**Fontos előismeretek:**

A szöveget, amelyet olvasunk, és a képeket , amelyeket látunk, a web böngészőnk állítja elő az egyes fájlokban található utasítások alapján. A fájlok jelölő- vagy leírókóddal ellátott szöveget tartalmaznak; ezek a HTML kódok mondják meg a böngészőnek, hogy miként kell megjeleníteni a szöveget - címsorként, bekezdésként, piros betűkkel, és így tovább. Egyes HTML-leírókódok arra utasítják a böngészőt, hogy egyszerű szöveg helyett egy képet vagy egy mozgóképes állományt jelenítsen meg, és ezzel máris visszakanyarodtunk a lényeghez: a web böngészőnk különféle típusú tartalmakat kap. A webes tartalom alapvetően egy egyszerű szövegfájllal kezdődik, amely HTML- vagy XHTML-leírókódot tartalmaz. A kódunk valószínűleg mindig "működni" fog, legalábbis amíg a webböngészők tartják magukat a szabványokhoz .

A webes tartalmat "helyben", vagyis a saját merevlemezünkről is megtekinthetjük, anélkül,hogy szükség lenne egy webkiszolgálóra. A böngésző megvizsgálja és értelmezi a HTML-fájlban található tartalmat és kódokat- de az út rövidebb, mert a böngészőnek a saját számitógépünk merevlemezén kell kutakodnia, nem pedig egy távoli gépen.

A webes tartalom nem csupán oldalakat jelent, hanem a képek, valamint a hang- és videófájlok is a részét képezik. A . html kiterjesztés árulja el a webkiszolgálónak, hogy a fájl HTML-kódot tartalmaz, és amikor a kiszolgáló elküldi a fájl tartalmát a böngészőnek, a böngésző is innen tudja, hogy HTML-ről van szó, és ennek megfelelően képezi le a fájlt.

**Fájlok átvitele ftp segítségével:**

A webes tartalmat egy webkiszolgálón kell elhelyeznünk, hogy mások számára hozzáférhetővé tegyük. Ezt a műveletet rendszerint a fájlátviteli protokoll (File Transfer Protocol, FTP) segítségével hajtjuk végre. Az FTP használatához egy FTP-ügyfélre van szükségünk, vagyis egy olyan programra, amellyel átvihetjük a fájlokat a számítógépünkről a webkiszolgálóra(pl. FileZilla).

Az FTP-ügyfelek háromféle információt igényelnek ahhoz, hogy kapcsolatot tudjanak teremteni a webkiszolgálóvaL Ezeket az információkat a webgazda adja meg nekünk, amikor felállítjuk a fiókunkat:

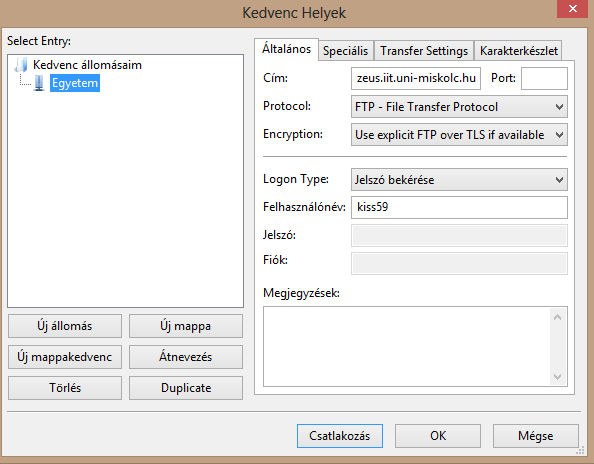
• Az állomásnév (hostname) vagy cím, amelyhez majd kapcsolódunk pl. zeus.iit.uni-miskolc.hu

• A fiókunk felhasználóneve (user name) pl.kiss59

• A fiókunk jelszava (password) pl.jelszo

ftp:(file transfer protocol):Teljesen titkosítatlan formában utaznak az adatok otthoni gépünk és a tárhely között.

sftp:(secure file transfer protocol):Sima ftp titkosított megfelelője.



**A webkiszolgálónk sokkal többre képes egyszerű HTML-fájlok szolgáltatásánál:**

Az index. html nevet annak a fájlnak (röviden indexfájlnak, ahogy általában hivatkoznak rá) adjuk, amelyet alapértelmezett oldalként szeretnénk megjeleníteni a Felhasználók számára, amikor felkeresik a webhelyünk egy adott könyvtárát. a weboldal nem más, mint egy egyszerű szövegfájl, amelynek a tartalmát HTML-kódok "jelölik", meghatározva ezzel a böngésző számára, hogy miként jelenítse meg a szöveget. Ezeknek a szövegfájloknak az elkészítéséhez egy egyszerű szövegszerkesztőre van szükségünk- ez lehet a W indows Jegyzettömbje (Notepad) vagy Macintosh rendszereken a TextEdit. Ne használjuk a WordPadet, a Microsoft Wordöt vagy más, hasonló fejlett szövegszerkesztőt, ezek ugyanis nem olyan egyszerű szövegfájlokat adnak végeredményként, arnilyeneket a webes tartalom megjelenítése megkövetel.

**HTML5 újításai:**

-Videólejátszás gyárilag be van építve a nyelvbe és nem kell külön külső beépülő mint pl.flsah,silverlight….

-Grafika 2D,3D

-Adattárolás

-CSS3

**IP:**

IP az alsóbb rétegektől független(tehát tök mindeggy,hogy vezetéken vagy mikrohullámon jön az internet).

**DNS(Domain Name System):**IP címhez név rendelés.Hierarchikus felépítés.

**Erőforrás azonosítás:**

statikus tartalom:

-szöveg

-html

-kép

-hang

-dokumentum

Dinamikus tartalom:

-szerver szolgáltatás(ssh,ftp,http)

-Generált tartalom(php,JSP,ASP - ezek html filet generálnak)

-Web szolgáltatás(servlet)

<http://localhost> :8080 /Mypage/Index.html

http: protocoll

localhost: DNS cím(kiszolgáló)

8080 port

MyPage/Index.html : Path

**html:**

A keresőprogramok hatékonyságát elősegítve,megadhatjuk a weboldalunk fő nyelvét.Ennek elhagyása nem befolyásolja a megjelenítést,csak a keresőprogramok keresését lassíthatja.

pl <html lang=”hu”>

.

.

</html>

Böngésző hogyan értelmezze a kódot:

<!doctype html>

<html>

<head>

<meta charset=utf-8 />

<title>cím</title>

<link rel=”stylesheet” type=”text/css” media=”screen” href=”css/master.css” />

<script type”text/javascript” src=<http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js>></script>

</head>

<body>

Hello Világ!

