

Oldalszerkezet kialakítása CSS segítségével



A webfejlesztés elmúlt éveiben sok zsákutcát megjártak a szakma művelői. A keretek indokolatlan használatánál talán már csak a táblázatos oldalkialakítás okozott több bonyodalmat és felesleges munkát. Régi „jól bevált” és sokak által még a mai napig is a legjobbnak ítélt módszer a táblázatok használata a menük oldalra, a fejléc felülre, a lábléc alulra stb. pozicionálásához. A módszer rövidesen kezdett elburjánzani, megjelentek a 2-3-4 szinten egymásba ágyazott táblázatok, az egyesített cellák, a csak térköz kialakításához létrehozott sorok és oszlopok, vagy ami még ennél is rosszabb, a „távtartó gif-ek”. Aki már megpróbált egyszer egy ilyen szerkezetű oldalt megérteni, esetleg a dizájnt megváltoztatni, lehet, hogy „néhány ősz hajszállal gazdagodott”. A táblázatos elrendezéssel ugyanis

- Nehezen átlátható forráskódot eredményez. (Nehezen javítható, bővíthető, módosítható).
- A táblázatok csak a teljes táblázat, és annak minden eleme betöltése után tudják a böngészők helyesen megjeleníteni. (Lassú a megjelenítés.)
- A táblázatok valódi célja az adatok rendezett formában történő megjelenítése.
- A táblázatos szerkezet a képernyő felolvasók dolgát is megnehezíti. (Celláról cellára ugrálna.)
- A táblázatos felépítés statikus szerkezetet eredményez, ami nem módosítható stíluslapokkal.

A CSS2 megjelenése óta semmi szükség az ilyen elavult és értelmetlen technikákra.

A CSS2 megjelenése lehetővé tette, hogy a weboldal szerkezetének (website layout) kialakításához általános tárolóelemeket (div) és stíluslapokat (CSS) használjunk. Ezek jellemzői:

- **Áttekinthető** forráskódot eredményez.
- Az egymás mellett található általános **tároló elemek megjelenítésének sorrendje nem a képernyőn való helyzetüktől függ**. A forráskód kialakításánál a hasznos tartalom előre helyezésével elérhetjük, hogy a felhasználó már tudja olvasni azt, miközben az oldal kevésbé fontos elemei még töltődnek.
- A tároló elemek helyzete és a tartalom **megjelenítése stíluslapokkal módosítható**, weboldalaink megjelenését stíluslapok segítségével gyorsan és látványosan megváltoztathatjuk. Akár a felhasználó is kiválaszthatja a neki leginkább tetsző megjelenítési formát.
- A tároló elemekkel kialakított szerkezet **dinamikusan illeszkedhet a képernyő felbontásához**. Napjainkban már nem elegendő egy weblapot egy szabványos felbontáshoz optimalizálni. **Fontos, hogy a mobil eszközök kis képernyőjén és a egyre nagyobb felbontást lehetővé tévő megjelenítő eszközökön is jól mutasson.** Lehetőség szerint egy nagy felbontású monitorral rendelkező látogatónak ne kelljen a képernyőt görgetni, miközben két oldalt széles nem használt csíkok éktelenkednek.

1. Az egyoszlopos elrendezés

Az egyoszlopos elrendezés esetén az egyes egységeket tároló elemeket egy fő tároló elembe helyezzük el. Ennek a tároló elem szélességének és magasságának megadásával meghatározzuk az oldal megjelenítésére használt terület méretét. A fő szerkezeti elemeknél így már elegendő csak a magasságot megadni.

Az egyoszlopos elrendezés esetén **4 fő szerkezeti elemet** használunk:

1. Fejléc (header)

Rendszerint az oldal vagy cég logóját és nevét tartalmazza, esetleg reklámot. Az oldal bal felső sarkának kiemelt szerepe van, hiszen a felhasználók rendszerint ott kezdik az oldal olvasását.



2. Menüsor (navigation)

A legfontosabb navigációs elemek (menüpontok) gyűjtőhelye.

3. Tartalom (content)

A konkrét oldalon közreadott egyedi tartalom.

4. Lábléc (footer)

Céginfó, jogi információk, az oldal készítőjének linkje a leggyakoribb elemei.

Az előző oldalon bemutatott minta szerkezetének leírását és a formázáshoz alkalmazott CSS-kódokat az alábbiak tartalmazzák:

a szerkezet leírása (HTML):

```
<div id="tarolo">
  <div id="fejlec" >
    Az oldal/cég/termék neve és alapadatai
  </div>
  <div id="menu">
    Termékek | Vásárlás | Kapcsolatok
  </div>
  <div id="tartalom">
    Az oldal tartalma
  </div>
  <div id="lablec">
    Céginfó, jogi információk
  </div>
</div>
```

az alkalmazott formázókódok (CSS):

```
#tarolo { height: 600px;
          width: 600px; }

#fejlec { background-color: yellow;
          text-align: center;
          height: 30px; }

#menu { background-color: lightgreen;
        height: 50px;
        text-align: center; }

#tartalom { background-color: lightblue;
            height: 400px; }

#lablec { background-color: yellow;
          text-align: center;
          height: 30px; }
```

2. Kétoszlopos elrendezések

A kétoszlopos elrendezés esetén szintén egy fő tároló elembe helyezünk el az egyes egységeket tároló elemeket és **5 fő szerkezeti elemet** használunk:

1. Fejléc (header)

Rendszerint az oldal vagy cég logóját és nevét tartalmazza, esetleg reklámot. Az oldal bal felső sarkának kiemelt szerepe van, hiszen a felhasználók rendszerint ott kezdik az oldal olvasását.

