

WEBTERVEZÉS – GYAKORLATI JEGYZET

3. gyakorlat

HTML5 szemantikus tagek,
CSS kijelölők, formázás és pozicionálás

Készítették:

Cservenák Bence

Farkas Anikó

Savanya Sándor

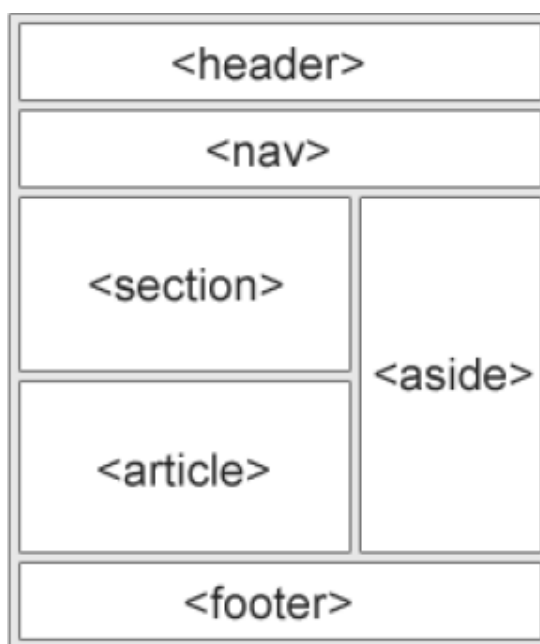
FIGYELEM!

A jegyzet folyamatosan készül, így előfordulhatnak benne apróbb hibák, hiányosságok, elírások. Ha valaki esetleg ilyenre talál, kérem, írjon a `Cservenak.Bence@stud.u-szeged.hu` címre, hogy mihamarabb javíthassuk.

1. A HTML5 szemantikus tagek

A HTML5 szabványban bevezettek új tageket a tartalom strukturálására. Ezek a tagek főként szemantikai jelentéssel bíró, egyszerű div elemeket helyettesítő blokk szintű (block) objektumok.

- `<header>...</header>`: fejléc
- `<footer>...</footer>`: lábléc
- `<nav>...</nav>`: navigációs menü
- `<aside>...</aside>`: oldalsáv
- `<main>...</main>`: az oldal lényegi részei
- `<section>...</section>`: logikai egység
- `<article>...</article>`: önálló tartalom



Példa: Egy lehetséges “layout” a fenti tagek használatával

2. CSS kijelölők (Folytatás)

A múlt órán megismertedtünk a legalapvetőbb CSS kijelölőkkel (elemkijelölő, osztálykijelölő, azonosítókijelölő, univerzális kijelölő). Nézzünk néhány példát egyéb kijelölőkre!

- Amikor több kijelölőre is ugyanaz a stílus vonatkozik, vesszővel elválasztva felsoroljuk a kijelölőket (group selector)

```
h1, h2, .warning { color: red; }
```

- `Element1 Element2`: az `Element1`-en belüli összes `Element2`-t kijelöli
- `Element1>Element2`: minden olyan `Element2`-t kijelöl, amelynek közvetlen szülője `Element1`

```
<body>
  <p>első</p>
  <div>
    <p>második</p>
  </div>
  <p>harmadik</p>
</body>
```

```
body p {
  /* első, második, harmadik */
}

body>p {
  /* első, harmadik */
}
```

- `Element[a]`: minden olyan `Element` elemet kijelöl, ami rendelkezik az `a` attribútummal
- `Element[a="val"]`: minden olyan `Element` elemet kijelöl, amelynek a `attribútumnak` az értéke `val`

```
<p id>első</p>
<p id="foo">második</p>
<p id="bar">harmadik</p>
<p>negyedik</p>
```

```
p[id] { /* első, második, harmadik */ }

p[id="foo"] { /* második */ }
```

- Pseudo-class segítségével történő kijelölés:

- a hivatkozások 4 állapota

```
a:link      { /* ha még nem kerestük fel... */ }
a:visited   { /* ha már felkerestük... */ }
a:hover     { /* ha a kurzor fölötte áll... */ }
a:active    { /* ha rákattintunk a hivatkozásra... */ }
```

- néhány egyéb pseudo-class kijelölő

- `:first-child`: a beágyazó objektumnak legelső gyermeke
- `:nth-child(n)`: a beágyazó objektumnak `n`-edik gyermeke (`n` pozitív egész)
- `:last-child`: a beágyazó objektumnak utolsó gyermeke
- `:first-of-type`: az azonos típusú, vele egy szinten lévő elemek közül a legelső
- `:nth-of-type(n)`: az azonos típusú, vele egy szinten lévő elemek közül a `n`-edik (`n` pozitív egész)
- `:last-of-type`: az azonos típusú, vele egy szinten lévő elemek közül az utolsó

```
/* a legelső listaelem */
li:first-child { ... }

/* a harmadik listaelem */
li:nth-child(3) { ... }
```