</body>

</html>

<!DOCTYPE html> fontos betenni mert bizonyos böngészők nem fogják jól megjeleníteni az oldalunkat.

<meta charset=utf-8 /> azért kell,hogy ne csak az amerikai karaktereket jelenítse meg rendesen,hanem a többit is(Unicode karakterkódolás,aminek implementációi utf-8,utf-16,utf-32).

**html elemek:**

Paragrafus elem:Teljesen kitölti az oldalt(sorra gondolok),más elem abba a sorba ahol ő van nem kerülhet.

<p>Hello Világ</p>

<b>kövér</b>

<u>underline(aláhúzott)</u>

<del>áthúzott szöveg</del>

<i>italic(dőlt)</i>

<br> sortörés

megjegyzés a kódba: <!--<h3>This is a heading</h3>-->

<!-- -->

A megjegyzés nem hajtódik végre,de a kódban látható!

fejlécek:

<h1>This is a heading</h1>  
<h2>This is a heading</h2>  
<h3>This is a heading</h3>

h1-es fejléc a legnagyobb és egyre csökkenő a tendencia.

h6-os méretig lehet fokozni.Tulajdonságuk,hogy a böngésző az elem előtt és után kihagy helyet.

html elemek tulajdonságait az első kacsacsőrben adhatjuk meg!

<a href="http://www.w3schools.com">This is a link</a>

href:kattintáskor hova fog minket vinni

Egy html elemben több tulajdonságot(attribútumot) is meghatározhatunk a kezdő tagban.

<a href="http://www.w3schools.com" target=”\_blank”>This is a link</a>

target=”\_blank”:Egy új lapon fogja megnyitni.

Kinézetük: tulajdonság neve + egyenlőségjel + idézőjelek között a tulajdonság értéke.

A tulajdonságokat egy vagy több szóközzel el kell választani!

lapon belül elemre ugrás:

<a href="#alma">vezerles</a>

<p id="alma">körte</p>

A vezerles linkre kattintva elnavigál minket a körte szöveghez a lapon.A kettőskereszt(#) az id-re mutat.Itt tehát ”#alma” az alma id-jű elemre mutat.

link elem(anchor-horgony):<a> </a>

**html head rész:**Olyan rész ahol nem vizuális elemeket helyezünk el,ezeket a felhasználó az oldalon nem látja.Ezek beállítások és olyan fileokra tudunk hivatkozni,amik a weboldal működéséhez szükségesek.

<head> ben elhelyezhetőek:

* <title> cím
* karakterkódolás <meta charset=utf-8 />
* <base> alapvető címe az oldalunknak.Ha nincsenek megadva címek hivatkozáskor akkor az az alapvető,tehát erre fog mutatni.Default érték linkeknél.
* <link> Kapcsolat létesítés a dokumentumunk és egyéb források között. <link rel=”stylesheet” type=”text/css” media=”screen” href=”css/master.css” />

rel: A rel jellemző jelzi a böngészőnek,hogy a hivatkozott(css) dokumentum milyen viszonyban van a hivatkozó(html) dokumentummal.A megadott érték szerint stíluslap viszonyban van vele.

type:típusa a linknek

media:amire vonatkozik(itt a képernyőre)

href:hivatkozás,honnan töltse be a filet

* <script> javascript nyelven megírt kódok,melyeket hozzárakhatjuk az oldalunkhoz.

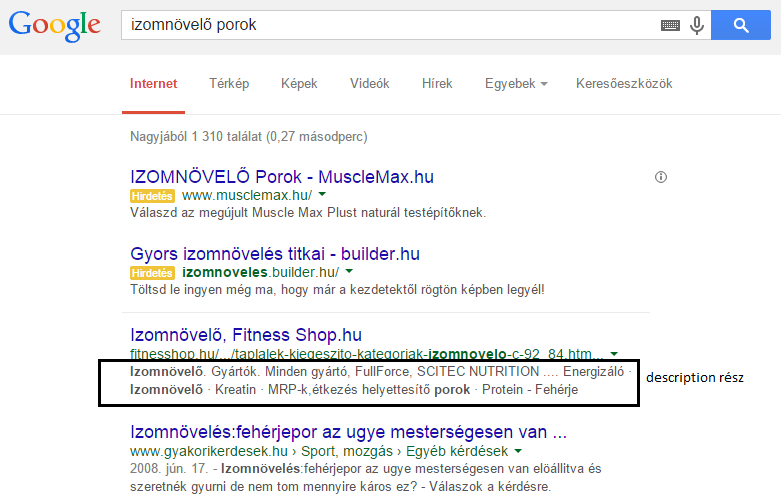
vagy külső forrásból is betölthetjük őket.

<script type=”text/javascript” src=<http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js>></script>

ez például a googlenak az egyik szerveréről töltődik be.

* <meta> Metaadatok.Itt adhatjuk meg a karakterkódolást is.

<meta charset="UTF-8"> (karakterkódolás)  
<meta name="description" content="Free Web tutorials"> (az oldal bővebb leírása)

- <meta name="keywords" content="HTML,CSS,XML,JavaScript"> (az oldal kulcsszavai,keresőmotorok számára(pl.google)mondanak információkat.  
- <meta name="author" content="Kiss Viktor"> (az oldal szerzője)

-<style> <style type="text/css">

p{

color:red;

}

</style>

Ha nincs id rendelve az elemhez akkor p,i, stb… használatával állíthatjuk rá.

Fontos a metaadatok sorrendjének betartása:

<head>

karakterkészlet megadása

weblap címe

külső stíluslap csatolása <link href=”style.css” rel=”stylesheet”>

beágyazott stílus megadása

</head>

A beágyazott stílus a <style> amit akkor érdemes használni,ha egyoldalas vagy különálló a weblap illetve ha egy oldalon a külső stíluslapokhoz képest meg akarjuk változtatni a formázást.

A <style> css kód </style> használatával tehát egy html oldalba egy nem-html kódolás ágyazódik be.

Ugyebár a <meta charset="UTF-8">

sor megadásával elvileg megjeleníthetünk áéőü stb.. karaktereket,de ha ez mégsem történne meg,akkor a fejlesztő környezetben,ami jelen esetbel legyen a Notepad++ ,ott is be kell állítani a dokumentum kódolását.(kódolás->UTF-8 kódolás BOM nélkül).

**File elérési utak:**

relatív:Ahoz a filehoz képest ahol most vagyunk(pl. index.html)

”css/master.css”

abszolút elérési út:nem vesszük figyelembe honnan indulunk

beolvasása kicsit tovább tart,viszont előnye,hogy mindig megtaláljuk.

