

# Webfejlesztés

# HTML5 + CSS3

2019-2022

# Bevezető

- A Weben lévő tartalmakat webböngésző alkalmazások (segítségével érhetjük el. (pl.: Mozilla Firefox, Internet Explorer, Google Chrome, Opera, stb.)
- Annak érdekében, hogy az adott weboldalt minél több böngészőprogramban is hibamentesen használhassuk, a keretrendszer fejlesztésénél be kell tartanunk a megfelelő ajánlásokban, szabványokban foglaltakat.
- A Web-alapú jelölőnyelvre vonatkozó ajánlásokat a W3C konzorcium dolgozta ki. (Tim Berners –Lee) - A W3C által közzétett ajánlásokra (de facto) szabvánnyként is lehet tekinteni.

# Hiperszöveg jelölőnyelvek

- HTML - (HyperText Markup Language – Hiperszöveg jelölőnyelv)  
- az SGML (Standard Generalized Markup Language – Szabványos általános jelölőnyelv) szabványon alapul ( 1986 – 1999, v4.01)
- XHTML - az XML szabvány (eXtensible Markup Language – Bővíthető jelölőnyelv) alapuló szabvány (2000, 2001)
- **HTML5** - előnye, hogy egyszerűbben készíthetünk multimédiás lehetőségekben gazdag, interaktív alkalmazásokat úgy, hogy kevésbé kell tartanunk a böngészőprogramokban egyénileg telepítendő beépülő modulok (pl. Flash, Silverlight meglététől, hiszen a szabványban megjelenő új lehetőségeket a web-böngésző programok alapszolgáltatásként (natív módon) támogatják.
- **CSS3** - (Cascading Style Sheets) - A CSS használatával *elkülöníthetjük a weboldal megjelenését a tartalomtól* úgy, hogy a megjelenésre vonatkozó definíciók akár külső állományban is elhelyezkedhetnek. A CSS szabványnak különböző verzió jelentek meg az évek folyamán általánosan használt verziója a CSS3

# Szerkesztő programok - Böngészők

- Editorok:

- Kódszintű karakteres programok (pl.: Notepad++, Netbeans, VS Code, )
- „Alakhű” WYSIWYG szerkesztők (pl.: Komposer, FirstPage, Dreamveawer, ...)

- Böngészők:

- Mozilla Firefox
- Internet Explorer
- Google Chrome
- Opera, ...

# HTML5 alapok

- Alaptstruktúra

```
<!DOCTYPE html>
<html>
  <head>
    <title> </title>
  </head>
  <body>
  </body>
</html>
```



index.html

- A HTML állomány .html kiterjesztéssel rendelkezik
- A HTML állomány egyszerű szövegállomány, amely rövid jelölő tag –eket tartalmaz
- Tagnek a < és > jelek között található utasítást nevezzük.
- Egy tag hatása odáig tart, míg le nem írjuk a tag záró pájrát, ami ugyanaz, mint a nyitó tag, csak egy per jel (/) vezeti be. Pl.: <html> </html>
- Vannak olyan tagok is, amelyeknek nincs záró páruk („önlezáró”). Pl.: ilyen az egy kép beszúrására alkalmas <img> tag is.
- A tagokat egymásba is ágyazhatjuk, de ekkor fontos, hogy mindenkor azt a tagot zárjuk le először, amelyet utoljára nyitottunk meg.

# HTML5 alapok

## • Alaptstruktúra

```
<!DOCTYPE html>
<html>
  <head>
    <title> </title>
  </head>
  <body>
    </body>
  </html>
```

- A HTML-kódot az alkalmazandó dokumentumtípus (**DOCTYPE**) megadásával kezdjük Ezzel informáljuk a böngészőprogramot, hogy milyen szabvány szerint készítettük a dokumentumot, hogy az megfelelő módon tudja megjeleníteni. Ezt elhagyva a böngészőprogram nem szabvánnykövető üzemmódba kerül
- (HTML5 előtt: <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN,, "http://www.w3.org/TR/html4/strict.dtd">
- **<HTML>** : Azt jelzi, hogy a közte lévő szöveget a HTML szabvány szerint kell értelmeznie a böngészőnek. Ezt a tag-et minden kötelező lezárni.
- **<HEAD>** : Ez a HTML oldal fejrésze. Sok hasznos információt elhelyezhetünk benne, de egyelőre még csak a következő tag-et tegyük bele:
- **<TITLE>** : A HTML oldal címe, vagy ha úgy tetszik fejléce, ami a böngésző fejlécében jelenik meg. Paramétereit nincsenek, és minden kötelező lezárni!
- **<BODY>** : A HTML oldal törzse. A nyitó és záró tag közé beírt szöveget a böngésző megpróbálja értelmezni, majd megjeleníteni. Mindig le kell zárni.

# HTML5 alapok

- A nyelv és karakterkódolás megadása:

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

- Az oldal nyelvét a `<html>` tag `lang` paraméterében kell megadni.
  - képernyőolvasó programok ez alapján el tudják dönten, hogy milyen nyelven olvassák fel a szöveget
  - Bizonyos nyelv megadások kihatnak arra, hogy a böngészőprogram milyen betűtípuszt választ a szöveg megjelenítésére.
  - A böngészőbe épített nyelvhelyesség-ellenőrző eszközök ez alapján tudják eldönten, hogy az oldal milyen nyelvű.
  - A keresőprogramok a nyelv alapján is indexelik az oldalakat, ezért ha adott nyelvű oldalak között keres valaki, akkor nagyobb eséllyel találja meg az oldalunkat.
- Ahhoz hogy pl.: a magyar nyelvben használatos speciális karakterek, ékezetes betűk helyesen jelenjenek meg a különböző operációs rendszerekben található különböző alkalmazásokban, megfelelő karakterkódolást kell beállítani.
  - A beállítás úgy történik, hogy a dokumentum head részében el kell helyeznünk `charset` meta tagot.

# HTML5 alapok

- **A kód tördelése, megjegyzések a kódban**

- a HTML-kód tördelését nem befolyásolja mennyi üres helyet teszünk és hol nyomunk entert
- Ez alól kivételt jelent, ha például a `<pre>` taget használjuk, vagy a stíluslapban megadjuk, hogy előre formázott szövegként akarunk valamit megjeleníteni.
- Megjegyzéseket a `<!--` és `-->` karakterláncok közé írhatunk. Az így elhelyezett szöveg nem jelenik meg a képernyőn, csak a forráskódban lesz olvasható.

```
<!DOCTYPE html>
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
  </head>
  <body>
    <p>
      Attól, hogy rengeteg üres helyet használlok
      és össze-vissza tördelem
      a sorokat a kódban, azzal az oldal megjelenését nem befolyásolom. </p>
      <!-- a pre elem a preformatted (előreformázott) szöveg rövidítése -->
      <pre>
        Kivéve ha a <pre> taget
          használom, mert ekkor bizony számít,
          hogy mi szerepel új sorban, és mennyi
          üres helyet hagyok ki. </pre>
    </body>
</html>
```

Attól, hogy rengeteg üres helyet használlok és össze-vissza tördelem a sorokat a kódban, azzal az oldal megjelenését nem befolyásolom.

Kivéve ha a taget

használom, mert ekkor bizony számít,  
hogy mi szerepel új sorban, és mennyi  
üres helyet hagyok ki.

# HTML5 alapok

- **Speciális karakterek:**

- A szövegben nem jelennek meg, megfelelően, speciális jelentésük miatt

```
<!DOCTYPE html>
<html lang="hu">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
      <title>HTML alapok - Speciális karakterek</title>
  </head>
  <body>
    <p>rossz: A <p> tag-et akkor használhatjuk, ha bekezdést akarunk létrehozni.</p>
    <p>jó: A &lt;p&gt; taget akkor használhatjuk, ha bekezdést akarunk létrehozni.</p>
  </body>
</html>
```

rossz: A

tag-et akkor használhatjuk, ha bekezdést akarunk létrehozni.

jó: A <p> taget akkor használhatjuk, ha bekezdést akarunk létrehozni.

Gyakran használt speciális karakterek

Karakter	HTML kód	unicode	Karakter	HTML kód	unicode
'	&acute;	&#180;	<	&lt;	&#60;
&	&amp;	&#38;	?	&macr;	&#175;
	&brvbar;	&#166;	?	&micro;	&#181;
.	&bull;	&#8226;		&nbsp;	&#160;
c	&cent;	&#162;	?	&not;	&#172;
©	&copy;	&#169;	?	&para;	&#182;
º	&deg;	&#176;	?	&permil;	&#8240;
?	&euro;	&#8364;	?	&plusmn;	&#177;
÷	&divide;	&#247;	"	&quot;	&#34;
?	&frac12;	&#189;	®	&reg;	&#174;
?	&frac14;	&#188;	>	&raquo;	&#187;
?	&frac34;	&#190;	§	&sect;	&#167;
>	&gt;	&#62;	"	&uml;	&#168;
!	&iexcl;	&#161;	×	&times;	&#215;
?	&iquest;	&#191;	™	&trade;	&#8482;
<	&laquo;	&#171;	o	&oslash;	&#248;

# HTML5 alapok

- A HTML5 szövegszintű elemei

- **<i>** – dőlt (kézírásra emlékeztető) betű
- **<em>** – hangsúlyos kiemelés
- **<b>** – félkövér betű
- **<strong>** – erős kiemelés
- **<u>** – aláhúzás
- **<s>** – nem pontos, idejét múlt információ
- **<mark>** – szövegkiemelés

```
<!DOCTYPE html>
<html lang="hu">
<head>
    <title>Szövegszintű elemek tesztelése képernyőolvasó programban</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<body>
    Ez egy teszt szöveg</p>
    <p><i>Ez egy teszt szöveg</i></p>
    <p><em>Ez egy teszt szöveg</em></p>
    <p><strong>Ez egy teszt szöveg</strong></p>
    <p><b>Ez egy teszt szöveg</b></p>
    <p><u>Ez egy teszt szöveg</u></p>

    <p><mark>Neumann</mark> János 1903. december 28-án született
    <mark>Neumann</mark> Miksa és Kann Margit első gyermeként
    Budapesten, a Váczi körút (ma Bajcsy-Zsilinszky út) 62. sz. házban. </p>
</body>
</html>
```

Ez egy teszt szöveg

*Ez egy teszt szöveg*

**Ez egy teszt szöveg**

Ez egy teszt szöveg

Ez egy teszt szöveg

Neumann János 1903. december 28-án született Neumann Miksa és Kann Margit első gyermeként Budapesten, a Váczi körút (ma Bajcsy-Zsilinszky út) 62. sz. házban.

# HTML5 alapok

- A HTML5 szövegszintű elemei – 2

- <a> – hiperhivatkozás
- A hiperlinkekkel (azaz hivatkozásokkal, vagy linkekkel) kapcsolatot hozhatunk létre a különböző oldalak, médiaelemek stb. között. Az egyes oldalakon belül is létrehozhatunk hivatkozásokat, az oldalon belül elhelyezett könyvjelzőkre.
- A böngészők a linkeket alapesetben aláhúzással illetve eltérő színnel jelölik. A még nem látogatott linkek színe alapesetben kék, a már látogatottaké lila, az aktív linkek színe (vagyis amire rákattintottunk és töltődik) piros.
- A linkek megjelenése a stíluslapok segítségével megváltoztatható.
- Paraméterek:

- **href:** értékként a hivatkozott oldal url-jét adhatjuk meg. Pl.:

```
<a href="http://www.elte.hu/">ELTE honlapja</a>
```

hivatkozásokat nem csak abszolút módon, hanem az aktuális oldal elérhetőségéhez képest relatívan is megadhatjuk

pl.: `<a href="../kepek/sziklamaszas.jpg"> megtekintheted nagyobb méretben </a>`  
`<a href="/kepek/sziklamaszas.jpg"> megtekintheted nagyobb méretben </a>`

- **target:** ezzel befolyásoljuk, hogy a link hova töltődjön be.

\_self: A jelenlegi böngészési környezetbe töltődik be az oldal

\_blank: Egy új, névtelen böngészési környezetbe töltődik be az oldal

\_parent: A szülő böngészési környezetébe töltődik be az oldal

\_top: a legfelső szintű böngészési környezetbe töltődik be az oldal.

- **rel:** a linket tartalmazó dokumentum és a linkelt oldal közti kapcsolat megadására szolgál.
- ...

# <a példa>

```
<!DOCTYPE html>
<html lang="hu">
<head>
<title>Link létrehozása</title>
<meta charset="utf-8">
</head>
<body>
<nav>
    <ul id="menu">
        <li><a href="#magamrol">Magamról</a></li>
        <li><a href="#elerhetoseg" target="_blank">Elérhetőség </a></li>
    </ul>
</nav>
<h1 id="magamrol">Magamról</h1>
<p><a href="http://www.budapest.hu" target="_blank">Budapesten</a> születtem, azóta is itt élek.</p>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit...</p>
<p><a href="#menu">Vissza a menűhöz</a></p>
<h1><a id="elerhetoseg"></a>Elérhetőség</h1>
<p>A <a href="mailto:gipszjakab@gipsz.jakab.hu">gipszjakab@gipsz.jakab.hu</a> címen írhatsz nekem emailt.</p>
<p>Aliquam in felis vel ipsum vehicula placerat ut vel lectus...</p>
<p><a href="#menu">Vissza a menűhöz</a></p>
</body>
</html>
```

- [Magamról](#)
- [Elérhetőség](#)

## Magamról

[Budapesten](#) születtem, azóta is itt élek.

Lorem ipsum dolor sit amet, consectetur adipiscing elit...