2. Menüsor (navigation)

A legfontosabb navigációs elemek (menüpontok) gyűjtőhelye.

3. Menüsáv

Itt tehetjük elérhetővé a webhely belső oldalait.

4. Tartalom (content)

A konkrét oldalon közreadott egyedi tartalom.

5. Lábléc (footer)

Céginfó, jogi információk, az oldal készítőjének linkje a leggyakoribb elemei.



Most a „Menüsáv” és a „Tartalom” egymás mellé kerül, amelyet többféle módszerrel is elérhetünk (például pozicionálás használatával vagy úsztatás beállításával). Ha a tartalmi rész túl hosszú, akkor arra a blokkra még a gördítősávokat is meg kell jelenítenünk (túlfolyás beállítása).

az alkalmazott formázókódok (CSS):

Az oldal fő tárolóeleme a „container” osztályú elem, annak szélessége alapvetően az egész oldal szélességét meghatározza, most kitölti a rendelkezésre álló helyet (100%), azaz a teljes ablakszélességet. A „margin” és a „border” tulajdonságok csak erre a dobozra lesznek hatással, de a „line-height” tulajdonság már öröklődni fog az összes tartalmazott dobozra. Az egységes színeket, betűtípusokat stb. is itt érdemes meghatározni.

A „header” és a „footer” elemek egységes megjelenítését szolgálja a közös stílusmegadás (a két kijelölő vesszővel van meghatározva). Az utolsó sorban lévő „clear: left” azt szolgálja, hogy a két doboz („header”, „footer”) bal oldalán ne lehessen lebegő (úsztatott) elem. (Használható még a „clear: both” is, ezzel mindkét oldalon megakadályozzuk az esetleges úszást.)

Hogyan lehet két elemet egymás mellé helyezni? Alaphelyzetben a „left” és a „content” dobozok egymás alá kekrülnének. Ha viszont egymás mellé szeretnénk helyezni azokat, akkor például beállíthatunk rájuk úsztatást („float: left”). Ezzel lehetővé tesszük, hogy az egymás alatt definiált elemek a definiálás sorrendje szerint egymás mellé kerüljenek úgy, hogy a bal oldalon kezdődik az első, mellette a következő (és így tovább, ha több elemet is definiáltunk).

Fontos, hogy az úsztatás beállításakor az egymás mellé került elemek ne csússzanak „egymás alá”. Ezt kétféle megoldással érhetjük el: vagy az összes elemre beállítjuk az úsztatást, vagy minden nem úsztatott, de egymás mellé kerülő elemre nagyobb bal margót határozzunk meg.

Ha egy blokk tartalma nagyobb, mint a definiált doboz (pl. div) szélessége és/vagy magassága, akkor a tartalom alapértelmezés szerint azon túlcsordulhat. Helyette jó lenne, ha gördíthetővé válna a blokk, ezért állítottuk be az „overflow: auto” tulajdonságot is. Fontos még, hogy az egymás mellé kerülő elemek legyen azonos magasságúak, azaz alkalmazzuk ugyanazt a „height” értéket.

```
div.container {
  width: 100%;
  margin: 0px;
  border: 1px solid gray;
  line-height: 150%;
}
```

```
div.header, div.footer {
  padding: 0.5em;
  color: white;
  background-color: gray;
  clear: left;
}
```

```
div.left {
  float: left;
  width: 160px;
  height: 400px;
  margin: 10px;
  padding: 1em;
  background-color: yellow;
}
```

```
div.content {
  background-color: lime;
  margin:
    10px 10px 10px 210px;
  padding: 1em;
  height: 400px;
  overflow: auto;
}
```



A táblázatos oldalkialakítási módszer után a tisztán CSS-re épülő megoldás logikája furcsa, nehézkes lehet. Hosszútávon azonban meg fogja hálálni a befektetett energiát.

A többoszlopos elrendezésnél talán az szokta a legtöbb nehézséget okozni, hogy a táblázat celláinál megszokott háttérbeállítások itt nem úgy működnek, hiszen itt nem két egyforma magasságú celláról van szó. Az ilyen jellegű probléma esetén van egy egyszerű megoldás: az egymás mellé kerülő dobozokat egy közös tartalmazó dobozba helyezzük és a háttér ügyes beállításával már el lehet érni azt a hatást, mint amit szeretnénk. (Nem az egymás melletti dobozok, hanem az őket közösen tartalmazó doboz háttérét kell beállítanunk.)

Gyakran az összes (pl. 2-3-4) egymás mellé kerülő dobozt úsztatjuk. Ekkor ugyanis nem szükséges a nagy margó alkalmazása. Hátránya viszont, hogy a dobozok előbb említett „közös háttérét” nehezebb formázni.