Fontos:Az index.html nem helyezhető semelyik al-mappába!

al-mappába ugrás: href=”alma/korte/jatek.html”

al-mappából visszalépés az indexhez:href=”../../index.html

továbblépés /

visszalépés ..

példa: E:\html\gyakorlas\css\masikcss és a gyakorlas mappában van az index.html

masikcss mappában lévő jatek.html –ból így ugorhatunk az index.html-re:

<a href=”../../index.html

Bekezdés(<p>):Érdemes minden szöveget bekezdésbe rakni.A bekezdések előtt és után egy sor kimarad.

<hr> :Egy vízszintes elválasztó vonal.

**Listák:**Listákat nem sortörésekkel vagy egyéb maszek megoldásokkal ,hanem html címkékkel hozunk létre.

Fajtái:

-Számozott lista:Fontos a sorrendiség,az elemek előtt számok vagy betűk állhatnak.

<ol>

(ordered list)

az egyes listaelemeket <li> (list item) </li> definiálja

</ol>

Számozott listánál nem feltétlenül muszály 1-ről indulnia a számozásnak: <ol start="5">

-Számozatlan lista:Itt csak felsorolás történik,nem fontos a sorrend.

<ul>

(unordered list)

<li>…….</li>

</ul>

Listák egymásba is ágyazhatóak:

<ol>

<li>alma

<ul>

<li>pálinka</li>

</ul>

</li>

<li>szilva

<ul>

<li>lekvár</li>

<li>bor</li>

</ul>

</li>

</ol>

**Táblázat:**<table> . . . . . .</table>

vízszintes sorokat <tr> . . .</tr> (table row) val egyesével kell felvenni

a cellákban lévő adatokat <td>. . . . </td>(table data)val egyenként adjuk meg

<caption> . . . </caption> egy címet definiál a táblához

<th> . . . </th>(table header) táblázat fejléce

<table>

<caption>Statisztika</caption>

<tr>

<th>idő</th><th>egy</th><th>kettő</th><th>három</th>

</tr>

<tr>

<td>alma</td><td>alm</td><td>al</td><td>a</td>

</tr>

</table>

- a táblázat minden oszlopában azonos számú cellának kell lenni.

- egy adott oszlopban a cellák szélessége azonos

- az üres cellát is fel kell tüntetni <td></td> vagy <th></th> formában

Nagyméretű tábláknál az átláthatóság érdekében használatosak a következők:

thead:Egy táblázatban csak egy lehet,de tartalmazhat egynél több <tr> -t.

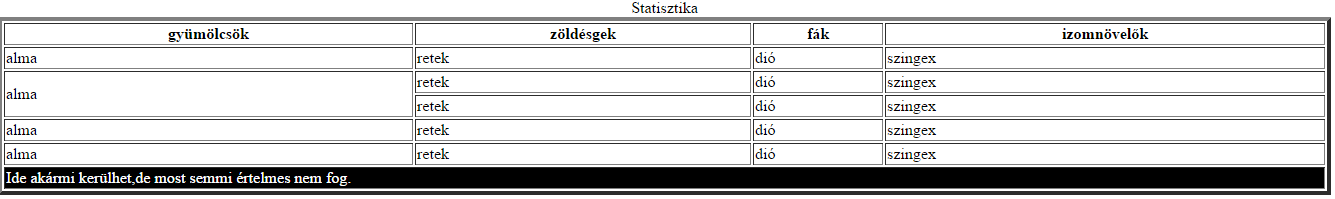
tbody:az hogy hány <tbody> . . . </tbody> párost használunk az nincs korlátozva.

tfoot: Egy táblázatban csak egy lehet,de tartalmazhat egynél több <tr> -t.

border:Tábla szegély vastagságát állítja be.

style="width:100%":A tábla szélességét állítja.

Egyes részeknél az id megadása a css-hez kell.

rowspan:Sorcellaegyesítő hatása van.Egy adat nem egy cellát foglal hanem pl kettőt(rowspan=”2”). colspan:Oszlopcellaegyesítő hatása van.(colspan=”4”)

<table border="4" style="width:100%">

<caption>Statisztika</caption>

<thead id="fej">

<th>gyümölcsök</th><th>zöldésgek</th><th>fák</th><th>izomnövelők</th>

</thead>

<tbody id=test">

<tr>

<td>alma</td><td>retek</td><td>dió</td><td>szingex</td>

</tr>

<tr>

<td rowspan="2">alma</td><td>retek</td><td>dió</td><td>szingex</td>

</tr>

<tr>

<td>retek</td><td>dió</td><td>szingex</td>

</tr>

<tr>

<td>alma</td><td>retek</td><td>dió</td><td>szingex</td>

</tr>

<tr>

<td>alma</td><td>retek</td><td>dió</td><td>szingex</td>

</tr>

</tbody>

<tfoot id="lab">

<tr>

<td colspan="4">

Ide akármi kerülhet,de most semmi értelmes nem fog.

</td>

</tr>

</tfoot>

</table>

Az ehez tartozó css:

#lab{

color:white;

background-color:black;

}

Egy táblázat celláinak tartalma lehet újabb táblázat vagy lista is!

**Képek:**Az <img> címke egy állóképet helyez el a weboldalon.Ennek kötelező attribútuma az src ,ami a kép elérési útvonalát adja meg.Ha a kép a html dokumentummal közös mappában van tárolva,akkor elég csak a kép nevét megadni az src után.

<img src=”filename.kiterjesztés”> ez egy páratlan címke(tag).

A böngészők a jpeg,png és gif kiterjesztésű képeket képesek kezelni.

A képek sorban elhelyezhetők,sorban az előtte lévőt követi.

Megadható attribútumok:

-height:magasság megadása

-width:szélesség megadása

Ezek nem kötelező jellemzők,de segítenek a böngészőnek már a kép letöltődése előtt elrendezni a megjelenítést,azonnal el tudják helyezni a szöveget.

-alt:Képet helyettesítő szöveg.Csak akkor jelenik meg,ha a kép nem betölthető.

-title:Olyan szöveget definiál,ami akkor írodik ki,ha a kép fölé visszük az egeret,ekkor egy kis szövegdobozban láthatjuk.

<img src="hatter.jpg" alt="csinos" width="400px" height="400px" title="Anjelica">

Fontos:Ha a képnek semmi más funkciója nincs a díszítésen kívül,akkor azt nem képként,hanem HÁTTÉRKÉPKÉNT kell megadni!

**Mozgókép és hang:**A HTML5 szabványa lehetőséget nyújt közvetlenül a böngészőben való lejátszásra,azaz nincs szükség bővítményekre mint például a Flashre,Silverlightra stb..

