetejére link) érdemes eltávolítani.

A nyomtatáshoz érdemes egy külön stíluslapot (nyomtatas.css) létrehozni. A médiatípus beállításánál ezúttal használjuk a print kulcsszót.

```
<head>
<link stylesheet="stylesheet" href="alap.css">

<!--500px-nél kisebb képernyőszélességre vonatkozó
stíluslap-->

<link stylesheet="stylesheet" href="mobil.css"
media="screen and (max-width: 500px)">

<!--Nyomtatási stíluslap-->

<link stylesheet="stylesheet" href="nyomtatas.css"
media="print">
</head>
```

A stíluslapban a következő módosításokat állítsuk be:

- fehér háttérszín (fontos, hogy ezt a header, a main, a section és a footer elemekre külön-külön be kell állítani)
- serif betűtípus
- centiméterben megadott margók
- a header, a main és a footer elemek 100% nagyságú szélessége
- a szegélyek törlése
- a navigációs szerepet betöltő link eltávolítása

```css
body {
  background-color: white;
  font-family: serif;
  margin: 1.5cm;
  }

header,
main,
footer {
  width: 100%;
  background-color: white;
  border: none;
  }

section {
  background-color: white;
  }

/*A navigációs elem szerepét betöltő link eltávolítása*/

footer a {
  display: none;
  }
```

A stílusállomány használata

A HTML-állományba jelenleg három különböző stíluslap van belinkelve. Ezeket összevonhatjuk egyetlen css állományba is. A @media szabályt használva a mobil.css és a nyomtatas.css beállításait az alapbeállításokat tartalmazó css állományba helyezhetjük el.

Példa a stíluslapok egy állományban való összevonására

Az új stilus.css fájl belinkelése a HTML-állományba

```html
<head>
  <meta charset="UTF-8">

  <meta name="viewport"
  content="width=device-width, initial-scale=1.0">

  <link stylesheet="stylesheet"
  href="stilus.css">
</head>
```

Példa a stíluslapok egy állományban való összevonására

A mobil- és nyomtatási nézetre vonatkozó beállítások a stilus.css fájlban

```
/*Mobilnézetre érvényes szabályok*/
@media screen and (max-width: 500px) {
  section {
    width: 100%;
    float: none;
    margin: 20px 0;
    border-bottom: 2px solid gray;
  }
/*További szabályok*/
}

/*Nyomtatásra érvényes szabályok*/
@media print {
  body {
    background-color: white;
    font-family: serif;
    margin: 1.5cm;
  }
/*További szabályok*/
}
```

Ismerkedés a Bootstrap keretrendszerrel

A Bootstrap keretrendszer

Az első lépések

Mi a Bootstrap?

A Bootstrap az egyik legnépszerűbb keretrendszer a reszponzív weboldalak fejlesztéséhez. Nagyon részletes, jól használható és példakódokkal van kiegészítve a dokumentációja (https://getbootstrap.com/). A fejlesztéshez egyszerűen használható rácsrendszert használ.

A Bootstrap kompatibilitása

A Bootstrap valamennyi gyakran használt böngészőprogrammal (mobil is) kompatibilis.

A Bootstrap használata a fájlcsomag letöltésével

A Bootstrap kétféle módon használható. Ebből az egyik, hogy a működéséhez szükséges CSS- és JavaScript állományokat letöltjük a saját gépünkre. A fájlcsomagot a https://getbootstrap.com oldalon találjuk. A Download gombra kattintva több fájlcsomag is elérhető.

A Bootstrap használata a fájlcsomag letöltése nélkül

A másik módszer, hogy a fájlcsomag letöltése helyett a HTML-kódban a webes, tartalomkézbesítési hálózaton (CDN) elérhető állományra hivatkozunk.
Legújabb verzió: https://getbootstrap.com/docs/5.3/getting-started/introduction/
A saját stíluslapunkat a kódban mindig a Bootstrap-stíluslapra hivatkozás után helyezzük el.

A Bootstrap oldalelrendezése

A Bootstrappel többféleképpen is elő lehet állítani reszponzív weboldalakat. Közülük az alábbi két módszer tekinthető a legelterjedtebbnek:

- fix szélességű reszponzív oldalak (az oldalak különböző felbontásintervallumokhoz alkalmazkodnak, a container osztálynevet kell használni)
- folyékony reszponzív oldalak (az oldalak elrendezése megváltozik a böngészőablak átméretezésekor, a container-fluid osztálynevet kell használni)

Példa a fix szélességű és a folyékony reszponzív oldalakra

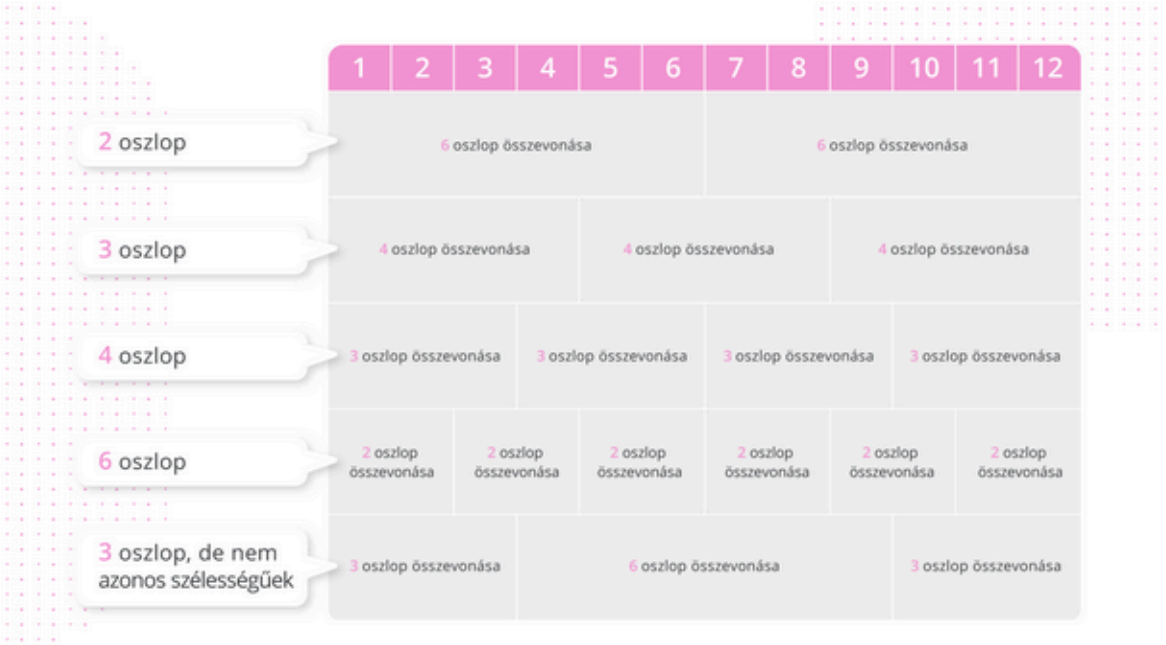
```
<head>
<style>
  div {border: solid black 1px}
  .container {background-color: lightyellow;}
  .cotainer-fluid {background-color: lightblue;}
</style>
</head>
```

```
<body>
<div class="container">
<h1>Reszponzív, fix szélességű</h1>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam tempor nunc magna, in cursus mauris facilisis quis. Quisque est eros, interdum quis magna ac, imperdiet ullamcorper dolor. Curabitur feugiat diam odio, in dignissim eros pretium vitae. Sed pulvinar nisl vitae tellus aliquet, sed tristique mauris dignissim. Phasellus nisl libero, ornare vitae ultrices porta, suscipit quis nisl. Nulla aliquam elementum ex.</p>
</div>

<div class="container-fluid">
<h1>Reszponzív, folyékony</h1>
<p>Lorem ipsum dolor sit amet, consectetur  adipiscing elit. Nam tempor nunc magna, in cursus mauris facilisis quis. Quisque est eros, interdum quis magna ac, imperdiet ullamcorper dolor. Curabitur feugiat diam odio, in dignissim eros pretium vitae. Sed pulvinar nisl vitae tellus aliquet, sed tristique mauris dignissim. Phasellus nisl libero, ornare vitae ultrices porta, suscipit quis nisl. Nulla aliquam elementum ex.</p>
</div>
</body>
```