[Vissza a menűhöz](#)

## Elérhetőség

A [gipszjakab@gipsz.jakab.hu](mailto:gipszjakab@gipsz.jakab.hu) címen írhatsz nekem emailt.

Aliquam in felis vel ipsum vehicula placerat ut vel lectus...

[Vissza a menűhöz](#)

# HTML5 alapok

- **A HTML5 szövegszintű elemei – 3**

- <**cite**> – Egy mű (könyv, film esszé, dal, ...) címe – ha hivatkozunk rá a szövegben
- <**q**> – idézet, ami egy más forrásból származó idézet jelölésére szolgál
- <**dfn**> – a szövegben az adott helyen található definiált fogalmat jelöli
- <**abbr**> – rövidítések jelölésére, és azok feloldásának megadására
- <**code**> – számítógépes programkód (vagy kódrészlet) elhelyezésére alkalmazható.
- <**var**> – főleg matematikai kifejezésekben előforduló változók jelölésére
- <**samp**> – egy számítógépes program kimenetének, üzenetének jelölésére.
- <**kbd**> – felhasználó által adott bemenet jelölésére pl.: billentyűzet konfiguráció
- <**sub**> és <**sup**> elemek – alsó- és felső index jelölésére
- <**br**> – sortörés = manuális sortörést jelent. A tagnek nincs záró párja. Ne használd térközként és bekezdések jelölésére!!
- <**wbr**> – feltételes sortörés – hosszabb sorok esetén ha nem férnek el az aktuális ablak méretnél
- <**span**> – karakterek formázására

# példák

```
<!DOCTYPE html>
...5 lines
<html>
    <head ...5 lines />
    <body>
        <p>A <cite>Szabadbölcsészet - Tipográfia</cite> oldal szerint a „folyóiratcikk esetén
            a folyóirat címe kurzív (vagy a cikk címe idézőjelek közé kerülhet), könyvnél a könyv
            címe kurzív”.</p>
        <p>A szerző ebben a témaban írt publikációja megjelent az
            <abbr title="Fertődi Növénynemesítési és Növénytermesztési
                Kutató Intézet Közleményei">FNNKIK</abbr>-ban.</p>
        <pre><code class="language-pascal">
var i: Integer;
begin
    i := 1;
    for i:= 1 to 100 do write(i);
end.
</code>
</pre>
        <p>A lineáris függvény képének egyenlete:
            <math>y = ax + b</math>
        <p>A teljes szöveg kijelöléséhez a <kbd>CTRL+A</kbd> billentyűkombinációt kell lenyomni.</p>
        <p>Pitagorasz tétele: <math>a^2+b^2=c^2</math>
        </p>
    </body>
</html>
```

A Szabadbölcsészet – Tipográfia oldal szerint a „folyóiratcikk esetén a folyóirat címe kurzív (vagy a cikk címe idézőjelek közé kerülhet), könyvnél a könyv címe kurzív”.

A szerző ebben a témaban írt publikációja megjelent az FNNKIK-ban.

```
var i: Integer;
begin
    i := 1;
    for i:= 1 to 100 do write(i);
end.
```

Fertődi Növénynemesítési és Növénytermesztési  
Kutató Intézet Közleményei

A lineáris függvény képének egyenlete:  $y = ax + b$

A teljes szöveg kijelöléséhez a CTRL+A billentyűkombinációt kell lenyomni.

Pitagorasz tétele:  $a^2+b^2=c^2$

# HTML5 alapok

- A HTML5 szövegszintű elemei – 4

- <span> – karakterek formázására
  - Önmagában a <span> tagnek nincs szemantikai jelentése. Arra szolgál, hogy néhány karaktert, szót speciális formázással, vagy például nyelv megadással láthatunk el.
  - A <span> egy soron belüli (in-line) elem, amely előtt és mögött a böngésző nem hagy ki margót. Ezt a tulajdonságot a stíluslapok segítségével felülbírálhatjuk, és a span elemet akár blokkszintű elemmé is alakíthatjuk
  - Leggyakrabban a *class*, *style*, *lang*, *dir* paraméterekkel használjuk a <span> tag-et. Ezeket később, a stíluslapok tárgyalásánál részletesebben bemutatjuk.

```
<!DOCTYPE html>
...5 lines
<html>
  <head ...5 lines />
  <body>
    <p>Kedvenc színem a <span style="color:red">piros</span>,
       így nem volt kérdéses, hogy milyen színű autót vásárolok.</p>
    <span style="color: greenyellow"> ----- neon zöld -----
    <span style="color: pink"> -----rózsaszín----- </span>
  </body>
</html>
```

Kedvenc színem a **piros**, így nem volt kérdéses, hogy milyen színű autót vásárolok.

----- neon zöld -----

-----rózsaszín-----



Kedvenc színem a **piros**, így nem volt kérdéses, hogy milyen színű autót vásárolok.

----- neon zöld -----

-----rózsaszín-----



# CSS3 bevezetés

- a HTML5 szövegszintű elemeit stíluslapok segítségével is formázhatjuk
- A oldalak tartalmától (HTML-kód) és a kinézetet stíluslapokat használva el lehet kell különíteni
- Pl. Bekezdések formázása – Betűtípus, stílus, szín, igazítás, ....
- **I. Inline (beágyazott) megadás**
  - a bekezdés stílusát úgy adjuk meg, hogy a style paramétert használjuk, amelyben megadjuk a szöveg igazításáért felelős *text-align* tulajdonságot, mégpedig a *center* [right, left, justify] értékkel. A tulajdonság és annak értéke között *kettőspontot* helyeztünk el.
  - Továbbá a betűtípus beállításához a *font-family* tulajdonságot  
Itt a az általános betűcsaládoból választhatunk:
    - serif: talpas betűtípus (pl. Times)
    - sans-serif: talpatlan betűtípus (pl. Arial, Nimbus sans)
    - cursive: kurzív betűtípus (pl. Zapf-Chancery)
    - fantasy: a hagyományos betűtípusoktól eltérő, fantáziadúsabb betűcsaládok tartoznak ide
    - monospace: olyan betűcsalád, amelyben minden karakter azonos szélességű.

```
<p style="text-align:center; font-family:monospace;">Lorem ipsum dolor sit amet, ...</p>
```

# CSS3 bevezetés

```
<!DOCTYPE html>
...5 lines
<html>
    <head ...5 lines />
    <body>
        <p style="text-align: center; font-family: serif; background-color: lightblue ">Egy h
            hogy a szöveg különböző funkciójú részei vizuálisan is megkülönböztethetők legye
            hogy a HTML5 számos szövegrésztipushoz rendelkezésünkre bocsát jelölőket (idézet,
            minden esetben elegendők). Nincs például olyan HTML5-jelölő, amely segítségével meg
            A HTML5-ben ezeket a - normál szövegtől eltérő funkciójú - szövegréseket is csak
            így viszont nincs mód arra, hogy ezek az elemek másként legyenek formázva, mint a
        </body>
    </html>
```

Egy hosszabb szövegen szükség lehet arra, hogy a szöveg különböző funkciójú részei vizuálisan is megkülönböztethetők legyenek. Az előző fejezetben láthattuk, hogy a HTML5 számos szövegrésztipushoz rendelkezésünkre bocsát jelölőket (idézet, definíció stb.), de ezek nem minden esetben elegendők. Nincs például olyan HTML5-jelölő, amely segítségével megjegyzést helyezhetünk el a szövegen, stb. A HTML5-ben ezeket a – normál szövegtől eltérő funkciójú – szövegréseket is csak <p> elemként jelölhetjük meg, így viszont nincs mód arra, hogy ezek az elemek másként legyenek formázva, mint a főszöveg.</p>

# CSS3 bevezetés

- **II. Lapon belüli definíció, típusszelktor használattal**

- A CSS szabvány lehetővé teszi, hogy az egyes tulajdonságokat ne közvetlenül az adott elemnél adjuk meg, hanem a lap <head> részében, illetve akár külső stílusállományban is
- Ekkor meg kell adnunk, hogy egy adott szabályt milyen elemre vonatkozóan szeretnénk érvényesíteni. → ez lesz a **szelktor**

```
<!DOCTYPE html>
...5 lines
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <style type="text/css">
            p{
                text-align: justify;
                font-family: monospace;
                background-color: lightgreen;
            }
        </style>
    </head>
    <body>
        <p>Egy hosszabb szövegben szükség lehet arra, hogy a szöveg különböző funkciójú részei vizuálisan is megkülönböztethetők legyenek. Az előző fejezetben láthattuk, hogy a HTML5 számos szövegrésztipushoz rendelkezésünkre bocsát jelölőket (idézet, definíció stb.), de ezek nem minden esetben elegendők. Nincs például olyan HTML5-jelölő, amely segítségével megjegyzést helyezhetünk el a szövegben, stb. A HTML5-ben ezeket a - normál szövegtől eltérő funkciójú - szövegrésekkel is csak <p> elemként jelölhetjük meg, így viszont nincs mód arra, hogy ezek az elemek másként legyenek formázva, mint a főszöveg.</p>
    </body>
</html>
```

Egy hosszabb szövegben szükség lehet arra, hogy a szöveg különböző funkciójú részei vizuálisan is megkülönböztethetők legyenek. Az előző fejezetben láthattuk, hogy a HTML5 számos szövegrésztipushoz rendelkezésünkre bocsát jelölőket (idézet, definíció stb.), de ezek nem minden esetben elegendők. Nincs például olyan HTML5-jelölő, amely segítségével megjegyzést helyezhetünk el a szövegben, stb. A HTML5-ben ezeket a - normál szövegtől eltérő funkciójú - szövegrésekkel is csak <p> elemként jelölhetjük meg, így viszont nincs mód arra, hogy ezek az elemek másként legyenek formázva, mint a főszöveg.

# CSS3 bevezetés

- Osztályok használata

- Az osztály használata azt jelenti, hogy az egyes <p> tagokat elláttuk olyan osztálynevekkel, amelyek utalnak a tulajdonságaikra → ehhez a class paramétert használjuk, és utána megadjuk az adott osztályneveket (pl.: balra, kozepre, jobbra, fixszeles).
- Ekkor meg kell adnunk, hogy egy adott szabályt milyen elemre vonatkozóan szeretnénk érvényesíteni.
- egy tagot akár több osztályba is besorolhatunk → ehhez az osztályok neve között szóközt kell kihagynunk

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>Stíluslap kitekintő - típus szelektor</title>
    <style type="text/css">
      p.balra {text-align:left;}
      p.jobbra {text-align:right;}
      p.kozepre {text-align:center;}
      p.fixszeles {font-family: monospace;}
    </style>
  </head>
  <body>
    <p class="balra">Egy hosszabb szövegben szükség lehet arra, hogy a szöveg különböző funkciójú részei vizuálisan is megkülönböztethetők legyenek. Az előző fejezetben láthattuk, hogy a HTML5 számos szövegrésztipushoz rendelkezésünkre bocsát jelölőket (idézet, definíció stb.), de ezek nem minden esetben elegendők. Nincs például olyan HTML5-jelölő, amely segítségével megjegyzést helyezhetünk el a szövegben, stb. A HTML5-ben ezeket a - normál szövegtől eltérő funkciójú - szövegrésekkel csak <p> elemként jelölhetjük meg, így viszont nincs mód arra, hogy ezek az elemek másként legyenek formázva, mint a főszöveg.</p>
  </body>
</html>
```

Egy hosszabb szövegben szükség lehet arra, hogy a szöveg különböző funkciójú részei vizuálisan is megkülönböztethetők legyenek. Az előző fejezetben láthattuk, hogy a HTML5 számos szövegrésztipushoz rendelkezésünkre bocsát jelölőket (idézet, definíció stb.), de ezek nem minden esetben elegendők. Nincs például olyan HTML5-jelölő, amely segítségével megjegyzést helyezhetünk el a szövegben, stb. A HTML5-ben ezeket a - normál szövegtől eltérő funkciójú - szövegrésekkel csak <p> elemként jelölhetjük meg, így viszont nincs mód arra, hogy ezek az elemek másként legyenek formázva, mint a főszöveg.</p>

# CSS3 bevezetés

- **III. külső stíluslapok használata**
  - A stíluslapot egy szöveges fájlban kell megírni, amit .css kiterjesztéssel kell elmenteni.
  - A stíluslapfájl legelső sorába ajánlott a karakterkódolásra vonatkozó információt írni.
  - Ezután az egész stíluslap nem más, mint *kijelölők* (selector) és *meghatározásblokkok* (declaration) felsorolása. A meghatározás két részből áll: *tulajdonságból* (property) és *értékből* (value).
  - A CSS megjegyzést vagy kommentet a a /\* és a \*/ jelek közé kell írni

```
@charset "utf-8";  
  
kijelölő {  
    tulajdonság1: érték1;  
    tulajdonság2: érték2;  
    tulajdonság3: érték3;  
    ...  
}
```

```
/*  
 * Created on : 2017.09.13., 2:53:18  
 * Author     : user  
 */  
  
@charset "utf-8";  
p.balra {  
    text-align:left;  
    color: red;  
}  
p.jobbra {
```

- Össze kell kötni a HTML fájlt és a stíluslapot
- a HTML fájl fejrészbe (<head>...</head>) kell írni a következőt:
  - A *rel* és a *type* attribútumnak mindig ez kell hogy legyen az értéke
  - A *href*-nek a stíluslap elérési útvonalát is kell tartalmaznia a HTML fájlhoz képest.

# Példa

F! DOCTYPE html>