<audio controls>

<source src="d.ogg" type="audio/ogg">

<source src="d.mp3" type="audio/mpeg">

Your browser does not support the audio element.

</audio>

**Beágyazott böngészőtartalom:**Másik weblapról böngészési lehetőséget illeszt be.

Az állóképhez hasonlóan ez is sorfolytonos ,illetve ha nem fér be a képbe,akkor görgetősáv jelenik meg.

<iframe src="http://index.hu" width="35%" height="40%"></iframe>

Mint látható src-hez a teljes cím szükséges ,nem elég [www.-val](http://www.-val) kezdeni.

Ezzel a módszerrel ágyazható be youtube video is.

**Karakter entitások:**Ezekre a foglalt karakterek és a billentyűzetről nehezen bekérhető karakterek megjelenítése végett van szükség.

Foglalt karakterek:

<

>

„

&

Nehezen bekérhető karakterek:Görög betűk,pénznemek,matematikai szímbólumok.

A foglalt és különleges karakterek két féle képpen is megadhatók:

-entitás számmal

-entitás névvel

|  |  |  |
| --- | --- | --- |
| karakter | entitás szám | entitás név |
| < | &#60; | &lt; |
| > | &#62; | &gt; |
| & | &#39; | &amp; |
| ” | &#34; | &quot; |
| # | &#35; | &num; |
|  |  |  |
|  |  |  |

<http://dev.w3.org/html5/html-author/charref>

ezen a weboldalon megtalálható az összes szimbólum listája.

Nem törhető szóköz:Előfordulhat olyan eset hogy nem szeretnénk külön látni például a coca cola,Hong Kong stb. neveket.

Megoldás: &nbsp

Kiss&nbspViktor

Bizonyos karakterek megjelenítéséhez HTML címkékre van szükség:

-alsóindex: <p>H<sub>2</sub>O</p> eredmeny: H2O

-felsőindex: <p>5cm<sup>3</sup></p> eredmeny: 5cm3

-idézet:<q>Lenni vagy nem lenni,ez itt a kérdés?</q> eredmeny:a szöveget idézőjelek közé teszi

-hosszabb blokkszerű idézetekhez: <blockquote> </blocquote>

Előre formázott szöveg : Lényege,hogy megtartja a formázást,és nem húzza össze a dolgokat sorfolytonosra.

<pre>

Az alma finom,

de ha romlott,

akkor ne edd meg!

</pre>

Ezt ugyan így fogja megjeleníteni a böngésző!

Az alma finom,

de ha romlott,

akkor ne edd meg!

**A weboldal egyes részeinek formázását elősegítő HTML címkék:**Ha a dokumentum egy részénél nincs olyan címke amihez hivatkozást rendelhetnénk(pl.css),akkor mesterségesen bevihető ilyen címke.A <div> . . . . . </div> páros címke ezért a felelős.A benne elhelyeszkedő tartalomra(kép,szöveg,lista,táblázat,hivatkozás) egységesen lehet hivatkozni és formázást hozzárendelni.Ha több div szakasz van a weboldalon ,akkor érdemes hozzájuk id-t rendelni.

Az id értékének egyedinek kell lenni!

Az id-knek betűvel kell kezdődnie,kis-nagybetű érzékeny és nem tartalmazhat szóközt.

Egy blokkszintű elemet semmi értelme div blokkba rakni,ott is adható id.

<table id=”alma”>

..

</table>

Figure:a <figure> tárolóelembe foghatók a képek,listák,diagrammok-ábrák és a hozzájuk tartozó címek,és ezzel lehet egy közös elemként formázni ,kezelni.

<figure>

<figcaption>A kép címe</ figcaption>

<img src=”alma.jpg”>

</figure>

(ha a cím felül van)

Ha a cím alul van:

<figure>

<img src=”alma.jpg”>

<figcaption>A kép címe</ figcaption>

</figure>

Soron belül:Soron belül a <span> . . . </span> páros címkével lehet kijelölni sorfolytonos(inline) tartalmat,ami külön formázható.Természetesen ehez is lehet id-t rendelni.

**Hivatkozások:**Webhelyen belül egy másik webhelyre történő ugrás,illetve küldő erőforrások elérése.

<a> . . . </a>(anchor – horgony)

<a href=”teljes url cím”>oldal neve,akármilyen szöveg,vagy kép</a>

oldalon belüli hivatkozás:<a id=”..”></a>

Azonos webhelyen belül másik weblapra ugrani (ha ugyan abban a mappában vannak) a következő módon lehet:

<a href=”jatek.html”>Játék</a>

Egy oldalon belül kiválasztott részhez a következő módon ugorhatunk:

<a href=”jatek.html#idnév>Játék</a>

pl:<a href=”jatek.html#alma”>jatek<a/>

Persze ennek akkor van értelme,ha egy oldalnál hosszabb a hivatkozott tartalom,és ott azonnal ugrani szeretnénk valahova.

Oldalon belül is lehet ugrálni:

<a id="eleje"></a>

.

.

<a href="#eleje">az oldal elejére</a>

**Képek használata hivatkozásokhoz:**Ha a képre kattintunk navigál minket tovább,nem feltétlenül fontos szövegnek lennie.

<a href="jatek.html">

<img src="hatter.jpg" alt="csinos" width="400px" height="400px" title="Anjelica">

</a>

**Hivatkozás használata letöltéshez:**download jellemzővel kiegészítve

<a href="hatter.jpg" download>kattints a letöltéshez</a>

A download jellemző csak href jellemzővel együtt használható!

**Képtérképek:**A hivatkozások speciális alkalmazása,amikor a kép egyes részeihez(hotspot) rendelhetünk hivatkozást.

Hivatkozási területként három alakzat (ami a legjobban illeszkedik a kijelölendő képrészletre)helyezhető a kép egyes részeire.Kör,derékszögű négyszög,általános háromszög.

A hivatkozási terület kódja az <area> páratlan címke.Ennek kötelező attribútumai a shape(alak) és a coords(koordináták).

Kör: A középpontjával és a sugarával lehet definiálni.Ennek tudatában nyilvánvaló,hogy három számra lesz szükség.A középpont x,y koordinátái pixelben(a kép bal felső sarkától) megadva.Illetve a kör sugara pixelben megadva.

<map id="ok">

<area shape="circle" coords="20,20,202" href="jatek.html" alt="alma" title="kep">

</map>

<img src="hatter.jpg" alt="csinos" title="Anjelica" width="400" height="400" usemap="#ok">

**Űrlap:**A webhely látogatóival való kommunikációt teszik lehetővé.Űrlapok kialakítása HTML-vel,formázása css-vel ,működtetése script-nyelvekkel történik.