A rács használatának alapjai

A Bootstrap rácsrendszere

A Bootstrap keretrendszerben a rács tizenkét oszlopból és tetszőleges számú sorból állhat. Ha nincs szükség mind a tizenkét oszlopra, össze lehet vonni őket.



A sorok és az oszlopok létrehozása

A rács sorait úgy alakíthatjuk ki, hogy a container / container-fluid osztályba tartozó tárolóelemen (pl. div) belül row osztályba sorolt tárolóelemeket helyezünk el. Ezután a sorokon belül col osztályba sorolt tárolóelemekkel hozhatjuk létre az oszlopokat. Ha egy sorba több elemet illesztünk a maximális tizenkettőnél, a többletelemek új sorba kerülnek.

Példa a sorok és az oszlopok létrehozására

```
<head>
<style>
  div {
  border: 1px solid black;
  }
</style>
</head>
```

```
<body>
  <div class="container">
    <div class="row">
      <div class="col">1. oszlop</div>
      <div class="col">2. oszlop</div>
      <div class="col">3. oszlop</div>
      <div class="col">4. oszlop</div>
      <div class="col">5. oszlop</div>
      <div class="col">6. oszlop</div>
      <div class="col">7. oszlop</div>
      <div class="col">8. oszlop</div>
      <div class="col">9. oszlop</div>
      <div class="col">10. oszlop</div>
      <div class="col">11. oszlop</div>
      <div class="col">12. oszlop</div>
      <div class="col">13. oszlop</div>
      <div class="col">14. oszlop</div>
      <div class="col">15. oszlop</div>
      <div class="col">16. oszlop</div>
    </div>
  </div>
</body>
```

A nem azonos méretű oszlopok használata

Ha nem azonos méretű oszlopokat szeretnénk létrehozni, akkor a col-szám osztályokat kell használni. A megadott számérték határozza meg, hogy hány rácsoszlopnyi legyen az adott oszlop szélessége. Például három oszlop összevonásához a col-3 osztályt kell használni.

Példa a nem azonos méretű oszlopok létrehozására

```
<head>
<style>
  div {border: solid black 1px}
  .container {background-color: lightyellow;}
</style>
</head>

<body>
<div class="container">
<div class="row">
  <div class="col-3">1. oszlop</div>
  <div class="col-6">2. oszlop</div>
  <div class="col-3">3. oszlop</div>
</div>
</div>
</body>
```

A rács eszközfüggő beállítása

Az eszközök csoportosítása

Az egy sorban megjelenített oszlopok száma nagyban függhet attól, hogy mekkora kijelzőn jelenik meg az adott honlap. A felhasználók eszközeinek képernyő-tulajdonságai különböznek. Ráadásul jó néhány eszközön a képernyő álló és fekvő helyzetben is használható, és ezzel együtt a viewport méretei is megváltoznak.
Ahhoz, hogy a megjelenés valamennyi eszközön megfelelő legyen, a Bootstrap a következő eszközcsoportokat különbözteti meg:

- a viewport szélessége nem haladja meg az 576 px-t, a korábbi Bootstrap verziók xs (extra small) azonosítóval jelölték, újabban nincs önálló azonosítója
- sm (small): a viewport szélessége 576-767 px, pl. okostelefonok
- md (medium): a viewport szélessége 768-991 px, pl. tabletek
- lg (large): a viewport szélessége 992-1199 px, pl. notebookok, asztali monitorok
- xl (extra large): a viewport szélessége 1200-1399 px
- xxl (extra extra large): a viewport szélessége 1400- px

A töréspontok használata a különböző eszközökön

A Bootstrap segítségével más-más felosztású rács készíthető a különböző eszközcsoportokra. Az oszlopok megadásánál a col szó után egy kötőjellel, majd az

eszközcsoport betűkódjával adható meg, hogy milyen típusú eszközre vonatkozzon a beállítás. Az összevonandó rácsoszlopok számát ebben az esetben az eszközcsoport megadását követően szükséges feltüntetni.
Ha nincs megadva eszköztípus, akkor a beállítás automatikusan az extra kicsi kijelzőjű eszközökre lesz érvényes.

Példa az eszköztípus beállítására

```
<head>
<style>
  div {border: solid black 1px}
</style>
</head>

<body>
<div class="container">
  <div class="row">
    <div class="col-sm-12">Lorem ipsum dolor sit amet
consectetur me.</div>
  </div>

  <div class="row">
    <div class="col-sm-4">Lorem ipsum</div>
    <div class="col-sm-4">Lorem ipsum</div>
    <div class="col-sm-4">Lorem ipsum</div>
  </div>
</div>
```

A Bootstrap-rács és a mobile first tervezési elv

A Bootstrap követi a mobile first tervezési elvet. Ez a gyakorlatban azt jelenti, hogy az extra kicsi eszközökre érvényes beállítás (pl. col-12) egészen addig öröklődik felfelé a nagyobb eszközökre, amíg nincs egy azt felülíró, nagyobb eszközökre vonatkozó beállítás (pl. col-lg-6).
Fontos, hogy a honlap elrendezését mindig a legkisebb felbontásra kell megtervezni, majd a nagyobb méretek felé haladva kell lépésenként végrehajtani a szükséges módosításokat.
Külön-külön elrendezést állíthatunk be akár mindegyik eszközcsoportra, de ez nem kötelező.