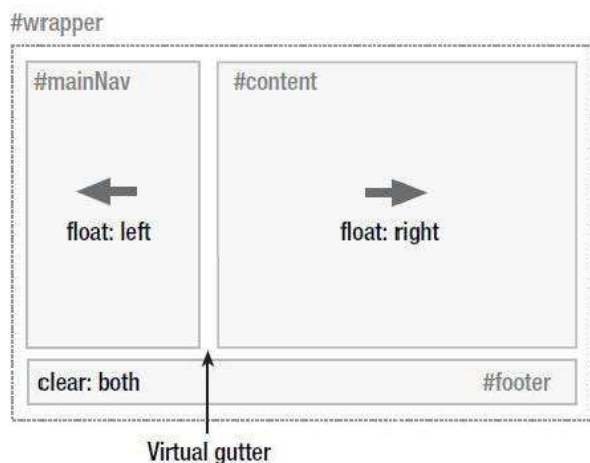
A többoszlopos elrendezésű oldalak esetén alkalmazott elvek

- Az egész oldal tartalmát egyetlen tároló elembe helyezzük el, amelynek margóit 0 értékkel határozzuk meg. Itt adjuk meg az elemek közös tulajdonságait (pl. betűtípus, -méret).
- Az egymás mellé kerülő elemeket a megfelelő irányba úsztatjuk. Magasságukat egyformára, szélességüket pedig úgy állítjuk be, hogy az összegük legfeljebb akkora legyen, mint a közös tároló elemük szélessége.

- Ha egy elem magassága kisebb, mint (várható) tartalma, akkor a blokkra érdemes az automatikus gördítősáv-megjelenítést is meghatározni.
- A fejléc és a lábléc esetén az úsztatást mindkét irányban letiltjuk.

Példa a kétszlopos elrendezés szerkeztére és megjelenésére:

a szerkezeti felépítés (HTML)



a megjelenés (stílusokkal formázva)



3. Többszlopos elrendezések

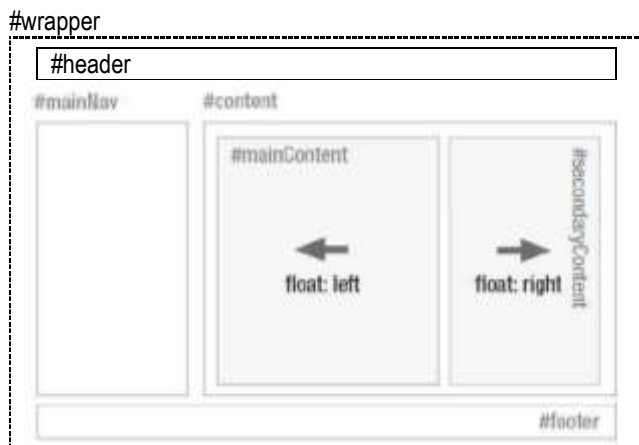
Háromszlopos elrendezés – 1. változat

Az alábbi mintában egy 3-oszlopos elrendezési formára láthat példát, amelyben a tartalmi rész a jobb oldalon található és két további részt tartalmaz. A példaoldalon az alábbi beállításokat használtuk:

- #wrapper** (az összes elemet tároló elem): szélessége 800px, margói: alul-felül 0 a többi auto
- #mainNav** (navigációs elem): szélessége 20px, balra úsztatva
- #content** (ebben helyeztük el a tartalmi dobozokat): szélessége 600px, jobbra úsztatva
- #mainContent** (a fő tartalmi rész): szélessége 350px, balra úsztatva
- #secondaryContent** (másodlagos tartalmi rész): szélessége 250px, jobbra úsztatva
- #header, #footer** (): az úsztatás mindkét oldalon le van tiltva, magassága meg van határozva

a szerkezeti felépítés (HTML):

a megjelenés (stílusokkal formázva):



Ha olyan elrendezést szeretnénk létrehozni, hogy a szélessége mindig az aktuális ablakmérethez igazodjon (**liquid / fluid layout**), akkor az egyes elemek képpontban megadott szélességi értékei helyett azokat a szülőelemhez viszonyítva, %-ban kell meghatároznunk. A fenti példa automatikusan változó szélességű változata így az alábbi stílusulajdonságok beállítását igényli:

#wrapper (az összes elemet tároló elem): szélessége 90%, margói: alul-felül 0 a többi auto

#mainNav (navigációs elem): szélessége 20%, balra úsztatva

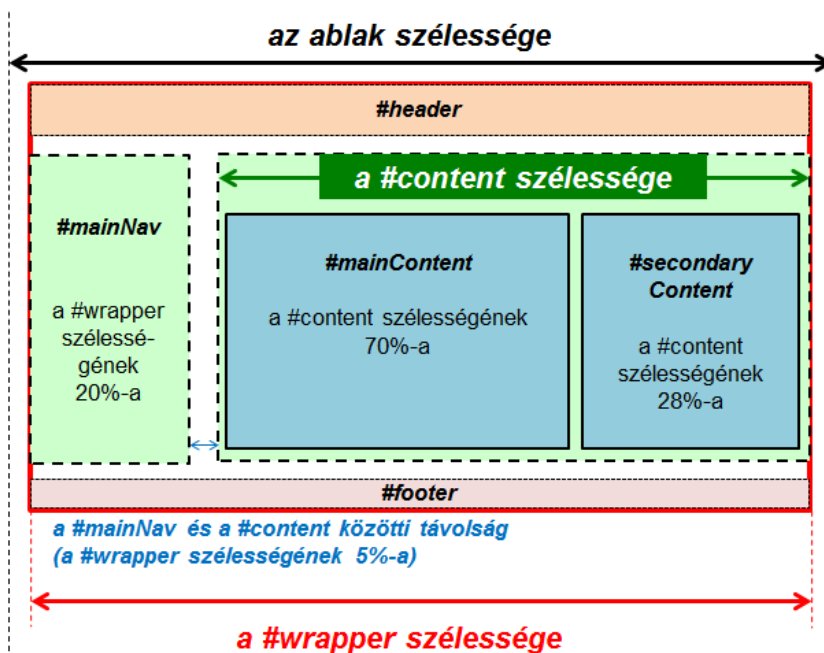
#content (ebben helyeztük el a tartalmi dobozokat): szélessége 75%px, jobbra úsztatva