```
...5 lines
<html>
  <head>
    <meta charset="UTF-8">
    <title>Stíluslap kitekintő - típus szelektor</title>
    <link href="alap.css" rel="stylesheet" type="text/css"/>
  </head>
  <body>
    <p class="balra">Egy hosszabb szövegben szükség lehet arra,
      hogy a szöveg különböző funkciójú részei vizuálisan is megkülönböztetők
      legyenek. Az előző fejezetben láthattuk, hogy a HTML5 számos szövegrésztípushoz rendelkezésünkre bocsát jelölőket.
    <p class="jobbra fixszeles" >minden esetben elegendők. Nincs például olyan
      A HTML5-ben ezeket a - normál szövegtől eltérő funkciójú - szövegrészeket
      így viszont nincs mód arra, hogy ezek az elemek másként legyenek formázva, mint a főszöveg.
  </body>
</html>
```

Index.html

```
@charset "utf-8";
p.balra {
  text-align:left;
  color: red;
}
p.jobbra {
  text-align:right;
}
p.kozepre {
  text-align:center;
}
p.fixszeles {
  font-family: monospace;
  margin: 15px;
}
```

alap.css

Egy hosszabb szövegben szükség lehet arra, hogy a szöveg különböző funkciójú részei vizuálisan is megkülönböztethetők legyenek. Az előző fejezetben láthattuk, hogy a HTML5 számos szövegrésztípushoz rendelkezésünkre bocsát jelölőket (idézet, definíció stb.), de ezek nem

minden esetben elegendők. Nincs például olyan HTML5-jelölő, amely segítségével megjegyzést helyezhetünk el a szövegben, stb. A HTML5-ben ezeket a - normál szövegtől eltérő funkciójú - szövegrészeket is csak `<p>` elemként jelölhetjük meg, így viszont nincs mód arra, hogy ezek az elemek másként legyenek formázva, mint a főszöveg.

# CSS3 bevezetés

- Szöveg szintű elemeknél gyakran használt stílusbeállítások
  - **A betűtípus beállítása (font-family):** A szövegeink betűtípusát állíthatjuk be. A tulajdonságnak egy vesszővel elválasztott listát adunk értékül. A lista betűtípusok neveit tartalmazza, mégpedig preferált sorrendben. Azaz, ha a böngészőprogram nem tudja az első betűtípus szerint megjeleníteni a szöveget (mert az nincs telepítve a gépre), akkor a listában a következőt próbálja majd meg beállítani. A lista végén egy általános betűcsaládot célszerű megadni, arra az esetre, ha egyik felsorolt betűtípus sincs támogatva. Ekkor ebből a betűcsaládból kerül kiválasztásra egy olyan, amely már támogatva van.
    - A **Serif** betűcsalád: - a talpas betűk tartoznak. Használatuk főleg nyomtatásban ajánlott, mivel a talpak vezetik a szemet olvasás közben. Ebbe a családba tartozik például: Times, Century, Palatino, Garamond, and Bodoni betűtípusok

Times  
Century  
Palatino  
Garamond  
Bodoni

Arial  
Comic Sans  
Gill Sans  
Franklin Gothic  
Trebuchet

ABCDEFGHIJKLM  
NOPQRSTUVWXYZ  
abcdefghijklm  
nopqrstuvwxyz  
0123456789!#  
%&\$@\*{(/{|})}

*Cursive Fonts*  
Clarkson  
Charlotte  
Heritage  
Black & White

AvQEST NORMAl	I23
Black Chancery	123456
Celtic Caps	DaleLand
Diamond Fantasy	Gismonda
Fantasy One	SandFire
Folkard	WARLORD
Nganjuk English	I23456

- A **Sans-serif** betűcsalád: nem tartalmaz talpakkal, ezáltal a megjelenése sokkal modernebb. A képernyőn jobban mutat, mint a Serif betűtípusok, ezért weboldalakra ideális választás. Az Arial, Comic, Trebuchet betűtípusok jó példák erre a családra.
- A **Monospace** betűcsaládban minden karakter azonos szélességű, így írógéppel írt hatást kelt. Lehet talpas, illetve talp nélküli. Viszonylag gyorsan olvasható, ezért rendszerint a különböző szerkesztőprogramok a programkódokat ilyen betűtípussal jelenítik meg. A Courier betűtípus ide tartozik.
- A **Cursive** betűcsaládba tartozó betűtípusok a kézirást próbálják leutánozni. Nehezen olvashatók, ezért általában csak dekoratív célokra szolgálnak.
- A **Fantasy** családhoz tartoznak azok a betűtípusok, amelyek betűket ábrázolnak azonban mégsem sorolhatók egyik előző kategóriába sem. Dekoráció céljából használják őket.

# Példa

```
<!DOCTYPE html>
...5 lines
<html lang="hu">
<head>
<title>A betütipus beállítása</title>
<meta charset="utf-8">
<style type="text/css">
    .talpatlan {font-family: Arial, Helvetica, sans-serif;}
    .talpas {font-family: 'Times New Roman', Verdana, serif;}
    .fixszeles {font-family: Courier, MS Courier New, monospace;}
    .keziras {font-family: Caflisch Script, Adobe Poetica, cursive;}
    .fantazia {font-family: Alpha Geometrique, Critter, Cottonwood, fantasy;}
</style>
</head>
<body>
    <p class="talpatlan">Az első bekezdés a Sans-serif betűcsaládból van.</p>
    <p class="talpas">A második bekezdés a Serif betűcsaládból van.</p>
    <p class="fixszeles">A harmadik bekezdés a Monospace betűcsaládból van.</p>
    <p class="keziras">A negyedik bekezdés a Cursive betűcsaládból van.</p>
    <p class="fantazia">Az ötödik bekezdés a Fantasy betűcsaládból van.</p>
</body>
</html>
```

Az első bekezdés a Sans-serif betűcsaládból van.

A második bekezdés a Serif betűcsaládból van.

A harmadik bekezdés a Monospace betűcsaládból van.

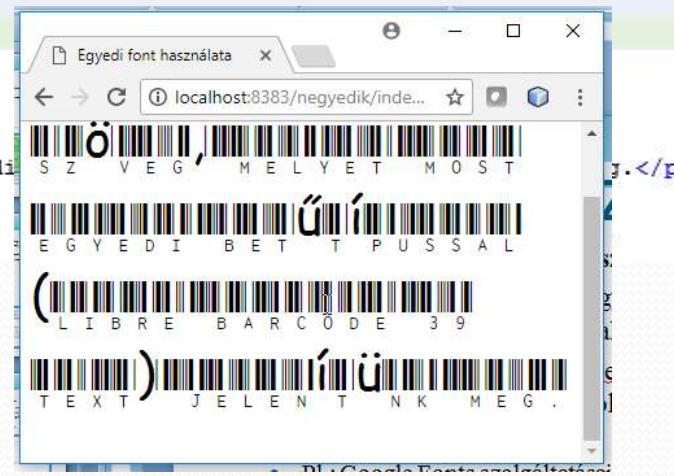
A negyedik bekezdés a Cursive betűcsaládból van.

**Az ötödik bekezdés a Fantasy betűcsaládból van.**

# CSS3 bevezetés

- **Egyedi (saját) betűtípusok használata**
- A megjelenítendő szövegeink egyedi betűtípusa esetén arra kell hagyatkoznunk, hogy a felhasználó gépén ezek közül valamelyik megtalálható- ez azonban nincs minden így.
- Eltárolhatunk betűtípusokat a webszerveren és a @font-face segítségével beimportálhatjuk őket. De mivel a különböző böngészők különböző formátumokat támogatnak, mindegyikre külön-külön gondolni kell.
  - Pl.: Google Fonts szolgáltatásai:

```
<!DOCTYPE html>
...5 lines
<html lang="hu">
<head>
<title>Egyedi font használata</title>
<meta charset="utf-8">
<link href='http://fonts.googleapis.com/css?family=Parisienne&subset=latin,latin-ext' rel='stylesheet' type='text/css'>
<style type="text/css">
body {
    font-family: 'Parisienne', cursive;
    font-size:250%;
}
</style>
</head>
<body>
<p>Szöveg, melyet most egyedi betűtípusnal mutatunk ki. A kódban a CSS-be több bar...
</body>
</html>
```



# CSS3 bevezetés

- A betűméretet a **font-size** tulajdonsággal állíthatjuk be.
- Lehetséges értékek
  - Lehetőségünk van abszolút méret megadására. Ezek növekvő sorrendben:  
*xx-small*  
*x-small*  
*small*  
*medium* (*ez az alapértelmezett*)  
*large*  
*x-large*  
*xx-large*.
  - Módunkban áll relatív érték megadására is a smaller, larger kulcsszavakkal.
  - Megadhatunk hosszúságértékeket (pl. 12px)
  - Továbbá százalékos értékeket is. (pl. 120%)

```
<!DOCTYPE html>
...5 lines
<html lang="hu">
<head>
<title>A betűméret beállítása</title>
<meta charset="utf-8">
<style type="text/css">
    p.normal      {font-size:100%}
    p.dupla       {font-size:200%}
    p.pixel       {font-size:14px}
    p.nagyobb     {font-size:larger}
</style>
</head>
<body>
    <p class="normal">A bekezdés alap betűméretű (100%).</p>
    <p class="dupla">A bekezdés dupla betűméretű (200%).</p>
    <p class="pixel">A bekezdés betűmérete 14 képpont.</p>
    <p class="nagyobb">A bekezdés betűmérete az alap méretnél nagyobb (larger).</p>
</body>
</html>
```

A bekezdés alap betűméretű (100%).

A bekezdés dupla betűméretű (200%).

A bekezdés betűmérete 14 képpont.

A bekezdés betűmérete az alap méretnél nagyobb (larger).

# CSS3 bevezetés

- A **font-style** tulajdonság segítségével *dőltté* tehetjük betűinket.

A CSS két módot is kínál dőlt betűs szövegek létrehozására.

*oblique* - dőlt stílusúra állítja a betűstílust.

*italic* - a böngésző kiválasztja az aktuális betűtípus dőlt variánsát, amennyiben az fellelhető.

*normal* - normál stílusúra állítja a betűstílust.

- A **font-weight** tulajdonság segítségével a *betűink súlyát* állíthatjuk be.

A lehetséges értékek:

*bold* – félkövérre teszi a szöveget

*relatív módon* - *bolder/lighter* a szülő elem értékhez képest növeli / csökkenti a betűk súlyát, ha lehetséges

*normal* - normálisra állítja a szöveget.

- A **font-variant** tulajdonság segítségével *kiskapitálissá* tehetjük betűinket

A lehetséges értékek:

*small-caps* - kiskapitálissá alakítja a szöveget

*normal* - normál stílusúra állítja szöveget

# Példa

```
<!DOCTYPE html>
...5 lines
<html lang="hu">
<head>
<title>A betűméret beállítása</title>
<meta charset="utf-8">
<style type="text/css">
.dolt1 {font-style: italic;}
.dolt2 {font-style: oblique;}
.normal {font-style: normal;}
.felkover {font-weight: bold;}
.normal2 {font-weight: normal;}
.felkoverebb {font-weight: bolder;}
.kiskapitalis {font-variant: small-caps;}
.normal3 {font-variant: normal;}
</style>
</head>
<body>
<p class="dolt1">Az első bekezdés italic dőlt.</p>
<p class="dolt2">A második bekezdés oblique dőlt.</p>
<p class="normal">A harmadik bekezdés nem dőlt.</p>
<hr>
<p class="felkover">Az első bekezdés félkövér.</p>
<p class="normal2">A második bekezdés betűsúlya normális.</p>
<p class="felkoverebb">A harmadik bekezdés betűsúlya az alapértelmezettéhez (normál) képest kövérebb.</p>
<hr>
<p class="kiskapitalis">Az első bekezdés kiskapitalis.</p>
<p class="normal3">A második bekezdés normál stílusú</p>
</body>
</html>
```

[ ]

Az első bekezdés italic dőlt.

A második bekezdés oblique dőlt.

A harmadik bekezdés nem dőlt.

---

Az első bekezdés félkövér.

A második bekezdés betűsúlya normális.

A harmadik bekezdés betűsúlya az alapértelmezettéhez (normál) képest kövérebb.

---

AZ ELSŐ BEKEZDÉS KISKAPITÁLIS.

A második bekezdés normál stílusú

# CSS3 bevezetés

- Szöveg szintű elemeknél gyakran használt stílusbeállítások -2

- Betűköz beállítása (**letter-spacing**)

Értéknek : pozitív vagy negatív távolságérték. A gyakorlatban **em** mértékegységgel ellátott értéket szoktak megadni, mivel az viszonyul a betűmérethez. Így ha az változik, vele változik a betűtávolság is. Negatív érték megadásakor a betűk összecsúsznak. Ha nulla értéket adunk meg, akkor a betűköz nem változik. Ezen kívül a tulajdonságnak a normal értéket is beállíthatjuk, ami a böngésző alapértelmezett betűköz távolságát állítja be, ami rendszerint változó.

- Szavak közti távolság (**word-spacing**)

Értéknek pozitív vagy negatív távolságértékeket adhatunk meg. Negatív érték megadásakor a szavak összecsúsznak. A gyakorlatban az em-es értékek használata ajánlott. Ezen felül a tulajdonságra alkalmazható a normal érték is, amivel a böngésző alapértékeire áll vissza a szóköz nagysága.

- Sorköz beállítása (**line-height**)

- Megadhatunk pozitív számértékeket, amelyekkel egy arányszámot fejezhetünk ki a jelenlegi sorközhöz képest.
- Megadhatunk pozitív hosszúságértékeket.
- Megadhatunk százalékos értékeket, amelyek arányszámként funkcionálva a befoglaló elemnél érvényes sorközhöz képest eredményeznek változást.
- A normal érték a böngészőbeli alapértékre állítja a sorközt.

- Szövegtranszformáció / kis- és nagybetűk (**text-transform**)

- Az uppercase minden betűt nagybetűvé alakít az érintett szövegben.
- A lowercase minden betűt kisbetűvé alakít az érintett szövegben.
- A capitalize minden egyes szó első betűjét nagybetűvé alakítja az érintett szövegben.
- A none felülbírálja az előző értékeket és meghagyja a szöveget eredeti formájában.

- Szövegaranyék (**text-shadow**)

- Először adjunk meg azt a színértéket, amilyen színű árnyékot szeretnénk.
- Ezt követi két hosszúságérték, ami az árnyék függőleges és vízszintes eltolásának nagyságát határozza meg. Az első érték a függőleges eltolásért felel, pozitív érték esetén lefele, negatív esetén felfele tolunk el. A második érték pedig a vízszintes eltolás nagyságát határozza meg. Pozitív érték esetén jobbra, negatív esetén balra tolunk el.
- Végül adjunk meg az árnyék nagyságát. Ezt egy pozitív hosszúságérték segítségével tehetjük meg.
- A none értéket is megadhatunk, így kikapcsolhatjuk az esetlegesen megörökült árnyékot.

# Példa