Minden űrlapnak a <form name=”…”> -vel kell kezdődnie,és </form>-val kell zárulnia.

<form>. . . . </form>

-input:Egysoros szöveg bevitelére alkalmas.Ha megtelik a doboz,akkor tolódik a szöveg.

<form name=”urlap”>

<input type="text" name="szov">

</form>

type lehet:

-"search"

-"tel"

-"email"

-"url"

-"password"

A jelszó különbözik a többitől,hisz ott pontokat jelenít meg a böngésző,biztonsági okok miatt.

Az űrlapon megadott adatok név-érték páronként értékelhetők ki,ezért mind az űrlapnak(form),mind az adatbevitelt,mind a választást lehetővé tevő űrlapelemnek name jellemzővel (a választógombok kivételével)egyedi értéket(egyedi nevet)kell tartalmaznia.

Attribútumok:

-value=”…” :A megadott kezdőérték jelenik meg a szövegmezőben.

-size=”…” :A szövegdoboz karakterhosszát adhatjuk meg.Ez a maximálisan beírható karaktesrzámot nem befolyásolja.

-maxlength=”….” :A felhasználó számára maximálisan bevigető karakterszámot határozza meg.

-readonly :Ennek a jellemzőnek nem kell értéket adni.A felhasználó nem tudja módosítani az esetleges kezdőértéket,beleírni sem tud.

-hidden :Ennek sem kell értéket adni.Ezt még látni sem láthatja a felhasználó(az értéket nem láthatja)

-autofocus :Ennek sem kell értéket adni.Az űrlap betöltődésekor a kurzor autómatikusan a szövegmezőbe ugrik.

-required :kötelezően kitöltendő űrlapelem

<form>

<p><input type="search" name="szov"></p>

<p><input type="tel" name="szovi"></p>

<p><input type="email" name="szoveg"></p>

<p><input type="url" name="szoveg" autofocus></p>

<p><input type="password" name="szoveges" size=”95”></p>

</form>

Szövegmezők feliratozása:

<form>

<label>jelszó

<input type="email" name="szovw">

</label>

</form>

A jelszó szöveg szabadon elhelyezhető a szövegdoboz előtt,mögött vagy felett.

Választóelemek kiválasztása:Megadott választási lehetőség bejelölhető vagy bejelölhetőek.

<label><input type="checkbox" name="nyelv1" value="Magyar" checked>magyar</label>

<label><input type="checkbox" name="nyelv2" value="Angol">angol</label>

<label><input type="checkbox" name="nyelv3" value="Nemet">német</label>

<label><input type="checkbox" name="nyelv4" value="Belga">belga</label>

Ezek inline elemek,tehát egymás mellé rakja őket.

checked attribútum:Alapból bepipáltra helyez.

C:\Users\Kiss Viktor\Desktop\golyo.PNG

type=”radio” :Itt csak egy lehetőség választható.(Választógomb)

<label><input type="radio" name="talan" value="igen">Igen</label>

<label><input type="radio" name="talan" value="nem">Nem</label>

<label><input type="radio" name="talan" value="talan">Talán</label>

Az egy kérdésre vonatkozó választógomboknak ugyan azzal a name értékkel kell rendelkeznie,hogy csak egy választható legyen közülük.C:\Users\Kiss Viktor\Desktop\csakegy.PNG

A checked attribútum megadása alapértelmezett bejelölést biztosít.

Időpont megadására alkalmas űrlapelemek:

type=”date”

type=”month”

type=”week”

type=”time”

Mennyiség megadására alkalmas űrlapelemek:

type=”number”

type=”range”

<label><input type="number" name="szam" value="4" min="-3" max="400" step="3">Szám</label>

C:\Users\Kiss Viktor\Desktop\fas.PNG

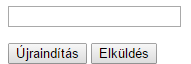
step azt jelenti hogy a nyílra kattintva 3-val módosul a szám értéke

A range is hasonló csak a type más,és ez egy csúszka.

Küldés gombok és az alaphelyzet:A reset-vel a kezdőállapot állítható vissza, a submit-vel a kitöltött űrlap elküldése történik.

<input type="reset" name="res" value="Újraindítás">

<input type="submit" name="send" value="Elküldés">



Fájlfeltöltés : Fájlokat(multimédiákat is) így küldünk

<input type="file" name="filefeltöltes">

Színválasztás:

<input type="color" name="szin">

**Legördülő lista:**<datalist>. . . . </datalist> páros elemmel.

Ennek elemeit az <option value=”…”> sorokkal lehet feltölteni.

A legördülő listát az <input type=”text” > szövegmezőhöz a list=” . .”-vel rendelhető hozzá.

<datalist id="nemzetiseg">

<option value="magyar">

<option value="angol">

<option value="német">

<option value="belga">

</datalist>

<form>

<p><input type=”text" name="szov" list="nemzetiseg"></p>

</form>

Látható,hogy a text típushoz hozzárendeltük a nemzetiseg id-jű legördülő listát.

**Selectes legördülő lista:**<select>. . . . </select> párossal lehet.

A lista elemei az <option>. . . </option> párossal vehetőek fel és azok értékei value= után kerülnek felvételre.

attribútumok:

-size=”3” :A listának hány eleme gördüljön le.

-selected :Nincs értéke.Kezdőérték beállításra való.

-multiple : Egy időben több érték is választható

<label>Nyelvtudás

<select name="skill" multiple>

<optgroup label="Többen beszélik">

<option value="angol">angol</option>

<option value="német">német</option>

<option value="francia" >francia</option>

<optgroup label="Kevesebben beszélik">

<option value="belga" selected >belga</option>

<option value="spanyol">spanyol</option>

</label>

Az <optgroup> csoportosítás.

**Űrlap elemeinek a csoportba foglalása:**

<fieldset>. . . </fieldset> párossal lehetséges,és ha nevet akarunk adni neki akkor a <legend>. . . </legend> párossal megtehetjük.

<fieldset>

<legend>Bejelentkezés</legend>

<form>

<p><label>Név<input type="text" name="szov"></label></p>

<p><label>Jelszó<input type="password" name="pw"></label></p>

<input type="submit" name="küld">

<form>

</fieldset>



**Többsoros szövegmező:**<textarea>. . . </textarea> párossal.

<p><textarea name="sok" cols="10" rows="50" placeholder=”röviden foglald össze” maxlength=”150”></textarea></p>

rows a sorok számát,cols az oszlopok számát adja meg.

placeholder:Kezdőszöveg berakása.

maxlength:maximális karakterszám.