#mainContent (a fő tartalmi rész): szélessége 70%, balra úsztatva

#secondaryContent (másodlagos tartalmi rész): szélessége 28%, jobbra úsztatva

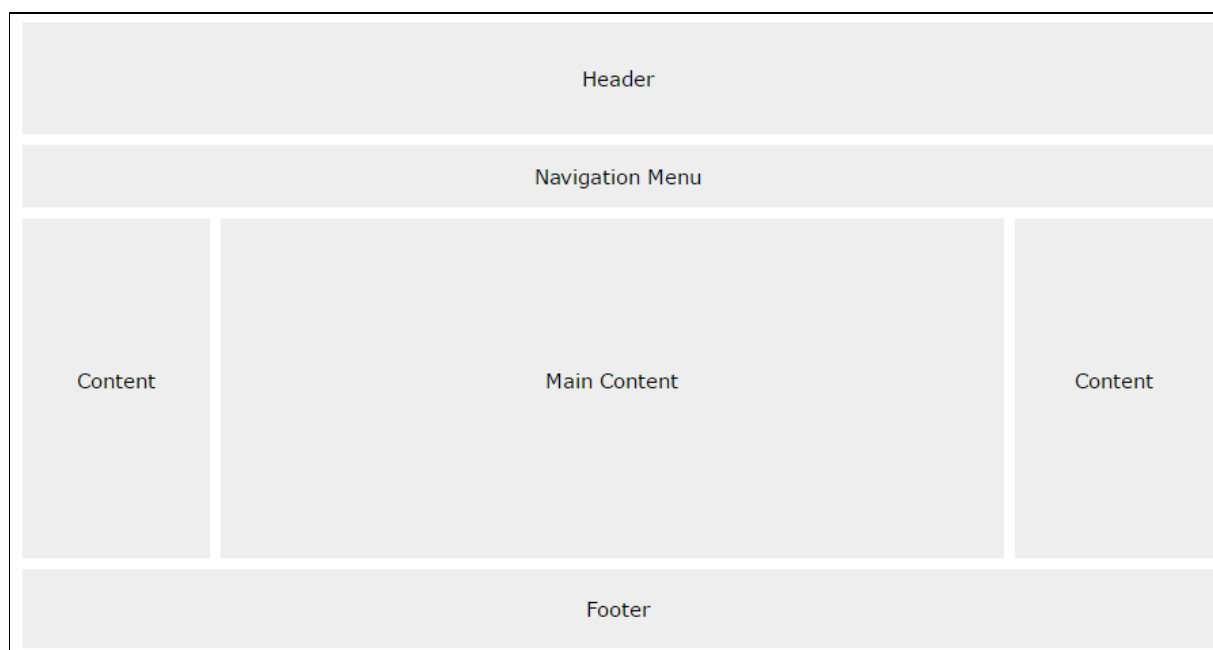
Néhány kiegészítés, magyarázat:

- A #wrapper elem a body 90%-át foglalja el.
- A #wrapper fő tároló elemen belül a #mainNav 20%, a #content doboz 75% nagyságú a szülőeleméhez, azaz a #wrapper-hez képest, így 5%-nyi rész lesz a két elem között ($100\% - 20\% - 75\% = 5\%$).
- A #content tároló elemében lévő #mainContent a szülőelem (azaz a #content) 70%-át, a másik rész, a #secondaryContent pedig 28%-át foglalja el, így a két elem között a #content doboz 2%-ának megfelelő hely lesz ($100\% - 70\% - 28\% = 2\%$).



Háromoszlopos elrendezés – 2. változat

Láttuk már, hogy a webhely gyakran fejlécre, menükre, tartalomra és láblécre oszlik, amelyeket nagyon sokféle módon lehet elhelyezni, így rengeteg különféle elrendezési minta közül válogathatunk. Az itt látható struktúra azonban az egyik leggyakoribb, így ezt mutatjuk be egyik kicsit részletesebben.



Header (fejléc)

A fejléc általában a weboldal tetején található (vagy közvetlenül a felső navigációs menü alatt) és gyakran tartalmaz egy logót vagy a webhely nevét.

A fejléc formátumának beállításai CSS-ben:

- általában tartalmaz valamilyen háttérszínt vagy háttérképet (background-color, -image),
- a belső tartalom és az esetleges szegély távolabb van elhelyezve (padding),
- a tartalma igazított (vízszintesen/függőlegesen középre).

formázott megjelenés



a szerkezeti felépítés (HTML)

```
<div class="header">
  <h1>Header</h1>
</div>
```

Navigation Bar (navigációs sáv)

A navigációs sáv tartalmazza azon linkek listáját, amelyek segítik a látogatókat a webhelyen való navigációban.