```
<!DOCTYPE html>
...5 lines
<!DOCTYPE html>
<html lang="hu">
  <head>
    <title>Betűköz beállítási példák</title>
    <meta charset="utf-8">
    <style type="text/css">
      .pelda1 {letter-spacing: normal; word-spacing: normal;}
      .pelda2 {letter-spacing: 5px; word-spacing: 5px;}
      .pelda3 {letter-spacing: .5em; word-spacing: .5em;}
      .pelda4 {letter-spacing: 1.5em; word-spacing: 2.0em;}
      .pelda5 {letter-spacing: -0.1em; word-spacing: -0.5em;}
    </style>
  </head>
  <body>
    <p class="pelda1">Az első bekezdés normál betűközzel és szóközzel.</p>
    <p class="pelda2">A második bekezdés 5 pixelnyi betűközzel és szóközzel.</p>
    <p class="pelda3">A harmadik bekezdés félszerves betűközzel.</p>
    <p class="pelda4">A negyedik bekezdés másfélszerves betűközzel és 2x szóközzel.</p>
    <p class="pelda5">Az ötödik bekezdés minusz egytizedes betűközzel -0,5-ös és szóközzel.</p>
  </body>
</html>
```

Az első bekezdés normál betűközzel és szóközzel.

A második bekezdés 5 pixelnyi betűközzel és szóközzel.

A harmadik bekezdés félszerves betűközzel. →

A n e g y e d i k b e k e z d é s      b e k e z d é s  
m á s f é l s z e r e s      b e k e z d é s  
b e t ü k ö z z e s      é s      2 x  
s z ó k ö z z e s      é s

Aötödik bekezdés minusz egytizedes betűközzel -0,5-ös szóközzel

# Példa

```
<!DOCTYPE html>
[...5 lines]
<html lang="hu">
    <head>
        <title>Sorköz beállítási példák</title>
        <meta charset="utf-8">
        <style type="text/css">
            .pelda1 {
                line-height: normal;
                text-transform: none;
                text-shadow: none;
            }
            .pelda2 {
                line-height: 30px;
                text-transform: uppercase;
                text-shadow: lightgreen 5px 5px 3px;
            }
            .pelda3 {line-height: 1.5;text-transform: lowercase; text-shadow: yellow 5px 5px 3px, lightgreen 5px -5px 10px; }
            .pelda4 {line-height: 200%;text-transform: capitalize}
            .pelda5 {line-height: 0;}
        </style>
    </head>
    <body>
        <p class="pelda1">Az első bekezdés <br> normál sorközzel.</p>
        <hr>
        <p class="pelda2">A második bekezdés nagybetsű, <br> 30 pixelnyi sorközzel.</p>
        <hr>
        <p class="pelda3">A harmadik bekezdés kisbetűs, <br> másfélszeres sorközzel.</p>
        <hr>
        <p class="pelda4">A negyedik bekezdés kiskapitális, <br> dupla sorközzel.</p>
        <hr>
        <p class="pelda5">Az ötödik bekezdés <br> sorköz nélkül.</p>
    </body>
</html>
```

Az első bekezdés  
normál sorközzel.

A MÁSODIK BEKEZDÉS NAGYBETŰS,  
30 PIXELNYI SORKÖZZEL.

a harmadik bekezdés kisbetűs,  
másfélszeres sorközzel.

A Negyedik Bekezdés Kiskapitális,  
Dupla Sorközzel.

A ötödik bekezdés

# CSS3 bevezetés

- Szöveg szintű elemeknél gyakran használt stílusbeállítások -3
- Szövegigazítás vízszintesen (**text-align**) - blokkszintű elemekre használható ( pl: <p>, <div>)Érték:
  - *left* – bal , *right* – jobb , *center* - közép , *justify* - sorkizárt, *inherit* (megörökli a szülő tulajdonságát)
  - *start*, *end* - az adott nyelv olvasási sorrendjének ( *bal*→*jobb* vagy *jobb*→*bal*) megfelelően („honnán –merre”)
  - *auto* – alapértelmezett
- Függőleges szövegigazítás (**vertical-align**)  
A vertical-align tulajdonság segítségével függőlegesen igazíthatjuk szövegeinket az őket tartalmazó tag függőleges méretéhez képest.
  - A *super* érték a felső indexbe helyezi az érintett szöveget.
  - A *sub* érték az alsó indexbe helyezi az érintett szöveget.
  - A *baseline* érték a normális helyére teszi a szöveget, ez az átlagos megjelenése a szövegeknek.Megadhatunk relatív értékeket, ami a szülő elemhez képpest adja meg a pozíciót:
  - A *top* érték a szülőelem tetejéhez igazít, a sor legmagasabb elemével egy vonalba.
  - A *middle* érték a szülőelem függőleges középpontjára igazít.
  - A *bottom* érték a szülőelem aljához igazít, a sor legmélyebben levő elemével egy vonalba.
  - A *text-top* érték a szülőelem tetejéhez igazít, a betűinek tetejéhez.
  - A *text-bottom* érték a szülőelem aljához igazít, a betűinek aljához.
  - Megadhatunk százalékos értékeket, ami a szülő elem betűméretéhez képpest igazítja a szöveget. Pozitív értékek felfele, negatívak lefelé mozdítják el szövegünket. A 100% egy teljes sornyi távolságot jelent.
  - Megadhatunk távolságértékeket is. A pozitív és negatív értékek a százalékos értékekkel analóg módon működnek.

# Példa

```
<!DOCTYPE html>
[...5 lines]
<html lang="hu">
    <head>
        <title>Vízszintes szövegigazítás</title>
        <meta charset="utf-8">
        <style type="text/css">
            p {border-bottom:1px dotted gray;} /* alsó szegély beállítása a bekezdésekre */
            .pelda1 {text-align: center;}
            .pelda2 {text-align: right;}
            .pelda3 {text-align: justify;}
            .pelda4 {text-align: left;}
            .start {text-align: start;}
        </style>
    </head>
    <body>
        <p class="pelda1">Az első bekezdés középre van igazítva.</p>
        <p class="pelda2">A második bekezdés jobbra van igazítva. I[<div style="direction:rtl">
            <p class="start">A blokkra balról jobbra olvasási sorrend van igazítva.
        </div>
        <div style="direction:rtl">
            <p class="start">A blokkra jobbról balra olvasási sorrend van igazítva.
        </div>
    </body>
</html>
```

Az első bekezdés középre van igazítva.

A második bekezdés jobbra van igazítva. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam augue sem, tempor id enim ut, rhoncus dictum ante. Aliquam erat volutpat. Pellentesque non felis vitae lorem tincidunt vestibulum. Donec rhoncus, est eu posuere tincidunt, nunc mauris molestie tortor, id ullamcorper nunc justo eget sem.

Nam egestas felis sed ante auctor, tincidunt facilisis metus convallis. Vivamus sollicitudin vestibulum felis, a varius lacus posuere dignissim.

A harmadik bekezdés sorkizárttá van téve, ami csak akkor látványos, ha a sor elér a szülő elem határáig. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam augue sem, tempor id enim ut, rhoncus dictum ante. Aliquam erat volutpat. Pellentesque non felis vitae lorem tincidunt vestibulum. Donec rhoncus, est eu posuere tincidunt, nunc mauris molestie tortor, id ullamcorper nunc justo eget sem. Nam egestas felis sed ante auctor, tincidunt facilisis metus convallis. Vivamus sollicitudin vestibulum felis, a varius lacus posuere dignissim.

A negyedik bekezdés balra van igazítva. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam augue sem, tempor id enim ut, rhoncus dictum ante. Aliquam erat volutpat. Pellentesque non felis vitae lorem tincidunt vestibulum. Donec rhoncus, est eu posuere tincidunt, nunc mauris molestie tortor, id ullamcorper nunc justo eget sem. Nam egestas felis sed ante auctor, tincidunt facilisis metus convallis. Vivamus sollicitudin vestibulum felis, a varius lacus posuere dignissim.

A blokkra balról jobbra olvasási sorrend van beállítva, az igazításnál a start érték van beállítva, ezért a bekezdés balra igazodik.

A blokkra jobbról balra olvasási sorrend van beállítva, az igazításnál a start érték van beállítva, ezért a bekezdés jobbra igazodik

# Példa

```
<!DOCTYPE html>
...5 lines
<html lang="hu">
<head>
<title>Függőleges igazítás példák</title>
<meta charset="utf-8">
<style type="text/css">
  span.felsoindex {vertical-align: super;}
  span.alsoindex {vertical-align: sub;}
  td {height: 100px; background-color: lightgray;}
  td.pelda1 {vertical-align: middle;}
  td.pelda2 {vertical-align: bottom;}
  td.pelda3 {vertical-align: top;}
  .pelda4 {vertical-align: 10px;}
  .pelda5 {vertical-align: -10px;}
</style>
</head>
<body>
  <p>Kettő<span class="felsoindex">a négyzeten</span> az négy.</p>
  <p>A vízmolekula képlete: H<span class="alsoindex">2</span>O.</p>
  <table>
    <tr>
      <td>Ez egy függőlegesen</td>
      <td class="pelda1">középre</td>
      <td class="pelda2"> le</td>
      <td class="pelda3"> fel igazított szöveg.</td>
      <td> Ez <span class="pelda4">+10 képponttal</span> van eltolva. Ez pedig <span class="pelda5">-10 képponttal</span>. </td>
    </tr>
  </table>
</body>
</html>
```

Kettő a négyzeten az négy.

A vízmolekula képlete:  $\text{H}_2\text{O}$ .

Ez egy függőlegesen	középre	le	fel igazított szöveg.	Ez +10 képponttal van eltolva. Ez pedig -10 képponttal
------------------------	---------	----	--------------------------	--

# CSS3 bevezetés

- Szöveg szintű elemeknél gyakran használt stílusbeállítások -3
- Bekezdés behúzása(**text-indent**)
  - Megadhatunk hosszúságértéket, ami lehetnek negatív. így az első sor a bekezdés többi sorához képest kintebb fog kezdődni.
  - Megadhatunk százalékos és relatív értékeket is. A százalékos értékek a szülő elem szélességéhez lesznek viszonyítva.
  - Az *inherit* értéket értékül adva a behúzás nagysága a szülő elemtől lesz megörökölve.
- Szövegdekoráció (**text-decoration**)

A text-decoration tulajdonság segítségével **dekorálhatjuk szövegeinket**.

  - Az *underline* érték aláhúzott szöveget eredményez.
  - Az *overline* érték egy felülvonást eredményez a szöveg fölött.
  - A *line-through* érték egy áthúzott szöveget hoz létre.
  - A *none* érték normál, dekorációmentes szöveget eredményez.
  - A *blink* villogó szöveget eredményez, amelynek használata viszont több szempontból sem javasolt.
- Szóköz tulajdonságainak beállítása(**white-space**)
  -

	Új sorok	Szóközök és tabulátorok	Szöveg tördelése
<b>normal</b>	Összevon	Összevon	Van tördelés
<b>nowrap</b>	Összevon	Összevon	Nincs tördelés
<b>pre</b>	Megtart	Megtart	Nincs tördelés
<b>pre-wrap</b>	Megtart	Megtart	Van tördelés
<b>pre-line</b>	Megtart	Összevon	Van tördelés

# CSS3 bevezetés