Példa az eszköztípusonként eltérő elrendezés készítésére

A példában látható szöveges tartalom extra kicsi kijelzőn egy, kis kijelzőn kettő, közepes kijelzőn három, nagy kijelzőn négy, míg extra nagy kijelzőn hat oszlopban jelenik meg.

```
<div class="container">
<div class="row">
  <div class="col-12 col-sm-6 col-md-4 col-lg-3 col-xl-2">Lorem ipsum.</div>
  <div class="col-12 col-sm-6 col-md-4 col-lg-3 col-xl-2">Lorem ipsum.</div>
  <div class="col-12 col-sm-6 col-md-4 col-lg-3 col-xl-2">Lorem ipsum.</div>
  <div class="col-12 col-sm-6 col-md-4 col-lg-3 col-xl-2">Lorem ipsum.</div>
  <div class="col-12 col-sm-6 col-md-4 col-lg-3 col-xl-2">Lorem ipsum.</div>
  <div class="col-12 col-sm-6 col-md-4 col-lg-3 col-xl-2">Lorem ipsum.</div>
  <div class="col-12 col-sm-6 col-md-4 col-lg-3 col-xl-2">Lorem ipsum.</div>
  <div class="col-12 col-sm-6 col-md-4 col-lg-3 col-xl-2">Lorem ipsum.</div>
  <div class="col-12 col-sm-6 col-md-4 col-lg-3 col-xl-2">Lorem ipsum.</div>
  <div class="col-12 col-sm-6 col-md-4 col-lg-3 col-xl-2">Lorem ipsum.</div>
  <div class="col-12 col-sm-6 col-md-4 col-lg-3 col-xl-2">Lorem ipsum.</div>
  <div class="col-12 col-sm-6 col-md-4 col-lg-3 col-xl-2">Lorem ipsum.</div>
</div>
</div>
```

Gyakorlati feladat (Microsoft Teams 20230808_1.pdf)

A feladatot a VS Code segítségével készítsd el.
Forrás: index_1.html
Nyisd meg a feladathoz csatolt index_1.html fájlt. Tedd az oldalt reszponzívvá a Bootstrap-rácsrendszer segítségével.
Az oldal head részében hivatkozz a Bootstrap 5.3 stíluslapjára.
Hozz létre egy div tárolóelemet és tedd a container Bootstrap-osztályba.

A tárolóelemen belül hozz létre egy újabb tárolóelemet és ebből készítsd el a rácsrendszer első sorát.
A soron belül helyezz el három, egy-egy címből és bekezdésből álló tárolóelemet. Ezek alkossák a rács oszlopait.
Módosítsd a sorok és oszlopok megjelenését eszköztípustól függően:

- *A kis méretű eszközökön használj két oszlopból álló elrendezést. Ehhez sorold az elemeket a col-sm-6 osztályba. Teszteld, hogy 576 px és 767 px között tényleg két oszlopban jelenik-e meg a szöveges tartalom.*
- *A közepes, illetve annál nagyobb viewportszélességű eszközökön a három oszlopból álló elrendezést részesítsd előnyben. Használd a col-md-4 osztályt. Teszteld az oldalt, hogy a nagyobb eszközökön valóban három oszlopban jelenik-e meg a szöveges tartalom.*

Vizsgáld meg, hogy az 576 px-nél kisebb kijelzőjű eszközökön hány oszlopos elrendezésben látható a szöveges tartalom.
A container osztályú elemet cseréld le container-fluid osztályra, és vizsgáld meg, hogyan változik az oldal megjelenése.

Gyakorlati feladat (Microsoft Teams 20230808_2.pdf)

A feladatot a VS Code segítségével készítsd el.
Forrás: index_2.html, style_2.css
Nyisd meg a feladathoz csatolt index_2.html fájlt. Tedd az oldalt reszponzívvá a Bootstrap-rácsrendszer segítségével.
Az oldal head részében hivatkozz a Bootstrap 5.3 stíluslapjára. Ügyelj rá, hogy a Bootstrap-stíluslapot a saját stílusod hivatkozása fölé illeszd be.
Mivel a kód már tartalmazza az oldal vázát, ezért az a feladatod, hogy a tárolóelemeket a megfelelő Bootstrap-osztályokba sorold:

- *Az oldal fej- és lábléce legyen 100% széles Bootstrap container elem. A container elemben ki kell alakítanod a sorokat és oszlopokat. A fej- és lábléc minden eszközön egyoszlopos elrendezésben jelenjen meg.*
- *A main tárolóelem legyen fix szélességű Bootstrap container elem. A section elemből készíts el a container elem első sorát, míg a cikkek alkossák a rácsrendszer oszlopait. A cikkek a kis méretű eszközöktől felfelé két oszlopban jelenjenek meg, míg a nagy méretű eszközöktől felfelé a háromoszlopos elrendezés érvényesüljön. 576 px alatt a cikkek egymás alatt jelenjenek meg.*

A folyékony médiaelemek

A képek folyékonyan átméretezett beillesztése

A weboldal reszponzivitásához elengedhetetlen, hogy a képek és médiaelemek folyékonyan méreteződjenek át a különböző nézettartományokban.

A Bootstrap keretrendszerben a képek esetén ezt az img-fluid osztállyal lehet megvalósítani.