A navigációs sáv formátumának beállításai CSS-ben:

- a sáv elemeit (azaz a ténylege linkeket) egy tároló elem (<div>) tartalmazza, amelyben a túlfolyás (overflow) rendszerint le van tiltva / automatikus gördítősávokkal van megjelenítve,
- az egyes elemek megvalósíthatók közvetlenül sorszintű elemekkel (<a>) vagy blokkszintű elemeknek (<div> vagy gyakran listáknak) sorszintűvé transzformálásával,
- az egyes oldalakra mutató linkek felülete gombként jelenik meg,
- a linkek „hover” állapotához egyedi formátumot rendelünk.

a formázott megjelenés:



a szerkezeti felépítés (HTML):

```
<div class="topnav">
  <a href=...>Link</a>
  <a href=...>Link</a>
  <a href=...>Link</a>
</div>
```

az alkalmazott formázókódok (CSS):

a navigációs sávot tartalmazó elem:

```
.topnav {
  overflow: hidden;           /* ha túlfolyás alakul ki, akkor az legyen elrejtve */
  background-color: silver;   /* a navigációs rész háttérszíne (ezzel egyező színűek lesznek a linkdobozok is) */
}
```

a navigációs sávban lévő linkek alapvetően egységesen (ugyanúgy) jelenjenek meg:

```
.topnav a {
  display: block;             /* blokkszerűen legyenek a sorszintű <a> elemek elrendezve */
  float: left;                /* balra úsztatva legyen az összes linkelem */
  color: white;                /* a linkek szövegének betűszíne */
  text-align: center;          /* vízszintesen középre igazítva legyenek a blokkokban a linkszövegek */
  padding: 14px 16px;          /* legyen távolítás a blokkokban */
  text-decoration: none;       /* ne legyen aláhúzva a link (elegánsabb aláhúzás nélkül...) */
}
```

ha egy linkre rámutatunk, akkor változzon meg a megjelenése:

```
.topnav a:hover {
  background-color: white;     /* más háttérszínnel jelenjen meg */
  color: black;                /* más betűszínnel jelenjen meg */
}
```

A navigációs sáv elemeit nagyon gyakran lista segítségével alakítjuk ki. Mivel a listák elemei külön blokkszintű sorokban (li) jelennek meg, ezért azokat a „display” tulajdonság alkalmazásával sorszintűvé kell alakítani úgy, hogy azokhoz a későbbiekben olyan tulajdonságokat lehessen hozzárendelni, mintha blokkszintűek lennének (pl. magasság, szélesség). A következő példában ezt mutatjuk be.

a formázott megjelenés:**a szerkezeti felépítés (HTML):**

```
<ul class="nav">
  <li><a href="#home">Home</a></li>
  <li><a href="#about">About Us</a></li>
  <li><a href="#clients">Our Clients</a></li>
  <li><a href="#contact">Contact Us</a></li>
</ul>
```

a lista**alapértelmezett megjelenése:**

- [Home](#)
- [About Us](#)
- [Our Clients](#)
- [Contact Us](#)

az alkalmazott formázókódok (CSS):

az egész listára (azaz a navigációs sávra) vonatkozó tulajdonságok beállítása:

```
.nav {
  background-color: yellow;           /* a navigációs sáv hátterszínének megadása
  list-style-type: none;             /* a lista jelölőjének (azaz a • jelnek) az eltávolítása
  text-align: center;
  margin: 0;
  padding: 0;
}
```

a navigációs sávban lévő egyes listaelemek jellemzői:

```
.nav li {
  display: inline-block;             /* a blokkszintű elemek sorszintűvé alakítása
  font-size: 20px;
  padding: 20px;
}
```

a navigációs sávban lévő egyes listaelemekben található linkek formai jegyei:

```
.nav li a {
  text-decoration: none;             /* a linkek aláhúzásai eltávolítása
  color: maroon;                    /* a linkszövegek alapértelmezett kék színének megváltoztatása
  font-weight: 800;                 /* a linkszövegek vastagságának növelése
}
```

ha a navigációs sávban lévő valamelyik elemre rámutatunk, akkor annak háttérszíne megváltozik:

```
.nav li:hover { background-color: green; }
```

ha a navigációs sávban lévő valamelyik elemre rámutatunk, akkor a benne található link szövegének színe megváltozik:

```
.nav li:hover a { color: white; }
```

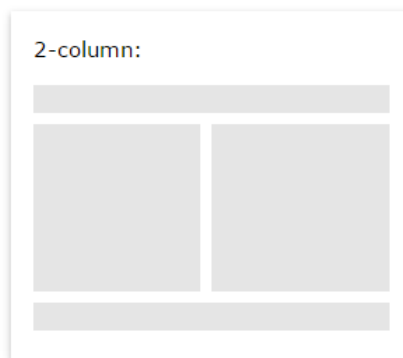
Content (tartalom)

A tartalom részben elhelyezett információk elrendezése gyakran a célfelhasználóktól függ. A leggyakoribb elrendezések az alábbiak (vagy ezek kombinálása):

- 1-oszlopos elrendezés: gyakran használt mobil böngészők számára,
- 2-oszlopos elrendezés: gyakran használatos tabletekhez és laptopokhoz,
- 3-oszlopos elrendezés: csak asztali számítógépekhez használatos.