```
<!DOCTYPE html>
...5 lines
<html lang="hu">
  <head>
    <title>Bekezdés első sorának behúzása</title>
    <meta charset="utf-8">
    <style type="text/css">
      .pelda1 {text-indent: 25px; text-decoration: overline}
      .pelda2 {text-indent: 10%; text-decoration: underline}
      .pelda3 {text-indent: each-line; text-decoration: line-through}
    </style>
  </head>
  <body>
    <p class="pelda1">Az első bekezdés első sora 25 pixellel be van húzva. Lor
    <p class="pelda2">A második bekezdés első sora a szülő elem szélességének
    <p class="pelda3">Ut quis tortor a diam scelerisque suscipit sit amet vita
  </body>
</html>
```

Az első bekezdés első sora 25 pixellel be van húzva. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris augue dui, consequat fringilla tempor adipiscing, eleifend ultricies est. Maecenas accumsan quis magna quis condimentum. Ut quis tortor a diam scelerisque suscipit sit amet vitae erat. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam vitae metus sagittis, imperdiet arcu in, sagittis est. Pellentesque hendrerit, purus quis condimentum sodales, nunc nibh mattis odio, at facilisis eros nulla volutpat metus. Pellentesque ligula sapien, fringilla nec varius id, tempor sit amet orci. Proin eu tellus luctus, scelerisque lectus sed, porttitor enim. In hac habitasse platea dictumst. Aliquam erat volutpat. Nam ac eros est. Duis a tortor laoreet, lacinia mi quis, condimentum nulla. Integer vehicula vehicula massa, lobortis condimentum velit tincidunt a. Donec aliquam urna sem, sit amet sodales erat lacinia a.

A második bekezdés első sora a szülő elem szélességének 10%-ával van beljebb húzva. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris augue dui, consequat fringilla tempor adipiscing, eleifend ultricies est. Maecenas accumsan quis magna quis condimentum.

Ut quis tortor a diam scelerisque suscipit sit amet vitae erat. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam vitae metus sagittis, imperdiet arcu in, sagittis est. Pellentesque hendrerit, purus quis condimentum sodales, nunc nibh mattis odio, at facilisis eros nulla volutpat metus. Pellentesque ligula sapien, fringilla nec varius id, tempor sit amet orci. Proin eu tellus luctus, scelerisque lectus sed, porttitor enim. In hac habitasse platea dictumst. Aliquam erat volutpat. Nam ac eros est. Duis a tortor laoreet, lacinia mi quis, condimentum nulla. Integer vehicula vehicula massa, lobortis condimentum velit tincidunt a. Donec aliquam urna sem, sit amet sodales erat lacinia a.

# CSS3 bevezetés

```
<!DOCTYPE html>
[... 5 lines ...]
<html lang="hu">
  <head>
    <title>Szóköz beállítás példák</title>
    <meta charset="utf-8">
    <style type="text/css">
      div {width:100px; border:1px dotted gray;}
      .pelda1 {white-space: normal;}
      .pelda2 {white-space: nowrap;}
      .pelda3 {white-space: pre;}
      .pelda4 {white-space: pre-wrap;}
      .pelda5 {white-space: pre-line;}
    </style>
  </head>
  <body>
    <div class="pelda1">Normal Lorem ipsum dolor sit amet
      amet
    </div>
    <div class="pelda2">Nowrap Lorem ipsum dolor sit amet
      amet
    </div>
    <div class="pelda3">Pre Lorem ipsum dolor sit amet
      amet
    </div>
    <div class="pelda4">Pre-wrap Lorem ipsum dolor sit amet
      amet
    </div>
    <div class="pelda5">Pre-line Lorem ipsum dolor sit amet
      amet
    </div>
  </body>
</html>
```

I

Normal Lorem ipsum dolor sit amet	
Nowrap Lorem ipsum dolor sit amet	
Pre Lorem ipsum dolor sit amet	
Pre-wrap Lorem ipsum dolor sit amet	
Pre-line Lorem ipsum dolor sit amet	

# HTML5 Alapok

- **Csoportosító elemek -1**

- **<p>** – Bekezdés - Az adott bekezdés tartalmát a <p> és </p> elemek közé kell tenni.
- **<div>** – Blokk definiálása - nincs önálló szemantikai jelentése arra használjuk, hogy csoportba foglaljunk nagyobb blokkokat, míg a **<span>** – tag a soron belüli elemek (pl. bekezdésen belüli szavak, szócsoportok, mondatok stb.) formázásánál használható
- **<hr>** – Vízszintes elválasztó - egy elválasztó vonalat jeleníthetünk meg vele, ami CSS-sel formázható is , nincs lezárója
- **<pre>** – Előreformázott szöveg : Olyan szövegnél alkalmazhatjuk, ahol fontos szerepe van a szöveg előre formázásának, vagyis az elhelyezett szóközöknek, sortöréseknek. A <pre> tag tartalmaként megadott szöveget a böngésző azonos szélességű (monosp ace) betűtípusossal jeleníti meg. ( programkód, ASCII-ábrák, versek dalszövegek, ...)
- **<blockquote>** – Idézet - egy külső forrásból származó idézetet elhelyezésére használható

# HTML5 Alapok

```
<!DOCTYPE html>
...5 lines
<html lang="hu">
    <head>
        <title>Szöveg dekoráció példák</title>
        <meta charset="utf-8">
        <style type="text/css">
            div.pelda1 {background-color: lightgray;}
            div.pelda2 {background-color: lightyellow;}
        </style>
    </head>
    <body>
        <div class="pelda1">
            <p>Lorem ipsum dolor sit amet.</p>
            <p>Phasellus suscipit arcu lacus, at volutpat nibh porttitor id</p>
        </div>
        <hr>
        <div class="pelda2">
            <p>Lorem ipsum dolor sit amet.</p>
            <p>Phasellus suscipit arcu lacus, <br> at volutpat nibh porttitor id</p>
        </div>
        <pre>
            Fájdalomtól
            megtört szí
            vvel tudato
m hogy hosszas szennedés után tegnap
délután meghalt a Vers Az elhunytat al-
ulíró személyes halottjának teki nti s
egyb en ezúton mond|
köszö netet mind
azoknak kik virágot pacsirtát és ligetet
helyeztek el szeretett halottunk sirján
            Emléke örökké élni fog
        </pre>
        <blockquote cite="http://hu.wikipedia.org/wiki/HTML5">
            <p>
                A HTML5 a HTML (Hypertext Markup Language, a web fő jelölőnyelve) korábbi verzióinak az átdolgozott változata. A kifejlesztésének egyik fő célja, hogy a webes alkalmazásokhoz ne legyen szükség plug-inek (pl. Adobe Flash, Microsoft Silverlight, Oracle JavaFX) telepítésére.
            </p>
        </blockquote>
    </body>
</html>
```

Lorem ipsum dolor sit amet.

Phasellus suscipit arcu lacus, at volutpat nibh porttitor id.

Lorem ipsum dolor sit amet.

Phasellus suscipit arcu lacus,  
at volutpat nibh porttitor id.

Fájdalomtól  
megtört szí  
vvel tudato

m hogy hosszas szennedés után tegnap  
délután meghalt a Vers Az elhunytat al-  
ulíró személyes halottjának teki nti s  
egyb en ezúton mond  
köszö netet mind  
azoknak kik virágot pacsirtát és ligetet  
helyeztek el szeretett halottunk sirján  
Emléke örökké élni fog

A HTML5 a HTML (Hypertext Markup Language, a web fő jelölőnyelve) korábbi verzióinak az átdolgozott változata. A kifejlesztésének egyik fő célja, hogy a webes alkalmazásokhoz ne legyen szükség plug-inek (pl. Adobe Flash, Microsoft Silverlight, Oracle JavaFX) telepítésére.

# HTML5 Alapok

- Csoportosító elemek -2

- **<ol>, <li> – Sorszámozott lista**

Az **<ol>** és **</ol>** tagok közé kell elhelyeznünk a listaelemeket, amelyeket a **<li>** tagok vezetnek be.

<ol> paraméterei:

- *globális attribútumok* : Olyan paraméterek (attribútumok), amelyeket minden tag esetén használhatunk: accesskey, class , contenteditable , contextmenu , dir , draggable , dropzone , hidden , id , lang , spellcheck , style , tabindex , title , translate.
- *reversed*: egy logikai érték, ha jelen van, akkor a lista csökkenő sorszámozással jelenik meg.
- *start*: a kezdő sorszámot tudjuk beállítani.
- *type*: a felsorolásjel típusát adhatjuk meg. Az értékek az alábbi táblázatban megtalálhatóak.

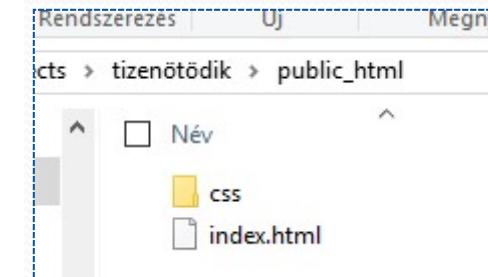
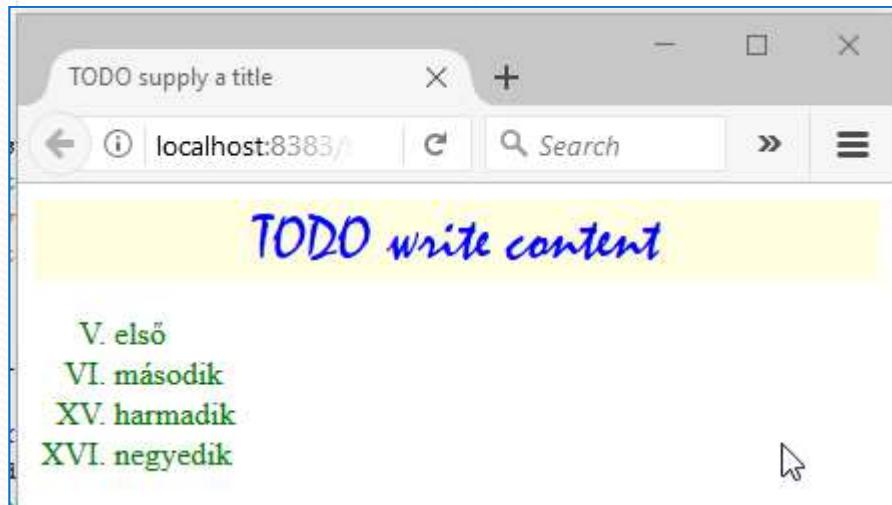
Kulcsszó	Állapot	Leírás
1	decimal	Számok (tizes számrendszer)
a	lower-alpha	Latin ábécé kisbetűi
A	upper-alpha	Latin ábécé nagybetűi
i	lower-roman	Római számok (kisbetűs)
I	upper-roman	Római számok (nagybetűk)

<li> paraméterei:

- *globális attribútumok*
- *value*: amennyiben sorszámozott listában van elhelyezve, a value paraméterrel megadható az elem sorszáma, a sorszámok kézi állítását teszi lehetővé

# HTML5 Alapok

```
<!DOCTYPE html>
...5 lines
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="css/base.css" type="text/css">
  </head>
  <body>
    <div>TODO write content</div>
    <ol type="I" start="5">
      <li>első</li>
      <li>második</li>
      <li value="15">harmadik</li>
      <li>negyedik </li>
    </ol>
  </body>
```



```
@charset "utf-8";

div{
  color: blue;
  background-color: lightyellow;
  font-size: xx-large;
  text-align: center;
  font-family: 'Mistral';
}

ol{
  color: green;
}
```

# HTML5 Alapok

- Csoportosító elemek -2

- **<ul>, <li> – Felsorolás lista**

Az <ul> és </ul> tagok közé kell elhelyeznünk a listaelemeket, amelyeket a <li> tagok vezetnek be.

<ol> paraméterei:

- *globális attribútumok* : Olyan paraméterek (attribútumok), amelyeket minden tag esetén használhatunk: accesskey, class , contenteditable , contextmenu , dir , draggable , dropzone , hidden , id , lang , spellcheck , style , tabindex , title , translate.
- *list-style-type*: listajel beállítására szolgál
- *list-style-position*: 4 tulajdonság segítségével beállíthatjuk listajelünk és az azt követő szöveg egymáshoz viszonyított pozícióját.
- *list-style-image*: tulajdonsággal saját, kép alapú listajelet állíthatunk be.

■ **disc**: egy **telített kör** lesz a listajel

■ **circle**: egy belül **üres kör** lesz a listajel

■ **square**: egy belül **telített négyzet** lesz a listajel

■ **decimal**: a listajel **számozottá** válik

■ **decimal-leading-zero**: a listajel **számozottá** válik, de vezető nullával (01, 02, 03...).

■ **upper-roman**: a listajel számozottá változik, de **római nagybetűs** számokkal

■ **lower-roman**: a listajel számozottá változik, de **római kisbetűs** számokkal

■ **upper-alpha**: a listajel számozottá változik, de csupa **nagybetűvel**, amelyek ábécésorrendben követik egymást

■ **lower-alpha**: a listajel számozottá változik, de csupa **kisbetűvel**, amelyek ábécésorrendben követik egymást

■ **lower-greek**: a listajel számozottá változik, de csupa **kis görög betükkel**, amelyek ábécésorrendben követik egymást

■ **gregorian**: a listajel számozottá változik, de a **klasszikus gregorián számozással**

■ **armenian**: a listajel számozottá változik, de **örmény számozással**

■ **none**: a listajel **nem kerül megjelenítésre**

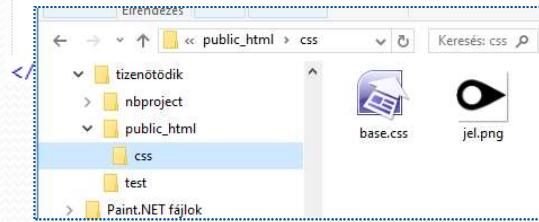
■ **inherit**: a listajel a szülő elemtől **öröklik**

- **inside**: a listajelet a böngésző **függőlegesen egy vonalba** helyezi a listaelem többi sorának bal széleivel.
- **outside**: a listaelem minden sorát annyira **húzza be**, mint amennyire a listaelem első sora van behúzva a listajel miatt.
- **inherit**: hatására az értéket a szülő elemtől **öröklik**.

- Megadhatunk egy **url**-t, ami a **szaját** listajelünk képére mutat. Ez a kép rendszerint egy jpg, gif vagy png kiterjesztésű fájl.
- A **none** érték hatására **nem lesz listajel** megjelenítve.
- Az **inherit** érték hatására a listajel a szülő elemtől **öröklik**.

# HTML5 Alapok