Példa a képek folyékonyan átméretezett beillesztésére

```
<div class="container">
<div class="row">
  <div class="col-12 col-lg-12">
    <img class="img-fluid" src="https://lorempixel.com
/1000/200/nature" alt="">
  </div>
  <div class="col-12 col-lg-6">
    <img class="img-fluid" src="https://lorempixel.com
/1000/200/nature" alt="">
  </div>
  <div class="col-12 col-lg-6">
    <img class="img-fluid" src="https://lorempixel.com
/1000/200/nature" alt="">
  </div>
</div>
</div>
```

A beágyazott elemek folyékonyan átméretezett beillesztése

A beágyazott médiaelemek (pl. Youtube-videók) folyékonnyá tételéhez a beágyazókódot olyan elemben kell elhelyezni, amely a ratio osztályba tartozik. A ratio-XxY osztállyal adjuk meg az adott elem oldalarányát (az X helyére a vízszintes, az Y helyére a függőleges oldalméretet kell megadni).
A Bootstrap alapértelmezetten a 21:9, a 16:9, a 4:3 és az 1:1 oldalarányokat támogatja.

```
<div class="container">
    <div class="row">
        <div class="col-12">
            <div class="ratio ratio-16x9">
                <iframe width="560" height="315" src="https://www.youtube.
                com/embed/jbA2dJV679M" title="YouTube video player"
                frameborder="0" allow="accelerometer; autoplay;
                clipboard-write; encrypted-media; gyroscope;
                picture-in-picture; web-share" allowfullscreen></iframe>
            </div>
        </div>
    </div>
</div>
```

Gyakorlati feladat (Microsoft Teams 20230808_3.pdf)

A feladatot a VS Code segítségével készítsd el.
Forrás: index_3.html, style_3.css, jegmadar.jpg
Nyisd meg a feladathoz csatolt index_3.html fájlt. Már ki van alakítva az oldal reszponzív rácsszerkezete, neked csak kisebb módosításokat kell végrehajtanod.
Az utolsó oszlop után készíts egy új oszlopot. Állítsd be, hogy ez az új oszlop a közepes méretű eszközöktől felfelé kétoszlopos elrendezésben jelenjen meg, párban az előtte álló szöveges oszloppal.
Az utolsó oszlopba illeszd be a jégmadarat ábrázoló képet. A kép alternatív és buborékszövege legyen “Jégmadár”.
Láthatod, hogy a beillesztett kép még nem folyékonyan jelenik meg, mivel kilóg a tárolóelemből. Használd az img-fluid Bootstrap-osztályt, és tedd reszponzívvá a képet.

A Bootstrap beépített osztályai

A szövegek formázása a Bootstrap beépített osztályaival

A szövegformázások

A Bootstrap keretrendszerben az alábbi beépített stílusokkal lehet formázni a szöveget:

- text-uppercase: nagybetűs megjelenés
- text-lowercase: kisbetűs megjelenés
- text-capitalize: nagy kezdőbetűs megjelenés
- fw-bold: félkövér megjelenés
- fw-bolder: félkövér megjelenés (a szülőelemhez képest)
- fw-semibold: félkövér megjelenés
- fw-medium: közepes megjelenés
- fw-normal: normál megjelenés
- fw-light: vékony megjelenés
- fw-lighter: vékony megjelenés (a szülőelemhez képest)
- fst-italic: dőlt stílusú megjelenés
- fst-normal: normál stílusú megjelenés
- text-decoration-underline: aláhúzott szöveg
- text-decoration-line-through: áthúzott szöveg
- text-decoration-none: aláhúzás eltávolítása
- fs-1, fs-2, fs-3, fs-4, fs-5, fs-6: betűméretek
- text-start: balra igazítás
- text-center: középre igazítás
- text-end: jobbra igazítás
- text-eszköztípus-igazítás: igazítás az adott mérettől felfelé

```
<p class="text-uppercase">Lorem ipsum dolor.</p>
<p class="text-lowercase">Lorem ipsum dolor.</p>
<p class="text-capitalize">Lorem ipsum dolor.</p>
<p class="fw-bold">Lorem ipsum dolor.</p>
<p class="fw-bolder">Lorem ipsum dolor.</p>
<p class="fw-semibold">Lorem ipsum dolor.</p>
<p class="fw-medium">Lorem ipsum dolor.</p>
<p class="fw-normal">Lorem ipsum dolor.</p>
<p class="fw-light">Lorem ipsum dolor.</p>
<p class="fw-lighter">Lorem ipsum dolor.</p>
<p class="fst-italic">Lorem ipsum dolor.</p>
<p class="fst-normal">Lorem ipsum dolor.</p>
<p class="text-decoration-underline">Lorem ipsum dolor.</p>
<p class="text-decoration-line-through">Lorem ipsum dolor.</p>
<a href="#" class="text-decoration-none">Lorem ipsum dolor.</a>
<p class="fs-1">Lorem ipsum dolor.</p>
<p class="text-start">Lorem ipsum dolor.</p>
<p class="text-center">Lorem ipsum dolor.</p>
<p class="text-end">Lorem ipsum dolor.</p>
<p class="text-lg-end">Lorem ipsum dolor.</p>
```

A színek és a háttérszínek

A Bootstrap tartalmaz olyan osztályokat, amelyekkel színt lehet beállítani. Ezeknek az osztályoknak a nevei az elem megjelenésére (pl. sötét, világos), illetve funkciójára (siker, veszély, stb.) utalnak:

- primary - kék
- secondary - szürke
- success - zöld
- danger - piros
- warning - sárga
- info - türkiz
- light - világos
- dark - sötét

Ha a szöveg színét szeretnénk megadni, akkor az elem osztályaként a text-színnév kódot használni, ha a háttér színét, akkor pedig a bg-színnév kódot. A felsorolt nevek mellett használható a white, amivel a fehér színt lehet beállítani.

Példa a szövegek színének beállítására

```
<p class="text-primary">text-primary</p>
<p class="text-secondary">text-secondary</p>
<p class="text-success">text-success</p>
<p class="text-danger">text-danger</p>
<p class="text-warning">text-warning</p>
<p class="text-info">text-info</p>
<p class="text-light bg-dark">text-light</p>
<p class="text-dark">text-dark</p>
<p class="text-white bg-dark">text-white</p>
```

Példa a háttérszínek beállítására

```
<p class="bg-primary">bg-primary</p>
<p class="bg-secondary">bg-secondary</p>
<p class="bg-success">bg-success</p>
<p class="bg-danger">bg-danger</p>
<p class="bg-warning">bg-warning</p>
<p class="bg-info">bg-info</p>
<p class="bg-light">bg-light</p>
<p class="bg-dark">bg-dark</p>
<p class="bg-white bg-white">bg-white</p>
```

Az átlátszóság

A Bootstrap használ egy --bs-text-opacity változót, aminek az értékét módosíthatjuk.

```
<div class="text-primary">This is default primary text</div>
<div class="text-primary" style="--bs-text-opacity: .5;">This is 50% opacity
primary text</div>
```

A másik lehetőség, hogy választunk egy text-opacity-érték osztályt az alábbiak közül:

```
<div class="text-primary">This is default primary text</div>
<div class="text-primary text-opacity-75">This is 75% opacity primary text</
div>
<div class="text-primary text-opacity-50">This is 50% opacity primary text</
div>
<div class="text-primary text-opacity-25">This is 25% opacity primary text</
div>
```

A szegélyek beállítása

A szegélyek beállítására a következő beépített osztályok alkalmasak additív módon:

- border - mind a négy oldalra érvényes szegély
- border-top - felső szegély
- border-end - jobb oldali szegély
- border-bottom - alsó szegély
- border-start - bal oldali szegély

A szegélyek beállítására a következő beépített osztályok alkalmasak szubsztraktív módon (szegélyek eltávolítása):

- border-0 - mind a négy oldalon eltűnik a szegély
- border-top-0 - a felső szegély tűnik el
- border-end-0 - a jobb oldali szegély tűnik el
- border-bottom-0 - az alsó szegély tűnik el
- border-start-0 - a bal oldali szegély tűnik el

Alapértelmezetten 1 px vastag szegély veszi körbe az elemet.

```
<div class="border"></div>
<div class="border-top"></div>
<div class="border-end"></div>
<div class="border-bottom"></div>
<div class="border-start"></div>
```

```
div{
    margin: 10px;
    width: 100px;
    height: 100px;
    background-color: azure;
}
```

```
<div class="border border-0"></div>
<div class="border border-top-0"></div>
<div class="border border-end-0"></div>
<div class="border border-bottom-0"></div>
<div class="border border-start-0"></div>
```

A szegélyek színének beállításához használhatók a szövegszíneknél is használható kulcsszavak.

```
<div class="border border-primary"></div>
<div class="border border-secondary"></div>
<div class="border border-success"></div>
<div class="border border-danger"></div>
<div class="border border-warning"></div>
<div class="border border-info"></div>
<div class="border border-light"></div>
<div class="border border-dark"></div>
<div class="border border-black"></div>
<div class="border border-white"></div>
```

A szegély átlátszóságát is kétféleképpen állíthatjuk be.
Az egyik lehetőség, hogy a --bs-border-opacity változó értékét módosítjuk.

```
<div class="border border-success">This is default success border</div>
<div class="border border-success" style="--bs-border-opacity: .5;">This is
50% opacity success border</div>
```

A másik lehetőség, hogy választunk egy border-opacity-érték osztályt az alábbiak közül:

```
<div class="border border-success">This is default success border</div>
<div class="border border-success border-opacity-75">This is 75% opacity
success border</div>
<div class="border border-success border-opacity-50">This is 50% opacity
success border</div>
<div class="border border-success border-opacity-25">This is 25% opacity
success border</div>
<div class="border border-success border-opacity-10">This is 10% opacity
success border</div>
```

A szegély szélessége (vastagsága) a border-érték osztályok segítségével állítható be.

```
<div class="border border-1"></div>
<div class="border border-2"></div>
<div class="border border-3"></div>
<div class="border border-4"></div>
<div class="border border-5"></div>
```

A sarkok lekerekítéséhez az alábbi osztályokat használhatjuk:

- rounded - mind a négy sarokra érvényes lekerekítés
- rounded-top - csak a felső sarkok lekerekítése
- rounded-end - csak a jobb oldali sarkok lekerekítése
- rounded-bottom - csak az alsó sarkok lekerekítése
- rounded-start - csak a bal oldali sarkok lekerekítése
- rounded-circle - négyzet esetén kör, téglalap esetén ellipszis alakú lekerekítés

- rounded-pill - nagyobb mértékű lekerekítés, négyzet esetén kör lesz

```
<div class="rounded"></div>
<div class="rounded-top"></div>
<div class="rounded-end"></div>
<div class="rounded-bottom"></div>
<div class="rounded-start"></div>
<div class="rounded-circle"></div>
<div class="rounded-pill"></div>
```

```
div{
    margin: 10px;
    width: 100px;
    height: 100px;
    background-color: lightblue;
}
```

```
.rounded-circle, .rounded-pill{
    width: 200px;
}
```

A lekerekítés mértéke is beállítható a rounded-érték osztályok segítségével.

```
<div class="rounded-0"></div>
<div class="rounded-1"></div>
<div class="rounded-2"></div>
<div class="rounded-3"></div>
<div class="rounded-4"></div>
<div class="rounded-5"></div>
```

Az előző két beállítás össze is vonható, pl. rounded-top-5.

A thumbnailek szegélyezése

Thumbnailnek (előkép) azokat a kis méretű képeket nevezzük, amelyekre kattintva a kép nagyobb változata tölthető be.
A Bootstrap keretrendszerben a thumbnailek szegélyezésére az img-thumbnail osztályt lehet használni.

```
<a href="mezo_nagy.jpg"><img src="mezo_kicsi.jpg" alt="mező"
class="img-thumbnail"></a>
```

A méretezés (a szélesség és a magasság beállítása)

A Bootstrap-osztályokkal a szülőelemek méretéhez képest relatív méretmegadást lehet beállítani.

A szélesség megadásához a w, a magasságéhoz a h karaktert kell használni. Ezután kötőjellel kell kapcsolni a kívánt értéket (25, 50, 75 és 100%). Az értékeket százalékjel nélkül kell feltüntetni.

Példa a szélesség és a magasság megadására

```
<div class="border border-dark w-100">Lorem ipsum dolor sit amet consectetur adipiscing elit. Cumque, ut! Culpa impedit in ratione earum vero expedita possimus nisi tenetur magnam dicta!</div>

<p></p>

<div class="border border-dark w-75">Lorem ipsum dolor sit amet consectetur adipiscing elit. Cumque, ut! Culpa impedit in ratione earum vero expedita possimus nisi tenetur magnam dicta!</div>

<p></p>

<div class="border border-dark w-25">Lorem ipsum dolor sit amet consectetur adipiscing elit. Cumque, ut! Culpa impedit in ratione earum vero expedita possimus nisi tenetur magnam dicta!</div>
```

A margó és a belső margó beállítása

A margó és a belső margó beállításához az m és a p karaktereket kell használni. Ezután a következő karaktereket használhatjuk:

- t - felső
- e - jobb
- b - alsó
- s - bal
- x - vízszintes (bal és jobb)
- y - függőleges (felső és alsó)

Ezekkel a betűkkel határozhatjuk meg, hogy melyik oldalra vonatkozik a beállítás. Ha nem adjuk meg, akkor a mind a négy oldalra teljesül.
A következő karakter a kötőjel, majd a margó vagy belső margó mérete következik. Ez 0 és 5 közötti érték lehet, automatikus beállításhoz használható az auto kulcsszó. Az értékként megadott számok a CSS-ben használt alábbi mértékeknek felelnek meg:

- 1 = 0,25 rem
- 2 = 0,5 rem
- 3 = 1 rem

- 4 = 1,5 rem
- 5 = 3 rem

A blokkszintű elemeket vízszintes automatikus margóbeállítással középre lehet igazítani. A Bootstrap esetében ezt az mx-auto osztállyal meg lehet valósítani. Inline elemek esetén szükséges még a d-block osztály használata is a blokkszintűvé alakításhoz.

```
<div class="border w-50 mx-auto p-3">Lorem ipsum dolor sit amet consectetur
adipisicing elit. A harum minus, debitis dolores deleniti facere
exercitationem iusto cum, explicabo quis, quas numquam enim fuga molestias?
Libero blanditiis alias nobis repudiandae.</div>
<div class="border w-50 ms-3 p-1">Lorem ipsum dolor sit amet consectetur
adipisicing elit. A harum minus, debitis dolores deleniti facere
exercitationem iusto cum, explicabo quis, quas numquam enim fuga molestias?
Libero blanditiis alias nobis repudiandae.</div>
<div class="border w-50 mt-3 p-1">Lorem ipsum dolor sit amet consectetur
adipisicing elit. A harum minus, debitis dolores deleniti facere
exercitationem iusto cum, explicabo quis, quas numquam enim fuga molestias?
Libero blanditiis alias nobis repudiandae.</div>
```

Gyakorlati feladat (Microsoft Teams 20230824_1.pdf)

A feladatot a VS Code segítségével készítsd el.
Forrás: index.html
Nyisd meg a feladathoz csatolt index.html fájlt. Használd a beépített Bootstrap-osztályokat és tedd reszponzívvá az oldalt.
Állítsd át zöldre a főcímet tartalmazó oszlop háttérszínét.
A rácsrendszer első sorának tetejére alkalmazz 4 egységnyi belső margót.
A főcím szövege legyen világos színű és nagybetűs.
Az első bekezdésre függőleges irányban állíts be 3 egységnyi margót.
A kettes szintű címsorok betűit színezd át zöldre, majd függőleges irányban állíts be 3 egységnyi belső margót.
A felsorolásban, a span tárolóelemben elhelyezett elemeket formázd félkövér stílussal.
A hármas szintű alcímeket szedd világos színű, dőlt betűkkel. A háttérszínt állítsd itt is zöldre.
A hármas szintű alcímek belső margója legyen 2 egység nagyságú, míg alájuk helyezz el 5 egységnyi margót.
Az oldalon lévő képeket formázd a következő utasítások alapján:

- *Minden kép legyen reszponzív.*
- *A felsorolás mellett található képet lásd el zöld színű szegéllyel, majd kerekítsd le kör alakúra. Az elem felett állíts be 3 egységnyi margót.*
- *Az oldal alján kiemelt gyógynövényekhez tartozó képek legyenek bélyegképek. Formázd őket a thumbnailekhez tartozó fehér szegéllyel.*

A megjelenés módosítása a Bootstrap beépített osztályaival

A megjelenés módosítása

A megjelenítési mód, azaz a display tulajdonság a következő beépített osztályokkal módosítható:

- d-block - blokkszintű megjelenés
- d-inline - inline (sorbeli) megjelenés
- d-none - a megjelenés letiltása

A megjelenítési mód beállítását kiegészíthetjük az egyes eszközcsoportok kódjaival (sm, md, lg, xl, xxl), így lehetőség van arra, hogy az eltérő képernyőméreteken más és más megjelenítési módot alkalmazzunk.

Példa a megjelenési mód beállítására

```
<p class="d-inline">Lorem ipsum</p>
<p class="d-inline">dolor sit amet.
  <span class="d-block">Navigare est.</span>
</p>
<p class="d-none">Titkos.</p>
```

Példa az eszközcsoporttól függő megjelenítés beállítására

Az alábbi kód eredményeként a kép extra kis és kis képernyőkön nem jelenik meg, míg ennél nagyobb viewportszélességű eszközökön blokkszintű elemként viselkedik.

```
<div>
  <img src="varos.jpg" alt="Város"
  class="d-none d-md-block">
  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.</p>
</div>
```

A lebegtetés

A beépített osztályokkal a médiaelemek lebegtetése is megvalósítható:

- float-start - balra lebegtetés
- float-end - jobbra lebegtetés
- float-none - nincs lebegtetés

A lebegtetett elemeknél be kell állítani a margót is, a megfelelő távolság érvényesüljön köztük és az őket körülvevő többi elem között.
A kódot a korábban megismert eszközcsoportok neveivel is ki lehet egészíteni.

```
<div>
    <img src="jegmadar.jpg" alt="jégmadár" class="float-start" width="300">
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quisquam
    ratione perspiciatis iusto voluptatum. Eveniet qui minus cupiditate
    beatae illum, harum quas inventore magnam eos commodi recusandae ullam
    in, ipsa facilis!</p>
</div>
```

```
<div>
    <img src="jegmadar.jpg" alt="jégmadár" class="float-end" width="300">
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quisquam
    ratione perspiciatis iusto voluptatum. Eveniet qui minus cupiditate
    beatae illum, harum quas inventore magnam eos commodi recusandae ullam
    in, ipsa facilis!</p>
</div>
```

```
<div>
    <img src="jegmadar.jpg" alt="jégmadár" class="float-md-start me-3"
    width="300">
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quisquam
    ratione perspiciatis iusto voluptatum. Eveniet qui minus cupiditate
    beatae illum, harum quas inventore magnam eos commodi recusandae ullam
    in, ipsa facilis!</p>
</div>
```

Az overflow kezelése

A lebegtetett elemeknél gyakran előfordul az overflow, azaz az elemek esetenként kilóghatnak a tartalmazóelemből.
Ezt a clearfix beépített osztály tárolóelemre való alkalmazásával lehet megakadályozni.

```
<!-- Az elem kilóg a tárolóelemből -->
<div class="border border-dark">
    <img src="jegmadar.jpg" alt="jégmadár" class="float-start me-3"
    width="300">
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Autem eius
    delectus esse officia eveniet tempora, consectetur perspiciatis
    cupiditate sequi sit, beatae maxime ad nulla blanditiis saepe rem odio,
    animi quisquam.</p>
</div>
```

```
<!-- A clearfix osztály alkalmazása -->
<div class="border border-dark clearfix">
    <img src="jegmadar.jpg" alt="jégmadár" class="float-start me-3"
    width="300">
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Autem eius
    delectus esse officia eveniet tempora, consectetur perspiciatis
    cupiditate sequi sit, beatae maxime ad nulla blanditiis saepe rem odio,
    animi quisquam.</p>
</div>
```

A táblázatok formázása

Ahhoz, hogy a Bootstrap stílusai a táblázatok formázásánál is érvényre jussanak, a <table> taget először a table osztályba kell helyezni. Ezután a következő formázási lehetőségek érhetők el:

- table-striped - váltakozó színben megjelenő táblázatsorok (csíkozott megjelenés)
- table-bordered - szegéllyel körülvett táblázat
- table-dark - sötét hátterű táblázat

Példa a table-striped osztály alkalmazására

```
<table class="table table-striped">
<caption>A szerencsés nyertesek</caption>
<tr>
  <th>Név</th>
  <th>Lakhely</th>
</tr>
<tr>
  <td>Kiss Ágnes</td>
  <td>Budapest</td>
</tr>
<tr>
  <td>Kővágó László </td>
  <td>Kecskemét</td>
</tr>
<tr>
  <td>Nagy Tibor</td>
  <td>Debrecen</td>
</tr>
</table>
```

Példa a table-bordered osztály alkalmazására

```
<table class="table table-bordered">
<caption>A szerencsés nyertesek</caption>
<tr>
  <th>Név</th>
  <th>Lakhely</th>
</tr>
<tr>
   <td>Kiss Ágnes</td>
   <td>Budapest</td>
</tr>
<tr>
   <td>Kővágó László </td>
   <td>Kecskemét</td>
</tr>
<tr>
   <td>Nagy Tibor</td>
   <td>Debrecen</td>
</tr>
</table>
```

Példa a table-dark osztály alkalmazására

```
<table class="table table-dark">
<caption>A szerencsés nyertesek</caption>
<tr>
   <th>Név</th>
   <th>Lakhely</th>
</tr>
<tr>
   <td>Kiss Ágnes</td>
   <td>Budapest</td>
</tr>
<tr>
   <td>Kővágó László </td>
   <td>Kecskemét</td>
</tr>
<tr>
   <td>Nagy Tibor</td>
   <td>Debrecen</td>
</tr>
</table>
```

A táblázat, a táblázat sorainak és celláinak színezése

A teljes táblázat, a táblázat sorai vagy cellái színének beállításánál használhatjuk a szövegszíneknél megismert kulcsszavakat.

<table class="table table-primary">...</table>
<table class="table table-secondary">...</table>
<table class="table table-success">...</table>
<table class="table table-danger">...</table>
<table class="table table-warning">...</table>
<table class="table table-info">...</table>
<table class="table table-light">...</table>
<table class="table table-dark">...</table>

```
<table class="table table-primary">
    <tr>
        <th>Név</th>
        <th>Lakhely</th>
    </tr>
    <tr>
        <td>Kiss Ágnes</td>
        <td>Budapest</td>
    </tr>
    <tr>
        <td>Kővágó László</td>
        <td>Kecskemét</td>
    </tr>
    <tr>
        <td>Nagy Tibor</td>
        <td>Debrecen</td>
    </tr>
</table>
```

<tr class="table-primary">...</tr>
<tr class="table-secondary">...</tr>
<tr class="table-success">...</tr>
<tr class="table-danger">...</tr>
<tr class="table-warning">...</tr>
<tr class="table-info">...</tr>
<tr class="table-light">...</tr>
<tr class="table-dark">...</tr>

```
<table class="table">
    <tr class="table-primary">
        <th>Név</th>
        <th>Lakhely</th>
    </tr>
    <tr>
        <td>Kiss Ágnes</td>
        <td>Budapest</td>
    </tr>
    <tr>
        <td>Kővágó László</td>
        <td>Kecskemét</td>
    </tr>
    <tr>
        <td>Nagy Tibor</td>
        <td>Debrecen</td>
    </tr>
</table>
```

```
<tr>
  <td class="table-primary">...</td>
  <td class="table-secondary">...</td>
  <td class="table-success">...</td>
  <td class="table-danger">...</td>
  <td class="table-warning">...</td>
  <td class="table-info">...</td>
  <td class="table-light">...</td>
  <td class="table-dark">...</td>
</tr>
```

```
<table class="table">
    <tr>
        <th>Név</th>
        <th>Lakhely</th>
    </tr>
    <tr>
        <td class="table-primary">Kiss Ágnes</td>
        <td>Budapest</td>
    </tr>
    <tr>
        <td>Kővágó László</td>
        <td>Kecskemét</td>
    </tr>
    <tr>
        <td>Nagy Tibor</td>
        <td>Debrecen</td>
    </tr>
</table>
```

Csíkozott megjelenésű oszlopokat is készíthetünk a table-striped-columns osztály segítségével. Az összevont cellák esetében viszont mindkét cella egyforma színű lesz.

```
<table class="table table-striped-columns">
    <thead>
      <tr>
        <th scope="col">#</th>
        <th scope="col">First</th>
        <th scope="col">Last</th>
        <th scope="col">Handle</th>
      </tr>
    </thead>
```

```
    <tbody>
      <tr>
        <th scope="row">1</th>
        <td>Mark</td>
        <td>Otto</td>
        <td>@mdo</td>
      </tr>
      <tr>
        <th scope="row">2</th>
        <td>Jacob</td>
        <td>Thornton</td>
        <td>@fat</td>
      </tr>
      <tr>
        <th scope="row">3</th>
        <td colspan="2">Larry the Bird</td>
        <td>@twitter</td>
      </tr>
    </tbody>
</table>
```

A table-hover osztály használata esetén a tbody elemben elhelyezkedő sorok esetében más háttérszínnel jelenik meg az a sor, amelyik felett az egérkurzor tartózkodik.

```
<table class="table table-hover">
```

Az aktív sor vagy cella kiemeléséhez használhatjuk a table-active osztályt.

```
    <tbody>
      <tr class="table-active">
        <th scope="row">1</th>
        <td>Mark</td>
        <td>Otto</td>
        <td>@mdo</td>
      </tr>
      <tr>
        <th scope="row">2</th>
        <td class="table-active">Jacob</td>
        <td>Thornton</td>
        <td>@fat</td>
      </tr>
      <tr>
```

A táblázatok szegélyezése

A táblázatok szegélyezésénél használhatjuk a szövegszíneknél megismert kulcsszavakat, pl. primary, secondary, stb.

```
<table class="table table-bordered border-primary">
```

Ha szegélynélküli táblázatot szeretnénk, használhatjuk a table-borderless osztályt.

```
<table class="table table-borderless">
```

Kis táblázatok

Kisebb méretű táblázatot készíthetünk a table-sm osztály használatával. A cellák belső margója lesz kisebb.

```
<table class="table table-sm">
```

Csoport elválasztó használata

A <thead>, <tbody> és <tfoot> elemek között csoport elválasztó szegélyt készíthetünk a table-group-divider osztály segítségével.

```
<tbody class="table-group-divider">
```