**HTML5:**A HTML5 visszafelé kompatiblis,tehát ami működött html korábbi verzióiban,az működik html5 alatt is.

**CSS**

Melyik stilus fog ervenyesulni, ha tobb stilust is definialunk ugyanahhoz a HTML elemhez?

A kovetkező negy beallitas ervenyesul egyre nagyobb prioritassal (tehat utkozes eseten a kesőbbi felulirja az előzőt).

1. a bongesző alapbeallitasa

2. kulső stiluslap

3. head elemben definialt stilus

4. soron beluli stilus

Tehat a soron beluli stilus a legmagasabb prioritasu, es felulir minden alacsonyabb szintű formazast.

Ezen kívül fontos még figyelembe venni egy általában jól használható ökölszabályt: az a stílusdefiníció fog érvényesülni, amelyik a legszűkebb hatókörrel rendelkezik. Pl. ha minden bekezdést barna színűre állítunk, de egy egyedi azonosítóval rendelkező bekezdést zöldre, akkor az utóbbi szabály rendelkezik szűkebb hatókörrel. Így ennél a bekezdésnél zöld lesz a szín.

A css kódolás lehet külön stíluslap,vagy más webhelyről betöltődő vagy dokumentum a fejrészben vagy egy html címkén belül(inline style).

a.)külső stíluslap:

<link rel=”stylesheet” href=”style.css” type=”text/css”>

b.)külső stíluslapok más webhelyről:

<style>

@import url(<http://www>. )

</style>

c.)Egyoldalas weboldal,vagy kis módosításokat igénylő(a többihez képest) akkor :

<style>

css formázás kódolása

</style>

A belső stílus felülírja a külső stílust.

d.)Speciális eset:

<p style=”css kód”>Bekezdés szövege</p>

PL. <p style="color:blue">alma</p>

**Média jellemző:**Kimeneti eszközre vonatkozó feltételek leírására használatos.

Típusok:

-all :Alapértelmezett.Minden médiában azonos lesz a megjelenítés.

-screen(képernyő) :Különböző méretű képernyőkre.

-print(nyomtatás) :Az all vagy a screen alá kerül a kódban,és elég csak az azokban képesti eltérést tartalmaznia.

A handled és a projection kikerült a szabványból!

Példa médiafüggő (400px és 700 px közötti szélességű) képernyő css definiálására.

@import url(<http://www>.) screen and (min-width:400px) and (max-width:700px);

vagy

@media screen and (min-width:400px) and (max-width:700px){

..

..

..

}

**Kijelölők:**A html kódban egy részre szeretnénk csak formázást kialakítani.

kijelölö{meghatározások}

Egy kijelölőhöz több meghatározás is tartozhat és több kijelölőre is érvényesíthetők ugyanazok a meghatározások.

A kijelölők lehetnek:

* elemi kijelölő:A weblap összes azonos elemére vonatkozik.Például p ,akkor az összes bekezdést formázzuk. p {meghatározások}
* csoport kijelölő:Több elemhez ugyan azt a stílust rendeli.A kijelölt elemek vesszővel és üres betűközzel választandóak el.

h1, h2{meghatározások}

Fontos:Ha a csoporton belül az egyik kijelölő érvénytelen,akkor az egész érvénytelen lesz.

* azonosító kijelölő:A html dokumentumban id-vel ellátottakra vonatkozik.

<p id=”hagyma”>balababla</p>

ekkor a css-ben : #hagyma{meghatározások}

* származtatott kijelölő:Egy id jellemzővel definiált szakaszon belül elemkijelölést végez.

<div id="elso">

<h1>alalalala</h1>

<p>alsjalksfjl</p>

<p>alsjalksfjl</p>

</div>

css:

#elso p {

color:red;

}

Tehát piros színűre csak a bekezdéseket(p) állítja,a h1-et nem.

* osztály kijelölők:általunk osztályokba sorolt elemekre megjelenítést határozhatunk meg.

<h1 class="egy">alalalala</h1>

<p class="ketto">alma</p>

<p class="egy">körte</p>

<p class="egy">szilva</p>

<p class="ketto">barack</p>

<p class="egy">dinnye</p>

css:

p.egy{

color:blue;

}

p.ketto{

color:red;

}

h1.egy{

color:green;

}

ha csak simán .egy{meghatározások}lenne akkor az összes egy classosra érvényes lenne beleértve a h1,stb….

Természetesen a kijelölő típusok kombinálhatók egymással:

#header p.stilus1{meghatározások}

**Az ál-osztály kijelölő öt csoportja:**

* link típusú ál-osztályok:A linkekre is szabhatunk stílust.A linkeknek két kategóriája van,az egyik ami már meglátogatott ,a másik pedig még nem volt meglátogatva.

<a href="http://www.sporthirado.hu">sh</a>

css:

a:link{

color:green;

}(még nem látogatottuk meg)

a:visited{

color:orange;

}(meglátogatott)

* dinamikus ál-osztály:

:hover :az egérrel rámutatás egy dobozra.Itt nincs kattintás.

a:hover{

color:pink;

}

:active :A kurzor az objektum felett van és lenyomott az egérgomb.Amíg lenyomva tartjuk,addig tart a hatás.

.egy:active{

color:white;

}

Ez működik akár p tagre is!!!!!!!

:focus :Egy elem által létrehozott doboz fókuszba kerül egy kurzor,vagy billentyüzet esemény következik be.

:focus{

color:red;

}

Természetesen ezek pontosíthatók is,hisz ha nem az összes linkre akarok alkalmazni hasonlókat,akkor lehet szűkítést végezni:

<a href="http://www.sporthirado.hu" class="nav">sh</a>

css:

a.nav:hover{

color:pink;

}

Linkekre kötött a sorrend:

a:link a:visited a:hover a:active

* target típusú ál-osztály: ugrásoknál az ugrási célpont megjelenítése így fog történni.

:target {

border: 2px solid #D4D4D4;

background-color: #e5eecc;

}

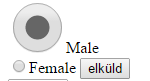
<a href="#alma">vezerles</a> ha ide ugrunk akkor az eredménye a css-ben megfogalmazott lesz.Például ha alma szövegről a „Béla” szövegre ugrok,akkor az ugrás eredményeként Béla be lesz keretezve.

* :checked {

height: 50px;

width: 50px;

}

Ez a jelölőnégyzetek és választógombok formázása.