```
<!DOCTYPE html>
[...5 lines]
<html>
    <head>
        <title>TODO supply a title</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <link rel="stylesheet" href="css/base.css" type="text/css">
    </head>
    <body>
        <div>TODO write content</div>
        <p>Palacsintászta hozzávalói</p>
        <ul class="lista1">
            <li>20 <abbr title="dekagram">dkg</abbr> liszt (1 púpos evőkanál=2 <abbr title="dekagram">dkg</abbr>)
            <li>2 tojás</li>
            <li>3 <abbr title="deciliter">dl</abbr> tej</li>
            <li>2 <abbr title="deciliter">dl</abbr> szénsavas ásványvíz</li>
            <li>1 csipet só</li>
            <li>3/4 <abbr title="deciliter">dl</abbr> étolaj</li>
        </ul>
        <ul class="lista2">
            <li>Első listaelem.</li>
            <li>Második listaelem.</li>
            <li>Harmadik listaelem.</li>
            <li>Negyedik listaelem.</li>
        </ul>
        <ul class="lista3">
            <li>Első listaelem.</li>
            <li>Második listaelem.</li>
            <li>Harmadik listaelem.</li>
            <li>Negyedik listaelem.</li>
        </ul>
    </body>
```



```
@charset "utf-8";
div{
    color: blue;
    background-color: lightyellow;
    font-size: xx-large;
    text-align: center;
    font-family: 'Mistral';
}

.lista1{
    color: green;
    list-style-type: circle;
}

.lista2{
    color: purple;
    background-color: lightgrey;
    font-family: 'Comic Sans';
    list-style-type: square;
}

.lista3{
    list-style-image: url("../css/jel.png");
    list-style-position: inside;
}
```

Palacsintászta hozzávalói

- 20 dkg liszt (1 púpos evőkanál=2 dkg)
- 2 tojás
- 3 dl tej
- 2 dl szénsavas ásványvíz
- 1 csipet só
- 3/4 dl étolaj

Első listaelem.

Második listaelem.

Harmadik listaelem.

Negyedik listaelem.

Első listaelem.

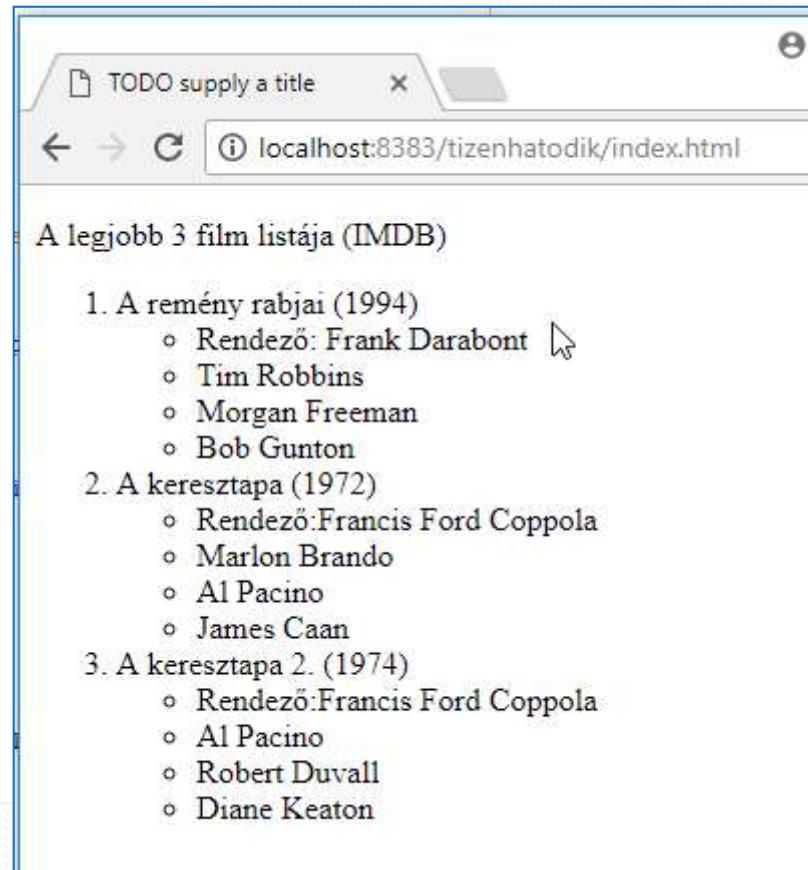
Második listaelem.

Harmadik listaelem.

Negyedik listaelem.

# HTML5 Alapok

```
<!DOCTYPE html>
...5 lines
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <p>A legjobb 3 film listája (IMDB)</p>
    <ol>
      <li>A remény rabjai (1994)
        <ul>
          <li>Rendező: Frank Darabont</li>
          <li>Tim Robbins</li>
          <li>Morgan Freeman</li>
          <li>Bob Gunton</li>
        </ul>
      </li>
      <li>A keresztapa (1972)<ul>
        <li>Rendező: Francis Ford Coppola</li>
        <li>Marlon Brando</li>
        <li>Al Pacino</li>
        <li>James Caan</li>
      </ul></li>
      <li>A keresztapa 2. (1974)<ul>
        <li>Rendező: Francis Ford Coppola</li>
        <li>Al Pacino</li>
        <li>Robert Duvall</li>
        <li>Diane Keaton </li>
      </ul></li>
    </ol>
  </body>
</html>
```



# HTML5 Alapok

- Csoportosító elemek -2

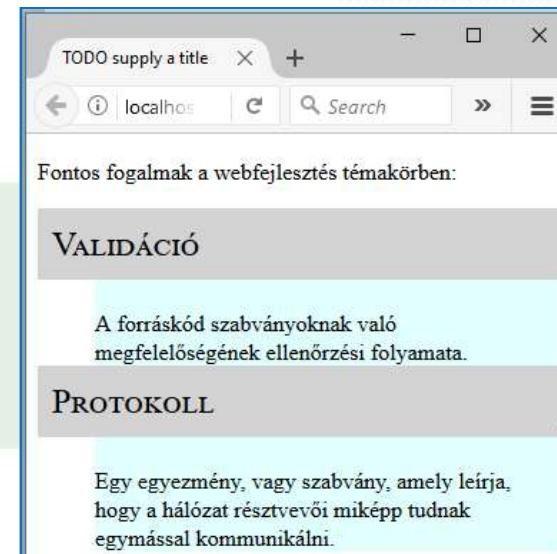
- **<dl><dd><dt> – Definíciós lista**

Ezen elemekkel úgynevezett definíciós, vagy meghatározás listát készíthetünk. A `<dl>` és `</dl>` tagok közé kell elhelyeznünk a listaelemeket. A fogalmat `<dt>` és `</dt>`, a magyarázatát `<dd>` és `</dd>` tagok közé kell tennünk.

`<dl><dd><dt>` paraméterei:

- *globális attribútumok* : Olyan paraméterek (attribútumok), amelyeket minden tag esetén használhatunk: `accesskey`, `class`, `contenteditable`, `contextmenu`, `dir`, `draggable`, `dropzone`, `hidden`, `id`, `lang`, `spellcheck`, `style`, `tabindex`, `title`, `translate`.

```
<!DOCTYPE html>
[...5 lines]
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style type="text/css">
      dt{
        font-family: 'Comic sans';
        font-variant: small-caps;
        font-size: 1.5em;
        padding: 10px;
        background-color: lightgrey;
      }
      dd{
        padding-top: 20px;
        background-color: lightcyan;
      }
    </style>
  </head>
  <body>
    <p>Fontos fogalmak a webfejlesztés téma körben:</p>
    <dl>
      <dt>Validáció</dt>
        <dd>A forráskód szabványoknak való megfelelőségének ellenőrzési folyamata.</dd>
      <dt>Protokoll</dt>
        <dd>Egy egyezmény, vagy szabvány, amely leírja, hogy a hálózat résztvevői miképp tudnak egymással kommunikálni.</dd>
    </dl>
  </body>
</html>
```



# HTML5 Alapok

- Csoportosító elemek -2

- **<figure>, <figcaption> – Kép, ábra, kép/ábra aláírása**

- A <figure> ábrák, fényképek, illusztrációk, diagramok jelölésére szolgál.
- A megjelenítendő tartalom lehet:
  - kép, amit az <img> tag segítségével lehet beilleszteni
  - Multimédiaelem
  - forráskód is ekkor a <code> taget lehet használni.

```
<!DOCTYPE html>
...5 lines
<html>
  <head>
    <title>Figure</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style type="text/css">
      figure{
        border: blueviolet dotted medium ;
      }
      figcaption{
        background-color: hotpink;
      }
    </style>
  </head>
  <body>
    <div>A számláló (vagy iteráló) ciklus általánosságban olyan előtesztelő ciklust jelent, amely egy felsorolható típus adott intervallumán léptet végig, speciálisan egész számokon. Üres intervallumra nem fut le.C++-ban a for ciklus
      <figure>
        <pre><code>
int sum = 0; // Kezdetben a szumma 0
for(int i=1; i<201; ++i) // Kezdőérték; feltétel; növelés
  sum = sum + i; // Az i hozzáadása sum-hoz
</code>
</pre>
        <figcaption> ábra aláírás lent: Ciklussal bekérjük a tömb elemeit</figcaption>
      </figure>
      majdnem teljesen ekvivalens az elői tesztelős ciklussal (a continue viselkedése más), de :
    </div>

    <figure>
      <q cite="http://www.citatum.hu/szerzo/Csokonai_Vitez_Mihaly">
        A hatámas szerelemnek<br />
        Megemészítő tüze bánt.<br />
        Te lehetsz írja sebemnek,<br />
        Gyönyörű kis tulipánt!
      </q>
      <figcaption>Csokonai Vitéz Mihály: <cite>Tartózkodó kérelem (részlet)</cite></figcaption>
    </figure>
  </body>
</html>
```

A számláló (vagy iteráló) ciklus általánosságban olyan előtesztelő ciklust jelent, amely egy felsorolható típus adott intervallumán léptet végig, speciálisan egész számokon. Üres intervallumra nem fut le.C++-ban a for ciklus

```
int sum = 0; // Kezdetben a szumma 0
for(int i=1; i<201; ++i) // Kezdőérték; feltétel; növelés
  sum = sum + i; // Az i hozzáadása sum-hoz
```

ábra aláírás lent: Ciklussal bekérjük a tömb elemeit

majdnem teljesen ekvivalens az elői tesztelős ciklussal (a continue viselkedése más), de más nyelvekben nem feltétlenül van ez így: a for ciklus átirányíthat más ciklussá, de visszafelé általában nem. A for ciklusnak kisebb lehet a kifejezőreje, de könnyebb olvasni ezért a használata javasolt ott ahol ez lehetséges.

\* A hatámas szerelemnek  
Megemészítő tüze bánt.  
Te lehetsz írja sebemnek,  
Gyönyörű kis tulipánt! ”  
Csokonai Vitéz Mihály: *Tartózkodó kérelem (részlet)*

# HTML5 Alapok

- Beágyazott elemek

- <img> – Kép beszúrása

- <img>paraméterei:

- *globális attribútumok*
      - *src*: itt adhatjuk meg a kép forrását (kötelező!).
      - *alt*: az alternatív, helyettesítő szöveg megadására szolgál (kötelező!). Ebben a paraméterben kell megadnunk azt szövegesen, hogy mi látható a képen. minden esetben töltük ki megfelelő értékkel!!
      - *longdesc*: a kép hosszú leírását tartalmazó fájl url-je kerül ide. Amikor az alt paraméterben már nem tudjuk teljes értékűen leírni a kép tartalmát (pl. bonyolult ábra, diagram), akkor megtehetjük, hogy a leírást egy külön HTML-állományban helyezzük el, és az ezen oldalra mutató linket tesszük a longdesc paraméterbe.
      - *width*: a kép az itt megadott szélességgel jelenik meg.
      - *height*: a kép az itt megadott magassággal jelenik meg.

*Megjegyzés: Ha a width és a height attribútum közül csak az egyiknek adunk értéket, a böngészőprogram a másik paraméter értékét automatikusan, a kép eredeti oldalarányainak megfelelően megváltoztatja. Ha minden két paraméter értékét kitöljük, a képet úgy is átméretezzük, hogy az eredeti oldalarányok nem maradnak meg, vagyis a kép eltorzulhat.*

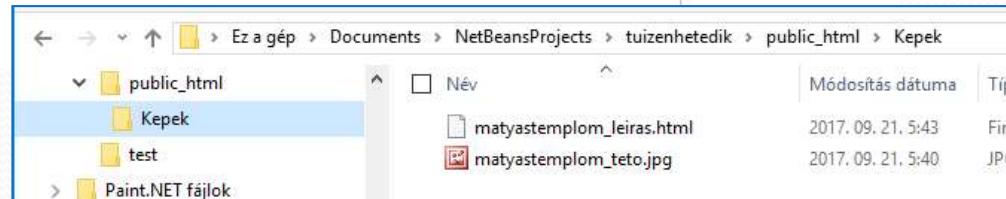
- *usemap*: Az <img> tag usemap attribútuma egy kliensoldali térképet kapcsol hozzá a képhez, amellyel gyakorlatilag a kép egyes részeit kattinthatóvá tehetjük
    - *ismap*: azt jelenti, hogy a kép része egy szerveroldali térképnak (vagyis egy képnek, amelyen kattintható területek vannak meghatározva). Amikor egy ilyen képre kattintunk, a koordinátákat a böngésző elküldi a szervernek. Csak akkor engedélyezett a használata, ha az <img> tag egy <a> tag leszármazottja.

# HTML5 Alapok