A táblázat címének elhelyezése a táblázat felett

```
<table class="table caption-top">
    <caption>List of users</caption>
    <thead>
```

Gyakorlati feladat (Microsoft Teams 20230907_1.pdf)

A feladatot a VS Code segítségével készítsd el.
Forrás: index.html
Nyisd meg a feladathoz csatolt index.html fájlt. Formázd az oldalt a beépített Bootstrap-osztályokkal.
Szedd világos színnel a főcím betűit, a hátterét állítsd türkiz (bg-info) színűre. A főcím paddingje legyen 2 egység nagyságú. A cím szövegét igazítsd középre.
Az első képet igazítsd jobbra úgy, hogy a szöveg balról vegye körbe a képet. A kép bal oldali margóját állítsd 3 egységnyire. Kerekítsd le a kép sarkait. A kép szélessége legyen a szülőelem szélességének fele.
A kettes szintű alcímek szövegszínét állítsd türkizre (text-info).
A második képet úsztasd balra. A kép jobb oldali margóját állítsd 3 egységnyire. A kép szélessége legyen a szülőelem szélességének 25%-a. Oldd meg, hogy a kép az extra kis méretű eszközöktől felfelé ne jelenjen meg az oldalon, míg a közepes méretű eszközöktől felfelé inline elemként szerepeljen. Kerekítsd le a kép sarkait.
Érd el, hogy a harmadik kép minden eszközön vízszintesen középre igazított, blokkszintű elemként jelenjen meg. A kép szélessége legyen a szülőelem szélességének 50%-a. Kerekítsd le a kép sarkait.
A table HTML-elemre alkalmazd a table class selectort, és formázd csíkozottra a táblázatot.
A táblázat fejlécében lévő tartalmat igazítsd vízszintesen középre.

A Bootstrap gyakran használt komponensei

A Bootstrap-komponensek gyűjteménye

Mik a Bootstrap-komponensek?

A Bootstrap-komponensek lényegében nem mások, mint a weboldalak fejlesztése során felhasználható építőelemek. Többek között a következők tartoznak ide:

- jumbotron: óriás infótábla
- alert: figyelmeztető üzenet
- badge: jelvény
- button: gomb, gombcsoport
- dropdown: legördülő menü
- carousel: lapozható médiaobjektum

A használható komponensek és azok mintakódjai a Bootstrap dokumentációban elérhetőek, onnan átmásolhatók.
A mintakódokban gyakran szerepel a role paraméter. Ez az adott elem funkcióját jelöli és az akadálymentesítésben játszik szerepet.

A Bootstrap-komponensek használata

A jumbotron

A jumbotron komponenssel nagy méretű információs táblák készíthetők a weboldalakra. A tartalmazóelemet kell a jumbotron osztályba helyezni, majd ezt követően lehet a táblát ellátni szöveges tartalommal, illetve gombokkal.

Példa a jumbotron komponens alkalmazására

```
<div class="jumbotron">
  <h1>Fontos változások!</h1>
  <p class="lead">A portál felhasználói feltételei
módosultak.</p>
  <a class="btn btn-primary btn-lg"
  href="hirek.html" role="button">
  További információ</a>
</div>
```

Az alert

Az alert komponenssel üzeneteket (figyelmeztetés, veszély, visszajelzés) lehet megjeleníteni.
Az üzenetek a Bootstrap témájához kapcsolódó alapszínekkel jelennek meg, így az üzenet típusától függően a következő beállítások közül lehet választani:

- alert-primary: elsődleges figyelmeztetés (kék)
- alert-secondary: másodlagos figyelmeztetés (szürke)
- alert-success: visszajelzés a sikeres műveletről (zöld)
- alert-danger: figyelmeztetés a veszélyes műveletre (piros)
- alert-warning: figyelmeztető üzenet (sárga)
- alert-info: alapvető információ (türkiz)
- alert-light: világos háttérszínű figyelmeztetés (fehér)
- alert-dark: sötét háttérszínű figyelmeztetés (fekete)

Példa az alert komponens alkalmazására

```
<div class="alert alert-primary" role="alert">
  Elsődleges figyelmeztetés
</div>
<div class="alert alert-secondary" role="alert">
  Másodlagos figyelmeztetés
</div>
<div class="alert alert-success" role="alert">
  Visszajelzés a sikeres műveletről
</div>
<div class="alert alert-danger" role="alert">
  Figyelmeztetés a veszélyes műveletre
</div>
<div class="alert alert-warning" role="alert">
  Figyelmeztető üzenet
</div>
<div class="alert alert-info" role="alert">
  Alapvető információ
</div>
<div class="alert alert-light" role="alert">
  Világos háttérszínű figyelmeztetés
</div>
<div class="alert alert-dark" role="alert">
  Sötét háttérszínű figyelmeztetés
</div>
```

Lapozás

```
<nav aria-label="Page navigation example">
  <ul class="pagination">
    <li class="page-item"><a class="page-link" href="#">Previous</a></li>
    <li class="page-item"><a class="page-link" href="#">1</a></li>
    <li class="page-item"><a class="page-link" href="#">2</a></li>
    <li class="page-item"><a class="page-link" href="#">3</a></li>
    <li class="page-item"><a class="page-link" href="#">Next</a></li>
  </ul>
</nav>
```

https://getbootstrap.com/docs/5.3/components/pagination/#overview

Legördülő menü

```
<div class="dropdown">
  <button class="btn btn-secondary dropdown-toggle" type="button" data-bs-toggle="dropdown
    Dropdown button
  </button>
  <ul class="dropdown-menu">
    <li><a class="dropdown-item" href="#">Action</a></li>
    <li><a class="dropdown-item" href="#">Another action</a></li>
    <li><a class="dropdown-item" href="#">Something else here</a></li>
  </ul>
</div>
```

https://getbootstrap.com/docs/5.3/components/dropdowns/#overview

Menü

https://getbootstrap.com/docs/5.3/components/navbar/#nav

Képlapozó

https://getbootstrap.com/docs/5.3/components/carousel/#how-it-works

Jelvények

https://getbootstrap.com/docs/5.3/components/badge/

Accordion

https://getbootstrap.com/docs/5.3/components/accordion/

Breadcrumb

https://getbootstrap.com/docs/5.3/components/breadcrumb/

Kártyák

https://getbootstrap.com/docs/5.3/components/card/

Gombok

https://getbootstrap.com/docs/5.3/components/buttons/

Collapse

https://getbootstrap.com/docs/5.3/components/collapse/

Lista csoportok

https://getbootstrap.com/docs/5.3/components/list-group/

Modal

https://getbootstrap.com/docs/5.3/components/modal/