Az összes kijelölő: https://www.w3schools.com/cssref/css_selectors.asp

3. Hivatkozások, listák és táblázatok formázása

A. Hivatkozások formázása

- A hivatkozásokra is használhatók a múlt órán tanult szöveg- és háttérformázások
- A hivatkozásokat állapot alapján is formázhatjuk (lásd: *előző pont*, *pseudo-class*)
- A `text-decoration` tulajdonsággal megadhatjuk a szövegdekorációt
 - `underline`: a hivatkozás alatt egy vonal jelenik meg (alapértelmezett)
 - `none`: általában a hivatkozás alatti vonal eltüntetésére használjuk

```
<a href="https://www.google.com/">Első link</a> <br/>
<a href="https://www.google.com/">Második link</a>

<!-- CSS formázás -->
a:nth-of-type(2) { text-decoration: none; }
```

[Első link](https://www.google.com/)
[Második link](https://www.google.com/)

B. Listák formázása

- A listák felsorolásjele a `list-style-type` tulajdonsággal módosítható
 - rendezetlen lista (`ul`) esetén: `disc` (teli karika), `circle` (üres karika), `square` (teli négyzet), `none` (nincs felsorolásjel)
 - rendezett lista (`ol`) esetén: `decimal` (arab szám), `lower-roman`/`upper-roman` (római szám), `lower-alpha`/`upper-alpha` (kisbetű/nagybetű), `none` (nincs felsorolásjel)
- A `list-style-image` tulajdonsággal kép is megadható listaelemként
 - a kép elérési útvonalát az `url` ('*elérési útvonal*') segítségével adjuk meg
- A `list-style-position` tulajdonsággal beállítható a felsorolásjelek pozíciója
 - `outside`: a felsorolás függőleges vonalába a felsorolt tartalom került
 - `inside`: a felsorolás függőleges vonalába a listajelek kerülnek

C. Táblázatok formázása

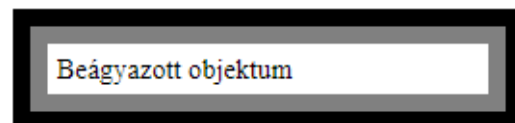
- A táblázatra, annak soraira, celláira is használhatók a tanult szöveg- és háttérformázások, valamint a térközre, margóra és szegélyre vonatkozó utasítások
- A `td:hover` kijelölő kijelöli azt a táblázatcellát, ami fölött a kurzor áll
- A `vertical-align` utasítással megadhatjuk a cella tartalmának függőleges igazítását (`middle`: középre igazítás)
- A `caption-side` tulajdonsággal megadhatjuk a táblázat címének (`caption`) pozícióját
 - lehetséges értékei: `top` (a táblázat fölött, alap), `bottom` (a táblázat alatt)
- A `border-collapse: collapse;` utasítással összevonhatjuk a táblázat szegélyeit
- A `table { margin: auto; }` utasítással a táblázat vízszintesen középre igazítható

D. A **box-sizing** tulajdonság

- Segítségével megadhatjuk, hogy az elem méretébe (szélességébe és magasságába) beleszámítsanak-e a belső térközök (padding) és a szegélyek (border)
- Gyakori értékei:
 - `content-box`: a méretbe csak a tartalom tartozik bele (alapértelmezett)
 - `border-box`: a méretbe a tartalom, a padding és a border is beletartozik
- Ha a méreteket %-ban adjuk meg, akkor kiemelten fontos lehet, hogy a padding és border is beletartozzon a méretbe

```
<div id="parent">
  <div id="child">Beágyazott objektum</div>
</div>

<!-- CSS formázás -->
#parent { border: 10px solid black; }
#child { box-sizing: border-box; width: 100%;
         border: 10px solid gray; padding: 5px; }
```



Megjegyzés: A példában `box-sizing: content-box`; esetén a beágyazott objektum "kilógna"

4. A CSS rangsor

Kérdés: Több, egymásnak ellentmondó CSS formázás közül melyik lesz érvényes az objektumra?