```
<!DOCTYPE html>
...5 lines
<html>
    <head>
        <title>Figure</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <style type="text/css">
            figure{
                border: cadetblue dotted thin;
                width: max-content;
                display: block;
            }
            figcaption{
                background-color: lightgreen;
            }
        </style>
    </head>
    <body>
        <div>A számláló (vagy iteráló) ciklus általánosságban olyan előtesztelő ciklust jelent, amely egy felsorolható típus adott intervallumán léptet végig, speciálisan egész számokon. Üres intervallumra nem fut le.C++-ban a for ciklus
            
            majdnem teljesen ekvivalens az elől tesztelős ciklussal (a continue viselkedés
        </div>

        <figure>
            
            <figcaption>A Mátyás-templom tetőcserepei</figcaption>
        </figure>

    </body>
</html>
```



A számláló (vagy iteráló) ciklus általánosságban olyan előtesztelő ciklust jelent, amely egy felsorolható típus adott intervallumán léptet végig, speciálisan egész számokon. Üres intervallumra nem fut le.C++-ban a for ciklus



majdnem teljesen ekvivalens az elől tesztelős ciklussal (a continue viselkedése más), de más nyelvekben nem feltétlenül van ez így: a for ciklus átírható más ciklussá, de visszafelé általában nem. A for ciklusnak kisebb lehet a kifejezőreje, de könnyebb olvasni ezért a használata javasolt ott ahol ez lehetséges.



A Mátyás-templom tetőcserepei

# HTML5 Alapok

- Beágyazott elemek

- Kliensoldali térkép létrehozása

- Olyan kép melyen részei linkként viselkednek
    - A létrehozáshoz szükség lesz a *<map>*, *<area>* tag-ekre és az *<img>* tag *usemap* paraméterére is
      - *<map>* - A képhez tartozó képtérkép elhelyezésére szolgál.

Paraméterek:

*name:* ezen attribútum segítségével adhatunk egyedi nevet a képtérképunknek. pl. *<map name="terkep1">*. Később erre a névre kell hivatkoznunk annál a képnél, ahol a térképet fel akarjuk használni. Erre szolgál a *usemap* paraméter. Pl. **

- *<area>* - Ezen tag segítségével tudjuk meghatározni a különböző alakú területeket a képen, amelyek linkként viselkednek.

Paraméterek:

*alt:* ua. mint kép esetén alternatív szöveg

*shape:* azt adja meg, hogy ilyen alakú területet szeretnénk létrehozni. Ez téglalap, kör és sokszög lehet.

- default : a teljes kép lesz a terület → nem kell a *coords* paraméter
- rect : téglalap alakú területet hozhatunk létre.
- circle : kör alakú területet definiálhatunk.
- poly : sokszöggel leírható területet hozhatunk létre.

*coords:* A terület meghatározó pontjainak koordinátáját állíthatjuk be. a koordinátarendszer origója, vagyis a (0,0) pontja a kép bal felső sarka lesz. Az x koordináta jobbra nő, az y koordináta pedig lefelé, vagyis itt csak 0 és pozitív koordináták adhatók meg vesszővel elválasztva.

- rect : a téglalap bal felső, illetve jobb alsó koordinátái → *shape="rect"*, *coords="15,30,150,300"*
- circle : középpont koordinátái, és a sugár → *shape ="circle"* *coords ="50,50,25"*
- poly: : A határoló pontok koordinátái → *shape ="poly"* *coords ="50,70,150,240,370,120"*

# HTML5 Alapok

```
<!DOCTYPE html>
...5 lines
<html lang="hu">
<head>
<title>Képtérkép létrehozása</title>
<meta charset="utf-8">
</head>
<body>
<map name="logok">
<area shape="circle"
      coords="87,101,73"
      alt="ELTE honlapja"
      target="ELTE" href="http://www.elte.hu" />
<area shape="poly"
      coords="196,19,196,50,206,160,253,174,296,163,305,50,296,20"
      alt="HTML5 honlapja"
      target="HTML5"
      href="http://www.w3.org/TR/html5/" />
<area shape="rect"
      coords="339,50,501,153"
      alt="W3C honlapja"
      target="W3C"
      href="http://www.w3c.hu" />
</map>


</body>
</html>
```



# HTML5 Alapok

- Beágyazott elemek

- **<audio>, <source> – Hangállomány beillesztése**

- Audió (hang) állomány beágyazására használhatjuk. Újdonság, hogy a HTML5 szabványt támogató böngészők alapértelmezetten támogatják a hangállományok lejátszását. Sajnos azonban az egyes webböngészők más-más formátumokat támogatnak, ezért ugyanazt a hanganyagot többféle formátumba is át kell konvertálnunk, és meg kell adnunk forrásként.

**Az <audio> tag paraméterei:**

- *src*: a hangállomány forrása (opcionális), az a *<source>* tagnél praktikus megadni
- *autoplay*: ezzel a logikai attribútummal elérhetjük, hogy a hangállomány automatikusan elinduljon betöltődéskor.
- *preload*: pufferelési technika megadása, ha az *autoplay* attribútum meg van adva, akkor figyelmen kívül hagyja a böngésző.

A preload értékei:

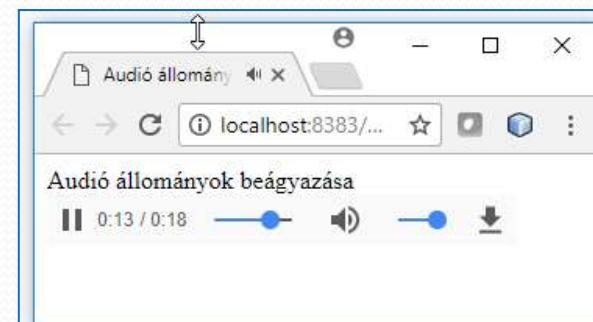
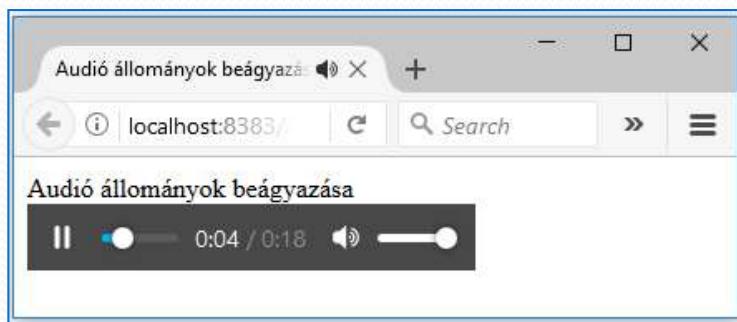
- *auto*: a hanganyag letöltése fontos, még akkor is, ha a felhasználó nem kattint rá a lejátszás gombra.
- *metadata*: nem szükséges pufferelni a hanganyagot, elég a metaadatok (pl. hang hossza) lekérése.
- *none*: ezzel jelezhetjük, hogy az audio állomány nem szükséges pufferelni.
- *loop*: végtelenítve játszódik le a hanganyag
- *muted*: a hanganyag alapesetben legyen-e elnémítva.
- *controls*: érdemes használni, ha a böngészőprogram által biztosított eszköztárat szeretnénk felkínálni a felhasználóknak, nem pedig mi oldjuk meg scriptek segítségével a hanganyag lejátszását, megállítását, hangerőváltoztatást stb.
- *mediagroup*: Lehetővé teszi több hang- illetve videósáv egyszerre történő, szinkronizált lejátszását. Ez azt jelenti, hogy (ha a böngésző támogatja,) az egyik médialelem lejátszásakor a másik is automatikusan lejátszódik az adott pozíciótól.

**A <source> tag paraméterei:**

- *src*: a hangállomány forrása
- *type*: a hangállomány típusát írhatjuk le. Itt a hangformátum mime-típusát írhatjuk le, illetve még pontosabban azt, hogy milyen kodek lett használva a hangállomány konvertálásakor.

# HTML5 Alapok

```
<!DOCTYPE html>
...5 lines
<html lang="hu">
  <head>
    <title>Audió állományok beágyazása</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <div>Audió állományok beágyazása</div>
    <audio controls autoplay>
      <source src="audio_files/lyricslap.mp3" type="audio/mpeg">
      <source src="audio_files/lyricslap.ogg" type="audio/ogg">
      <source src="audio_files/lyricslap.wav" type="audio/wav">
    </audio>
  </body>
</html>
```



# HTML5 Alapok

- Beágyazott elemek

- **<video>, <source> – videó állományok beillesztésére**
- A tag segítségével videót ágyazhatunk be az oldalunkra. Újdonság, hogy a HTML5 szabványt támogató böngészők alapértelmezetten támogatják a videóállományok lejátszását. Sajnos azonban a formátumok esetén ugyanaz a helyzet, mint amit a hangállományoknál is tapasztaltunk, vagyis az egyes webböngészők más-más formátumokat támogatnak, ezért ugyanazt a videót többféle formátumba is át kell konvertálnunk, és meg kell adnunk forrásként.

Az **<video>** tag paraméterei:

- *src*: a hangállomány forrása (opcionális) , az a *<source>* tagnél praktikus megadni
- *autoplay*: ezzel a logikai attribútummal elérhetjük, hogy a hangállomány automatikusan elinduljon betöltődéskor.
- *preload*: pufferelési technika megadása, ha az *autoplay* attribútum meg van adva , akkor figyelmen kívül hagyja a böngésző. A preload értékei:
  - *auto*: a videoanyag letöltése fontos, még akkor is, ha a felhasználó nem kattint rá a lejátszás gombra.
  - *metadata*: nem szükséges pufferelni a videoanyagot, elég a metaadatok (pl. video hossza) lekérése.
  - *none*: ezzel jelezhetjük, hogy az video állományt nem szükséges pufferelni.
- *loop*: végtelenítve játszódik le a videoanyag
- *muted*: a videólejátszó alapesetben legyen-e elnémítva.
- *controls*: ua . mint az audiónál
- *mediagroup*: ua . mint az audiónál
- *width*: a videó a megadott szélességen jelenik meg.
- *height*: a videó a megadott magassággal jelenik meg.
- *poster*: a kezdőkép. Beállítására szolgál

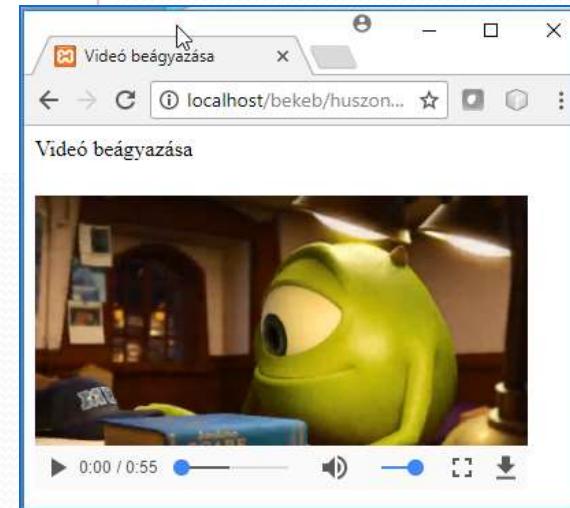
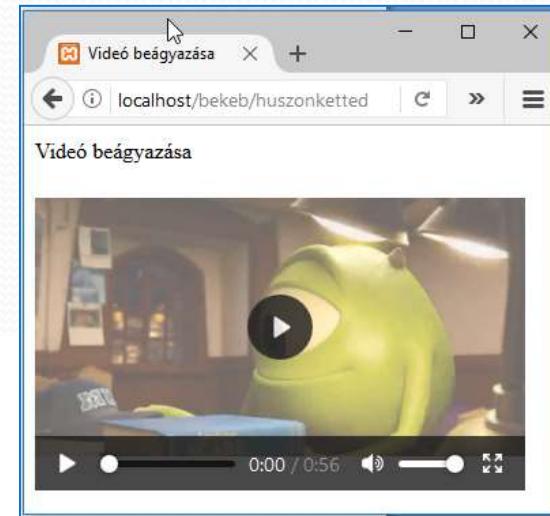
Böngésző	MP4	WebM	Ogg
Internet Explorer	Igen	Nem	Nem
Chrome	Igen	Igen	Igen
Firefox	Igen	Igen	Igen
Safari	Igen	Nem	Nem
Opera	Igen (Opera 25 óta)	Igen	Igen

A **<source>** tag paraméterei:

- *src*: a hangállomány forrása
- *type*: a hangállomány típusát írhatjuk le. Itt a video formátum mime-típusát írhatjuk le, illetve még pontosabban azt, hogy milyen kodek lett használva a video állomány konvertálásakor.

# HTML5 Alapok

```
<!DOCTYPE html>
...5 lines
<html lang="hu">
  <head>
    <title>Videó beágyazása</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <div>Videó beágyazása</div>
    <video width="360" height="240" poster="video_files/Poster.png" controls>
      <source src="video_files/demo.mp4" type="video/mp4">
      <source src="video_files/demo.ogv" type="video/ogg">
      <source src="video_files/demo.webm" type="video/webm">
    </video>
  </body>
</html>
```



# HTML5 Alapok

- Beágyazott elemek

- **<iframe> – Beágyazott keret**

- A beágyazott keret nem más, mint egy adott méretű terület, amelyben egy külső weblapot tudunk megjeleníteni.

**Az <iframe> tag paraméterei:**

- *src*: a megjelenítendő állomány webcíme.
- *width*: a beágyazott keret vízszintes mérete.
- *height*: a beágyazott keret függőleges mérete.
- *srcdoc*: a keretben megjelenő tartalom közvetlen megadása HTML kódokkal.
- *name*: nevet adhatunk a beágyazott keretnek, hogy hivatkozhassunk rá.
- *sandbox*: az iframe-be betöltött dokumentumra vonatkozó megszorításokat ill. engedélyeket lehet megadni benne. Például: szkriptek futtatása (pl. JavaScript), tartalom betöltése az oldalról ami a taget tartalmazza, űrlapok elküldése, és így tovább.
- *seamless*: új logikai típusú attribútum, melynek beállításával elérhetjük, hogy a keret tartalma úgy jelenjen meg, mintha a szülő dokumentum közvetlen része lenne. (Nem látszik szegély a beágyazott keret körül.)
- *allowfullscreen*: logikai attribútum, mellyel megadhatjuk, hogy a beágyazott oldal megjelenhet-e teljes képernyőn.

# HTML5 Alapok