* Nyelv szerinti kiválasztás :lang

**Pszeudo elemek:**Két csoportba sorolhatóak:

* első sor vagy első sor első elemére való formázás.

minden bekezdés első sorára: p::first-line{meghatározás}

minden bekezdés első betűjére: p::first-letter{meghatározás}

* Forrásdokumentumban nem szereplő tartalmat lehet beilleszteni ::before és az ::after kijelölőkkel.Például ha minden bekezdés elé ugyanazt az elemet illesztjük:

p::before{meghatározás}

vagy utána:

p::after{meghatározás}

**Jellemző kijelölő:**Vagy csak jellemző,vagy jellemző és meghatározott érték alapján.

<p class="ketto">alma</p>

<p class="egy" name="ok">körte</p>

<p class="egy">szilva</p>

<p class="ketto">barack</p>

<p class="egy">dinnye</p>

Látható ez az öt bekezdés.De én csak azt a bekezdést akarom formázni,amelyiknek van name attribútuma és annak az értéke „ok”

css:

p[name="ok"]::first-letter{

color:green;

}

Ha azt szeretnénk,hogy azokat formázza ami csak rendelkezik a tulajdonsággal,de az értéke nem számít,akkor

p[name]::first-letter{

color:green;

}

Természetesen a name attribútum helyén szerepelhet tile,stb…

**Strukturális ál-osztályok:**

-:root a dokumentum gyökerét képviseli.

css:

:root{

color:red;

} ekkor mindent pirossal ír ki.

-:nth-child(an + b)

nth-child(0)-nál nincs semmi kijelölve.

nth-child(1) a szülő első gyerekét kiválasztja pl:

<div id="elso">

<h1>alalalala</h1>

<p>alsjal<i>alma</i>ksfjl</p>

<p>alsjalksfjl</p>

</div>

css:

body > div :nth-child(1){

color:red;

}

Ekkor a body(a dokumentum) div-elemei közül azoknak az első elemét formázza.

nth-child(2n) Minden második gyereket kijelöl.

nth-child(10n - 1) -1 től indul tizesével tehát amilyen indexű gyerekeket kijelöl:9,19,29,…..

nth-child(10n + 9) ez ugyan azt jelenti mine a felette lévő.

-:nth-last-child() az utolsó elemtől.

:nth-last-child(-n + 2) az utolsó két elem kijelölése.

:nth-last-child(n + 2) az utolsó elem kivételével mindent kijelöl

Az első és az utolsó elem kivételével mindent kijelöl:

body > div :nth-child(n + 2):nth-last-child(n+2){

color:red;

}

vagy ez is megoldás:

body > div :not(:first-child):not(:last-child){

color:red;

}

Megjegyzés:first/last-child az első vagy utolsó gyereket jelöli ki,tehát ekvivalens az nth-child(1)-vel a first.

-:nth-of-type(2n+1) : Az azonos típusúak közül jelöl ki.

img:nth-of-type(2n)

**Kombinátorok:**Öröklési viszonyok általi kijelölést tesznek lehetővé.

-Származtatott kombinátor:Egy elem a másik elem leszármazottja.

Például formázás:

<p>alsjal<i>ksk<b>ksk</b>sk</i>ksfjl</p>

p i kombinátorral jelölhető ki.

A div \* p olyan bekezdéseket jelöl ki,amik div unokái vagy még mélyebb rokonai(ükönoka).

href jellemzőjü elemeket jelöl ki,melyek egy bekezdésben vannak melyek egy div tárolóhoz tartoznak:

div p \* [href]

-Gyermek kombinátor: p bekezdés gyermeke a body-nak,akkor

body > p

Olyan i elem kiválasztása,mely leszármazottja egy p-nek,ami gyereke egy divnek:

div > p i

-Testvér kombinátor:Két elem melyeknek közös az őse,de egymásnak nem leszármazottjai.

i elemet közvetlenül követő b elem kijelölése: i + b

html-ben: <p>alsjal<i>ksksk</i><b>ksk</b>ksfjl</p>

általános testvér az aki nem közvetlenül követi a másikat:

jelölése i ~ b

h1 ~ a{

color:red;

}

h1 után a legközelebbi anchor,amivel egy ősük van.

Fontos:Idézőjelek közé csak azt tehetjük ami nem egy szóból áll.A kötőjellel elválasztott(ál-osztály)kifejezések egy szónak számítanak.

div > p{

font-family:"Times New Roman";

}

/\*megjegyzés\*/

**Több stíluslap ütközésekor a rangsor:**

öröklés:Ha egy elem egy másik elembe ágyazódik bele,és a beágyazottnak nincs stílus megadva,de a beágyazónak igen,akkor a beágyazottra is érvényes a stílus.Ha a beágyazottra van definiálva stílus akkor az érvényes amit rá szabtak.

szűkítés:A pontosabb kijelölők felülbírálják az általánosabb kijelölőket.Az azonosítókijelölőknek(id #)nagyobb a súlyuk mint az elemkijelölőknek(p,div).

**Szinek definiálása:**Több féle képpen történhet.

color:red;

color:rgb(255,0,0);

color:#ff0000;

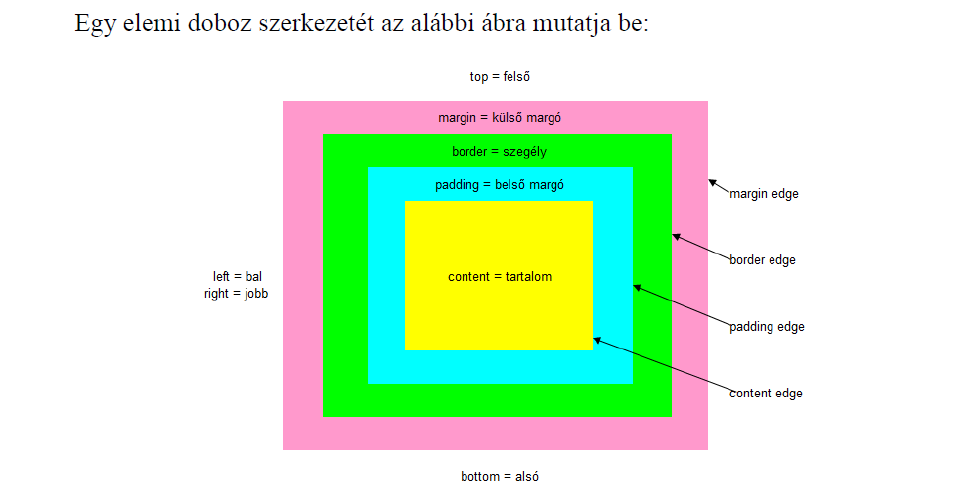
background-color:háttérszín definíció;

**Méretek:**

width(szélesség)

height(magasság)

min/max-width/height



belső margó állítása:padding(alsó,felső,jobboldali,baloldali egyben állítása)

paddig-top/bottom/right/left

külső margó állítása:margin(az összesre ua.mint padding)

margin-left/top/bottom/right

Alapértelmezett értékük mindnek nulla.