1-oszlopos elrendezés
(mobilokhoz)



2-oszlopos elrendezés
(tabletekhez, laptopokhoz)



3-oszlopos elrendezés
(asztali számítógépekhez)

Példaként létrehozunk egy 3-oszlopos elrendezést úgy, hogy ha egy adott értéknél kisebb a megjelenítő nagyság, akkor azt 1-oszlopos elrendezésre cseréljük:

a szerkezeti felépítés (HTML):

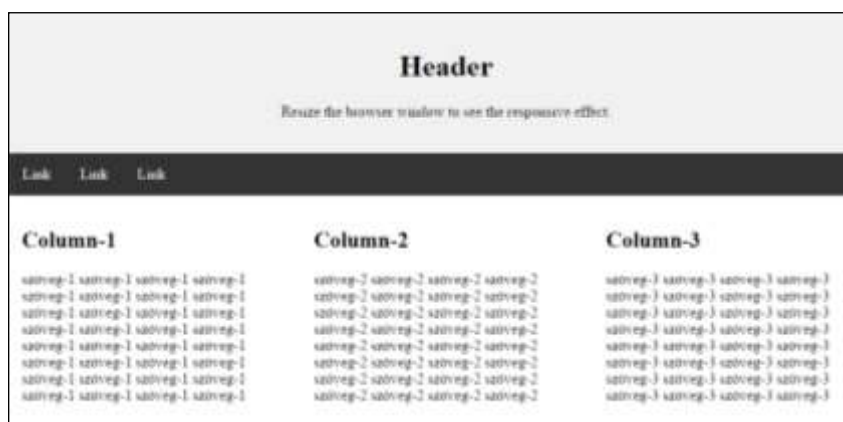
```
<div class="row">
  <div class="column">                                <!-- az oszlopokat tartalmazó elem definiálása -->
    <h2>Column-1</h2>                                <!-- az 1. oszlop tartalmának definiálása -->
    <p>szöveg-1 szöveg-1 szöveg-1 szöveg-1 szöveg-1 szöveg-1 szöveg-1 szöveg-1 szöveg-1
      szöveg-1 szöveg-1 szöveg-1 szöveg-1 szöveg-1 szöveg-1 szöveg-1 szöveg-1 szöveg-1
      szöveg-1 szöveg-1 szöveg-1 szöveg-1 szöveg-1 szöveg-1 szöveg-1 szöveg-1 szöveg-1
      szöveg-1 szöveg-1 szöveg-1 szöveg-1 szöveg-1 szöveg-1 szöveg-1 szöveg-1 szöveg-1
    </p>
  </div>
  <div class="column">                                <!-- a 2. oszlop tartalmának definiálása -->
    <h2>Column-2</h2>
    <p>szöveg-2 szöveg-2 szöveg-2 szöveg-2 szöveg-2 szöveg-2 szöveg-2 szöveg-2 szöveg-2
      szöveg-2 szöveg-2 szöveg-2 szöveg-2 szöveg-2 szöveg-2 szöveg-2 szöveg-2 szöveg-2
      szöveg-2 szöveg-2 szöveg-2 szöveg-2 szöveg-2 szöveg-2 szöveg-2 szöveg-2 szöveg-2
      szöveg-2 szöveg-2 szöveg-2 szöveg-2 szöveg-2 szöveg-2 szöveg-2 szöveg-2 szöveg-2
    </p>
  </div>
  <div class="column">                                <!-- a 3. oszlop tartalmának definiálása -->
    <h2>Column-3</h2>
    <p>szöveg-3 szöveg-3 szöveg-3 szöveg-3 szöveg-3 szöveg-3 szöveg-3 szöveg-3 szöveg-3
      szöveg-3 szöveg-3 szöveg-3 szöveg-3 szöveg-3 szöveg-3 szöveg-3 szöveg-3 szöveg-3
      szöveg-3 szöveg-3 szöveg-3 szöveg-3 szöveg-3 szöveg-3 szöveg-3 szöveg-3 szöveg-3
      szöveg-3 szöveg-3 szöveg-3 szöveg-3 szöveg-3 szöveg-3 szöveg-3 szöveg-3 szöveg-3
      szöveg-3 szöveg-3 szöveg-3 szöveg-3 szöveg-3 szöveg-3 szöveg-3 szöveg-3 szöveg-3
    </p>
  </div>
</div>
```

a kívánt formai elrendezést
a mellékelt ábra tartalmazza

az alkalmazott formázókódok (CSS):

létrehozunk három egyforma oszlopot, amelyek egymás mellé vannak úsztatva:

```
.column {
  float: left;
  width: 33.33%;
  padding: 15px;
}
```



az oszlopokat tartalmazó „row” osztályú blokk után már nem engedjük az úsztatást

```
.row:after {
  content: "";
  display: table;
  clear: both;
}
```