```
<!DOCTYPE html>
...5 lines
<html>
  <head>
    <title>iframe (beágyazott keret) használata</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style type="text/css">
      h1, h2 {
        font-family: 'MV Boli', cursive;
        text-shadow: grey 1px 2px 1px;
      }
    </style>
  </head>
  <body>
    <div><h1>iframe használata</h1></div>
    <div>
      <iframe src="http://www.wikipedia.hu"
              width="600"
              height="300"
              seamless>
    </iframe>
    </div>
    <div>
      <iframe srcdoc=<h2>Srcdoc paraméter használata</h2><img src='http://lorempixel.com/300/300/people/'>
              width="600"
              height="380"
              seamless>
    </iframe>
    </div>
  </body>
</html>
```

## iframe használata

The screenshot shows a web browser window displaying the Wikipedia homepage. The title bar says "iframe használata". The main content area shows the Wikipedia logo and the text "Üdvözünk a Wikipédiaban!". Below the logo, there is a sidebar with links like "Kezdőlap", "Tartalom", "Kiemelt szócikkek", and "Friss változtatások". On the right side, there is a sidebar with categories such as "Társadalom", "Történelem", "Földrajz", "Kultúra", "Művészet", and "Sport". A search bar at the top right says "Keresés ezen: Wikipédia".

## Srcdoc paraméter használata



# HTML5 Alapok

- Beágyazott elemek

- **<embed> – külső tartalom beágyazása**
- Külső tartalom beágyazására szolgáló tag. Ez lehet akár Flash vagy más (interaktív) alkalmazás (tipikusan nem HTML-állomány).

Az **<embed>** tag paraméterei:

- *src*: a beágyazandó tartalom webcíme.
- *type*: a beágyazandó tartalom mime-típusa.
- *width*: a beágyazandó tartalom szélessége.
- *height*: a beágyazandó tartalom magassága

```
<!DOCTYPE html>
...5 lines
<html>
  <head ...13 lines />
  <body>
    <div>SVG Beágyazása:</div>
    <div> <embed type="image/svg+xml"
      src="kepek/NewTuxKis.svg"
      height="180"
    >
    </div>
    <div>Youtube Videó Beágyazása:
      <div><embed type="video/mp4"
        src="https://www.youtube.com/embed/VPwpMWucvQs"
        width="480"
        height="360"
      >
      </div>
    </div>
  </body>
</html>
```

SVG Beágyazása:



Youtube Videó Beágyazása:



# HTML5 Alapok

- Beágyazott elemek

- **<object> és <param>– Objektum beágyazására**

- Külső erőforrás beillesztésére, amely lehet egy kép, egy beágyazott böngészési környezet, vagy egy olyan erőforrás, amelynek lejátszásához valamilyen plug-in szükséges. Ezen objektumnak a <param> tag használatával tudunk paramétert adni.

**Az <object> tag paraméterei:**

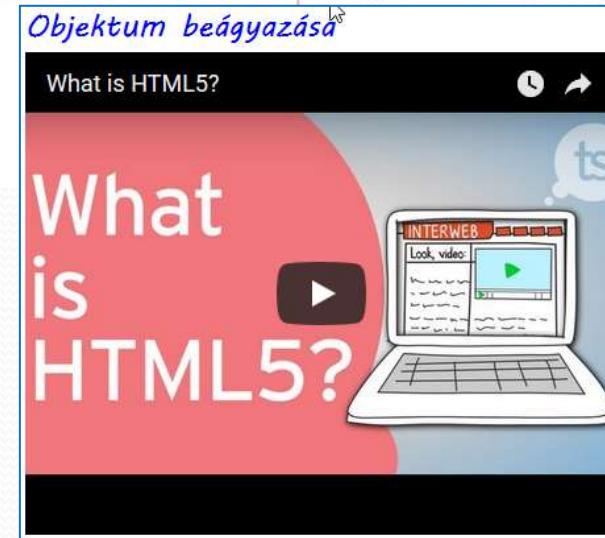
- *data*: az objektum webcíme.
- *type*: az objektum mime-típusa.
- *typemustmatch* : logikai attribútum. Ha szerepelhetünk, akkor az objektum csak akkor fog funkcionálni, ha a type paraméterben megadott mime típus ténylegesen megegyezik a letöltött elem tartalom típusával.
- *name*: az objektumnak egyedi nevet adhatunk.
- *usemap*: az objektumhoz tartozhat képtérkép, ennek az azonosítóját tudjuk itt megadni.
- *form*: ha az objektum egy űrlaphoz tartozik, akkor az űrlap nevét kell ide írni.
- *width*: az objektum szélessége.
- *height*: az objektum magassága.

**Az <param> tag paraméterei:**

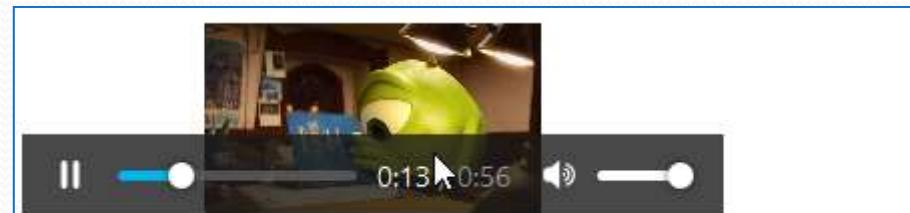
- *name*: a paraméter neve.
- *value*: a paraméter értéke.

# HTML5 Alapok

```
<div>
  <object type="application/x-shockwave-flash"
    style="width:450px; height:366px;"
    data="http://www.youtube.com/v/mzPxo7Y6JyA?color2=FBEC9E&version=3">
    <param name="movie"
      value="http://www.youtube.com/v/mzPxo7Y6JyA?color2=FBEC9E&version=3" >
    <param name="allowFullScreen"
      value="true" >
    <param name="allowScriptAccess"
      value="always" >
  </object>
```



```
<object type="video/mp4"
  data="kepek/demo.mp4"
  width="350"
  height="95">
</object>
```



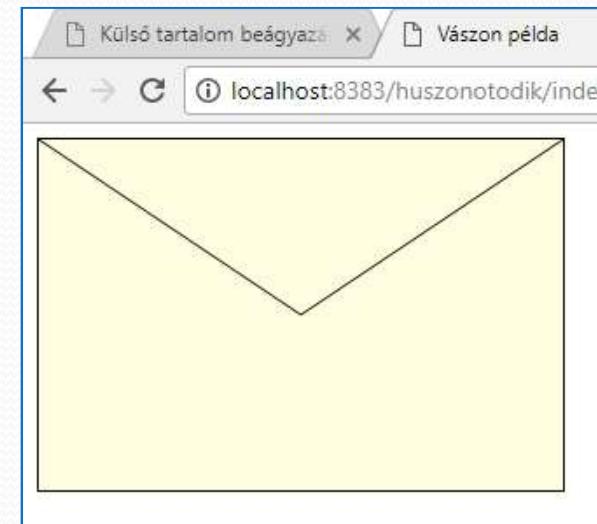
# HTML5 Alapok

- Beágyazott elemek

- A **HTML5 vászon (canvas) használata**
  - A <canvas> tag segítségével egy rajzvásznat lehet beilleszteni az oldalba. A rajzvászon tartalma JavaScript segítségével módosítható. Ezáltal dinamikusan jeleníthető meg rajta bármilyen kép, grafika vagy egy videó aktuális képe, alkalmas webes játékok fejlesztésére, animációk készítésére ... stb.
  - Egy egyszerű (üres) vászon létrehozása az alábbi módon történik:

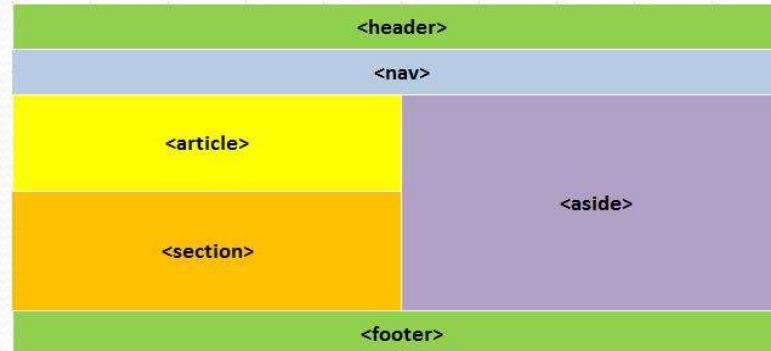
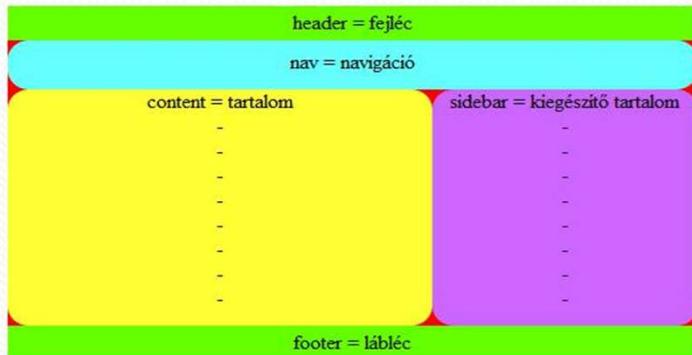
- A vászon  <canvas id="rajzvaszon" width="300" height="200"></canvas>
- A vászon szélessége és magassága a width és a height attribútumok segítségével adható meg

```
<!DOCTYPE html>
...5 lines
<html lang="hu">
  <head ...4 lines />
  <body>
    <canvas id="rajzvaszon" width="300" height="200"></canvas>
    <script>
      var cnv = document.getElementById('rajzvaszon');
      var ctx = cnv.getContext('2d');
      rajzvaszon.style.border = "black 1px solid";
      ctx.fillStyle = 'lightyellow';
      ctx.fillRect(0, 0, 300, 200);
      ctx.moveTo(0, 0);
      ctx.lineTo(150, 100);
      ctx.lineTo(300, 0);
      ctx.stroke();
    </script>
  </body>
</html>
```



# HTML5 Alapok

- Oldalszerkezet elemek



- **<body>**
  - A HTML dokumentum törzse, ebben kell elhelyezni az oldalon megjelenő tartalmakat (szöveg, kép, táblázat stb.)
- **<h1>, <h2>, <h3>, <h4>, <h5>, <h6>**
  - Segítségükkel tagolhatjuk a tartalmat fejezetekre és alfejezetekre, valamint a navigációt is segítik.
- **<main>**
  - Az oldal fő tartalmi egységének megjelölésére szolgál.
  - *csak egy* ilyen elemet helyezhetünk el a dokumentumban
  - nem lehet gyerekeleme a következő oldalszerkezet elemeknek: <article>, <aside>, <footer>, <header>, <nav>
- **<article>**
  - itt helyezhetjük el az oldal tartalmi egységeit, ami lehet egy fórumbejegyzés, újságcikk, blogbejegyzés, vagy bármilyen más független tartalom
  - lehet saját fejléce és lábléce, amelyek tartalmazhatja például a cikk címét, alcímét, publikálás dátumát, a szerző neve, elérhetősége
  - Az <article> elemek egymásba ágyazhatók

# HTML5 Alapok

- **Oldalszerkezet elemek**

- **<header>** – fejléc megadása
  - ez a rész tartalmaz bevezető információt a weboldalról vagy annak egy részéről.
  - nem egy külön része az oldalnak, hanem az egyes részeknek is lehet saját fejléce
- **<footer>** – lábléc jelölésére szolgáló tag
  - az oldal egészére, vagy egy különálló részre vonatkozóan is megadható
  - webhellyel kapcsolatos információkat tartalmazza, mint például a készítő neve, kapcsolódó dokumentumok linkjei, szerzői jogi információk stb.
- **<section>** – szakasz
  - az oldal egy nagyobb részét foglalja magába
  - azonos témahez tartozó elemek csoporthoz kötött címe van pl.: egy <section> tagbe tartoznak a fejezetek, fülekkel ellátott oldalak egyes füleinek tartalma. Egy honlap például felbontható szakaszokra az alábbiak szerint: bemutatkozás, hírek, elérhetőség
  - A <section> elemek egymásba ágyazhatók
- **<nav>** – navigációs elemek
  - az oldal azon részének van fenntartva, amely más oldalakra, vagy az adott lap egy bizonyos területére mutató linkeket tartalmaz.
  - előfordulhatnak benne más elemek is, mint például címsorok
- **<aside>** – Kapcsolódó, járulékos információ (jellemzően oldalsáv)
- Az tag tartalma kapcsolódik az oldal fő részéhez és érintőlegesen kapcsolódik a körülötte levő tartalomhoz
  - Általában oldalsáv formájában jelenik meg
- **<address>** –Kontakt információ megadása
  - Kontakt információ megadására használható, amely a legközelebbi <article>, vagy a <body> elemre vonatkozik.
  - tipikus, hogy a <footer> elemben kerül elhelyezésre