Válasz: Egy előre definiált rangsor alapján dől el

1. **!important** CSS utasítások (**legmagasabb priorítás**)
 2. inline CSS (a tagek `style` attribútumaként adjuk meg)
 3. azonosítókijelölő (`id` alapján)
 4. osztálykijelölő (`class` alapján)
 5. elemkijelölő (**legalacsonyabb priorítás**)
- A felhasználó által definiált **!important** stílusok erősebbek, mint a programozó által definiáltak
 - A felhasználó által definiált egyéb stílusok viszont önmagukban **nem** erősebbek, mint a programozó által definiáltak
 - Az összetett kijelölő esetén a részkijelölők prioritásai összeadódnak

Példa:

Legyenek adottak a következő prioritások: $p(\text{id selector}) = 100$, $p(\text{class selector}) = 10$, $p(\text{element selector}) = 1$. Ekkor a `#main .container p` összetett kijelölő prioritása: $100 + 10 + 1 = 111$.

- Ha az eddigiek alapján nem dőlt el a rangsor, akkor a kódban későbbi formázás lesz érvényes

5. Pozicionálás, helyzetmegadás

A. A position CSS tulajdonság

- A HTML objektumok pozícióját a `position` CSS tulajdonsággal adhatjuk meg, értékei:
 - `static` (alap): a normál szövegfolyamban lévő hely, a pozicionálásnak nincs hatása
 - `relative`: a normál szövegfolyambeli helyéhez képest elmozdítja a kért mértékben
 - `absolute`: a legközelebbi, **nem** static helyzetű beágyazó objektum belső széléhez (annak hiányában a viewport széléhez) képest pozicionál
 - `fixed`: az elem gördítéskor is rögzített helyen marad

A **sticky** kulcsszó: https://www.w3schools.com/howto/howto_css_sticky_element.asp

- Nem** static módon pozicionált HTML objektumok esetén megadhatók:
 - `top`, `bottom`: függőleges igazítás
 - `left`, `right`: vízszintes igazítás

```
<div class="container">
  <b>Pozicionálás</b>
  <div class="first box">Első</div>
  <div class="second box">Második</div>
  <div class="third box">Harmadik</div>
</div>

<!-- CSS utasítások -->
.box { padding: 20px 10px; }
.container { background-color: lightblue; position: relative; height: 800px; }
.first { background-color: tomato; position: absolute; top: 0; right: 0; }
.second { background-color: mediumaquamarine; /* top: 500px; */ }
.third { background-color: lightgreen; position: fixed; }
```

Kódpélda: A pozicionálással kapcsolatos CSS utasítások működése

A fenti példa magyarázata:

- Az `absolute` módon pozicionált "Első" feliratú doboz az őt beágyazó nem static (hanem jelen esetben `relative`) helyzetű doboz (`container`) jobb felső sarkához igazodik.
- A "Második" feliratú doboz `static`, így a pozicionálásnak nincs hatása.
- A `fixed` helyzetű "Harmadik" feliratú doboz gördítéskor is rögzített helyen marad.

B. Úsztatás

- A `float` tulajdonság úsztatja az elemet, amelyet a másik irányból a tartalom körbefolyhat
 - értékei: `left` (balra úsztatás), `right` (jobbra úsztatás), `none` (nincs úsztatás)
- A `clear` tulajdonság megakadályozza másik, úsztatott elem elhelyezését
 - értékei: `left` (balról nem lehet úsztatott elem), `right` (jobbról nem lehet úsztatott elem), `both` (sem balról, sem jobbról nem lehet úsztatott elem), `none` (balról és jobbról is lehet úsztatott elem)

```

<p>A zsiráf (Giraffa camelopardalis) Afrikában élő párosujjú patás emlős, a legmagasabb és leghosszabb nyakú szárazföldi élőlény. A szavannák lakója az ókortól kedvelt attrakció volt – Rómában először Julius Caesar mutatott be zsiráfokat az amfiteátrumi játékokon – és ma is népszerű szafarikon és állatkertekben. Neve arab eredetű, olasz közvetítéssel jutott el a magyarba a késő középkorban. A tudományos nevében szereplő camelopardalis a faj addig használt nevére, a görög kamélopardaliszra („tevepárduc”) utal.</p>

<!-- CSS utasítások -->
img {
  height: 140px;
  float: left;
  margin-right: 10px;
}
```



A zsiráf (*Giraffa camelopardalis*) Afrikában élő párosujjú patás emlős, a legmagasabb és leghosszabb nyakú szárazföldi élőlény. A szavannák lakója az ókortól kedvelt attrakció volt – Rómában először Julius Caesar mutatott be zsiráfokat az amfiteátrumi játékokon – és ma is népszerű szafarikon és állatkertekben. Neve arab eredetű, olasz közvetítéssel jutott el a magyarba a késő középkorban. A tudományos nevében szereplő camelopardalis a faj addig használt nevére, a görög kamélopardaliszra („tevepárduc”) utal.