A @media szabályt a média-lekérdezésekben használják különböző stílusok alkalmazására a különböző médiatípusokhoz / eszközökhöz. Ennek segítségével sokféle információt ellenőrizhet: a nézetablak szélességét és magasságát, az alkalmazott eszköz szélességét és felbontását, a tájolást (pl. hogy a tablet vagy telefon tájkép vagy álló módban van-e), a felbontást. A média-lekérdezések használata egy népszerű technika a személyre szabott stíluslapok (érzékeny webdesignok) különféle típusú eszközök (asztali számítógépekre, laptopokra, táblagépekre és mobiltelefonokra) történő alkalmazásához. A média-lekérdezések segítségével azt is megadhatja, hogy bizonyos stílusok csak nyomtatott dokumentumokhoz vagy képernyőolvasókhoz legyenek használhatóak (mediatype: nyomtatás, képernyő vagy beszéd). A média-lekérdezések során a média típusa mellett a média jellemzőit is meg lehet tudni, így lehetővé válik, hogy például egyes stílusokat csak azokhoz a képernyőkhöz alkalmazzunk, amelyek nagyobbak vagy kisebbek, mint egy bizonyos szélesség.

reszponzív elrendezés kialakítása: a három oszlopot egymás alá helyezi ahelyett, hogy azok egymás mellett lennének, de csak akkor, ha a megjelenítő mérete egy adott értéket elér vagy kisebb:

```
@media screen and (max-width:600px) { /* 600px vagy kisebb felbontású képernyő esetén
  .column {
    width: 100%; /* minden oszlop szélessége a befoglaló elem szélességével egyezzen meg
  }
}
```

Tippek további szerkezetek kialakításához:

- ha 2-oszlopos elrendezést szeretnénk kialakítani, akkor mindkét oszlop szélességét állítsuk 50%-ra, ha pl. 4-oszlopot, akkor ($100\% / 4 =$) 25%-ra, stb.
- az oszlopok létrehozásának modernebb módja a CSS Flexbox alkalmazása (az IE-10 és korábbi verziói nem támogatják, ezért ebben az esetben a fentiek szerinti úsztatás („float”) tulajdonságot érdemes alkalmazni)

Egyenlőtlen oszlopokkal megvalósított elrendezések

A fő tartalom a webhely legnagyobb és legfontosabb része, ezért az oszlopokat (és esetleg még más információkat is) itt helyezzük el, például az egyik oldaltartalmat gyakran használják alternatív navigációként vagy a fő tartalom szempontjából releváns információk megadására. Ekkor az egyes oszlopok szélességét eltérő nagyságúra állítjuk be, de fontos, hogy az összes szélesség nem haladhatja meg a 100%-ot.

A mintában látható elrendezéshez a két szélső oszlopnak a meglévő osztálymegadáson kívül külön osztályt is meghatározunk (pl. <div class=

„column side”) és ugyanígy teszünk a középső oszloppal is (pl. <div class=



A reszponzivitást továbbra is biztosíthatjuk a @media-hoz rendelt tulajdonságok beállításával:

```
@media screen and (max-width: 800px) {
  .column.side, .column.middle {
    width: 100%;
  }
}
```

Footer (lábléc)

A lábléc az oldal alján található, amely gyakran tartalmaz olyan információkat, mint a szerzői jog és a kapcsolattartási információ.

A lábléc formátumának beállításai CSS-ben:

- gyakran tartalmaz valamilyen háttérszínt, esetleg háttérképet,
- a magassági mérete általában kisebb, mint a fejlécé, mert jóval kevesebb információt szokás benne elhelyezni,
- a szöveges tartalma rendszerint kisebb betűmérettel formázott, mint a törzsszöveg,
- a belső tartalom és az esetleges szegély távolabb van elhelyezve (padding),
- a tartalma igazított (vízszintesen/függőlegesen középre).

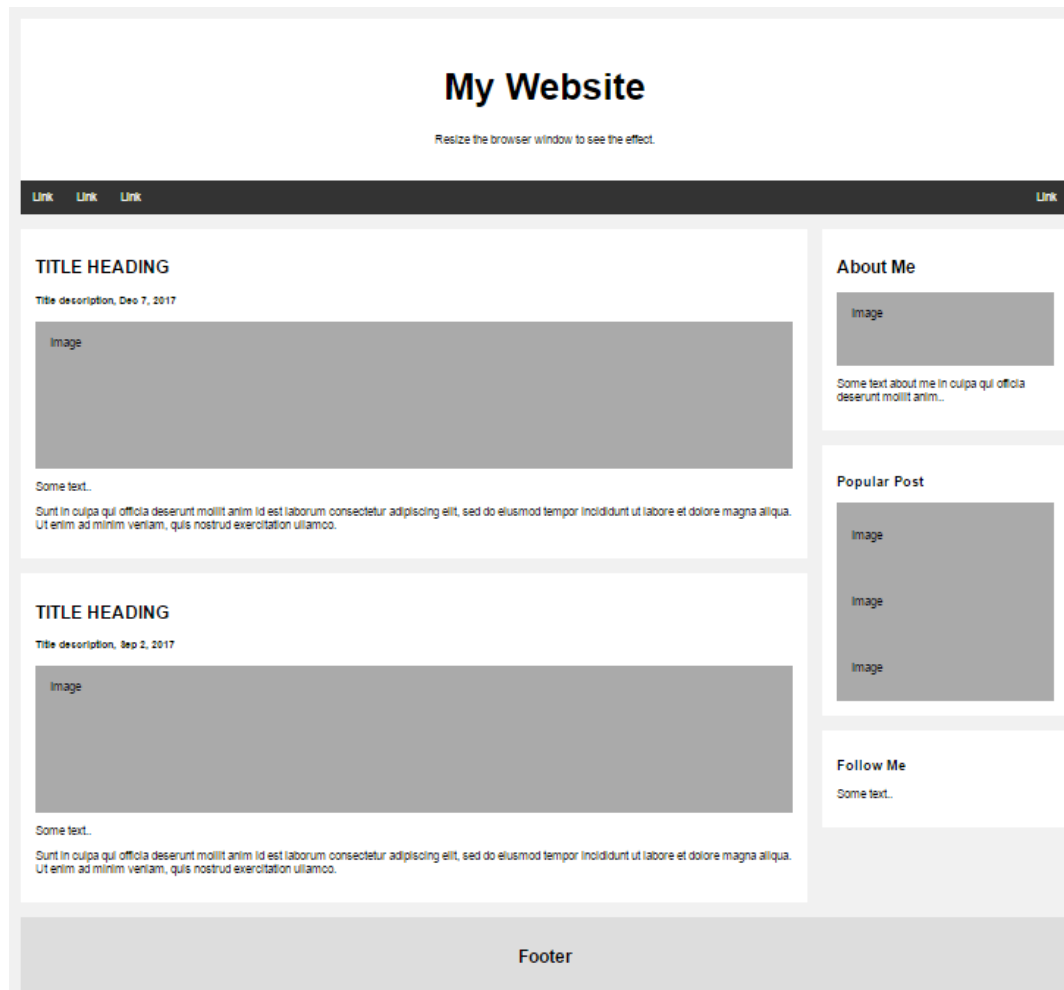
formázott megjelenés



a szerkezeti felépítés (HTML)

```
<div class="footer">
  <p>Footer</p>
</div>
```

Példa egy reszponzív (érzékeny) weboldal elrendezésére



4.A HTML5 tartalomfüggő címkéinek használata

Tárolócímkék

A HTML5 tulajdonságainak ismertetésénél már volt róla szó, hogy alkalmazhatók a weboldal részeinek meghatározására a HTML5 tartalomfüggő tárolóelemei a <div> blokkszintű elemek helyett. Ezek közös jellemzője, hogy páros címkéjűek.

A HTML5 tartalomfüggő tárolóelemei:

címke	hatása
<header>	a weboldal fejlécét tartalmazza
<nav>	(navigation) a navigációs elemek (menüpontok) tárolóeleme
<article>	független, önálló tartalmat jelöl (pl. cikk, hír, fórumhozzászólás, blogbejegyzés, megjegyzés)
<aside>	a fő tartalomhoz nem közvetlenül kapcsolódó információk megjelenítésére szolgáló oldalsáv
<section>	a weboldal tartalmi részének egy nagyobb szakasza, fejezete
<footer>	a weboldal láblécét tartalmazza

Példa a HTML5 tartalomfüggő tárolóelemeinek alkalmazására:

```
<body>
  <div id="tarolo">
    <header id="fejlec">
      Az oldal/cég/termék neve és alapadatai
    </header>
    <nav id="menu">
      Termékek | Vásárlás | Kapcsolatok
    </nav>
    <div id="balmenu">
      <b>Kínálat</b><br>-PC<br>-Laptop<br>-Egér
    </div>
    <div id="tartalom">
      Az oldal tartalma
    </div>
    <footer id="labc">
      Céginfó, jogi információk
    </footer>
  </div>
</body>
```

Speciális tárolócímkék ábrákhoz

Képek speciális megjelenítéséhez alkalmazzuk a **<figure>** páros címkéjű tároló elemet, amelybe egyszerre akár több ábra is beágyazható. Az ábrá(k)hoz kapcsolódó feliratot (címet, képaláírást, leírást, magyarázatot) a **<figcaption>** páros címkével határozzuk meg. A <figcaption> közvetlenül a <figure> címke után vagy a </figure> előtt adható meg. Tulajdonképpen hasonlóan használjuk, mint a nyomtatott anyagokban (pl. könyvek és folyóiratok esetén) a képek, diagramok, vagy példakódok képaláírásainál megszoktuk.

Ha a fő tartalomhoz tartozó fontos információt adunk közre, akkor a `<figure>`, ha kevésbé fontos vagy a tartalomhoz csupán érintőlegesen kapcsolódó információkat, akkor az `<aside>` elemet használjuk.

Egyetlen kép megjelenítése képaláírással és forrásmegjelöléssel

```
<figure>
  <blockquote>
    
  </blockquote>
  <figcaption>
    1. ábra Intel P5 processzora....<br>
    <cite>
      <a href="https://www.intel.com">Intel Corporation</a>
    </cite>
  </figcaption>
</figure>
```



Több kép megjelenítése képaláírással és forrásmegjelöléssel

```
<figure style="text-align:center;">
  
  
  

  <figcaption>
    A kiválasztott alaplaphoz még egy processzort, egy pro-
    cesszor hűtőventilátort és memória modulokat is kell vá-
    sárolni.
  </figcaption>
</figure>
```



Több kép megjelenítése képaláírással és forrásmegjelöléssel

```
<figure>
  <figcaption>
    Ha HTML forráskódot jelenítünk meg, akkor a <code> < és
    ></code> karakterek helyett entitásukat használjuk:
  </figcaption>

  <p>
    <code>
      &lt;a rel="license" href=https://creativecommons.org/licenses/by-nc-sa/2.5/hu/ &gt;
      Creative Commons licenc &lt;/a>&gt;
    </code>
  </p>
</figure>
```