**Szöveg formázás:**

Szöveg behúzása:

h1{

text-indent:3em;

}

p{

text-align:left vagy right vagy center(jobbra középre vagy balra igazítás)

}

justify(sorkizárt)értéket is felvehet,ez néha egyenetlen szövegoszlopot eredményezhet.

szoveg díszítés:

text-decoration-color:

text-decoration-line:none,underline,overline,line-through

text-decoration-style:solid(folytonos),wavy(hullámos),dashed(szaggatott),double

text-decoration-skip: vonal kihagyjon-e valamit?

értékei a skipesnek:none,images(képek kihagyása az aláhúzásból),spaces(elválasztók kihagyása)

kisbetű/NAGYBETŰ átalakítás: text-transform:

lehetséges értékei:

none(alapértelmezett)

uppercase(nagybetűsre)

lowercase(kisbetűsre)

capitalize(első betű nagy a többi kicsi)

Betűköz: letter-spacing:0.1em;

szóköz: word-spacing:0.2em;

betűméret: font-size:120px;

szöveg-árnyék:

text-shadow: 5px 5px 10px blue,5px 10px 10px red;

ezen példán két árnyékot is ráeresztetteünk a szövegre.

első px:pozitív értéknél a szövegtől jobbra,negatívnál ballra tolódik az árnyék

második px:pozitívnál lefelé,negatívnál felfelé tolódik

harmadik px:az árnyék elmosódása.Ha egy szöveg akkor az egyre elmosódottabb minnél nagyobb a px.

negyedi szín:Szinét határozza meg az árnyéknak.

Megjegyzés:Egy szöveghez akár 7 árnyék is rendelhető.

Függőleges igazítás: :vertical-align

p #egye{

vertical-align:sub;

}

p #igya{

vertical-align:sub;

}

lehetséges értékek:top,middle,bottom,super,sub,central,baseline

eredmény:

C:\Users\Kiss Viktor\Desktop\dfgdfgd.PNG

Szöveg,kép és beviteli mező egymáshoz illesztéséhez is a vertical-align használatos.

**Listák formázása:**

list-style-type: a felsorolásjelek típusát adja meg.

none(nincs jel),decimal(1-től kezdve 10-es számrendszerben),decimal-leading-zero(01,02,03…..99),lower-roman(római számok kicsi),upper-roman,georgian,disc(kitöltött kör),circle(üres kör),square(kitöltött négyzet),lower-greek(görög kisbetűk),

Felsorolásjel lehet egy kép is(értelemszerűen nem sorszámozott listáknál használatos):

list-style-image:

ul {

list-style-image:url(hatter.jpg);

}

Ha a kép a gyűjtőkönyvtárunkban van.

Lista soraihoz is rendelhető formázás:

ol li{

padding-left:25px;

}

**Szögletes szegélyek:**

h1 {

border-top-style:double;

border-left-style:double;

border-right-style:double;

border-bottom-style:double;

}

Beállítható külön az alja ,teteje stb.

Milyen lehet a szegély?

none(nincs,ez az alapértelmezett)

hidden

dotted(pontozott vonal)

dashed(szagatott vonal)

solid(folytonos vonal)

double

groove(bemélyített vonal)

ridge(kidomborodó vonal)

inset(süllyesztett)

outset(kiemelkedő)

szín beállítása: border-bottom-color:blue;

ugyan úgy lehet mindre külön állítani.

border-bottom-color:transparent; transparent(átlátszó)

Szegély vastagságának megadása: border-width:25px;

ezzel mind a 4 oldalra beállítjuk.

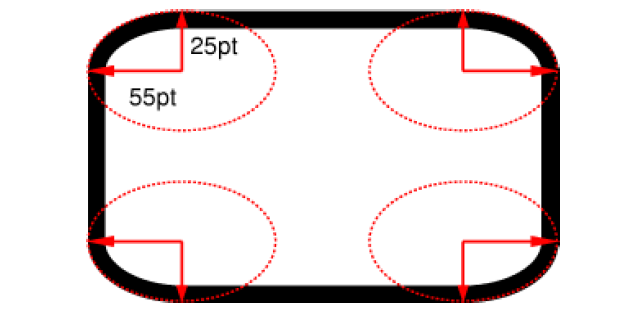
**Szegély lekerekítés:**

border-top-left-radius:25pt 25pt;(bal felső sarok)

border-top-right-radius:25pt 25pt;

border-bottom-left-radius:25pt 25pt;

border-bottom-right-radius:25pt 25pt;



az első pt a vízszintes sugárméret,a másik a függőleges!

Egyszerű submit nyomógomb lekerekítés:

input[type="submit"]{

background-color:red;

border-top-left-radius:25pt 25pt;

border-top-right-radius:25pt 25pt;

border-bottom-left-radius:25pt 25pt;

border-bottom-right-radius:25pt 25pt;

}

**Blokkszintű elemek elhelyezése:**

position: utnán írandók

-static(alapértelmezett)

-relative(top,bottom,left,right paramétereknek érték adható,amikkel megadjuk,hogy az eredeti helyéhez képest mennyivel toljuk.(alapértelmezett helyéhez)

#mamam{

background-color:yellow;

width:50px;

height:100px;

position:relative;

top:-15px;

}

-absolute(itt is top,bottom,left,right paramétereknek érték adható,de itt nem a saját pozíciójához képest tologatjuk,hanem egy másik elem vagy a kezdeti tárolótömb(body)

-fixed (ennél a képernyő görgetésekor ez is mozog,a képernyő a fix viszonyítási pont)

Láthatóság: visibility:hidden;

az elem ott marad ahol volt,csak láthatalna lesz,a heléyre ugyebár semmi nem kerülhet.

**Láthatósági sorrend:**Az egymást takaró elemekre be lehet állítani egy sorrendet.Alapértelmezetten ha ütközés van,akkor az takarja a többit,amelyik a kódban leghátrébb van.

A láthatósági sorrendet a z-index tulajdonsággal változtathatjuk meg.

#mam{

background-color:blue;

width:120px;

height:30px;

z-index:2;

}

amelyik objektumnak a legnagyobb a z-index értéke az lesz legfelül és így haladunk egészen lefelé a hierarchiában.A z-indexelés indulhat mínusz számról is,ebben nincs kikötés ,hogy nulláról kellene.

**Doboz hierarchia:**Lehet:

-blokkszintű(bekezdés,lista,táblázat,div szakasz)

-sor(line box)

-inline(sorfolytonos szöveg,kép stb..)

display: megszabhatja az egyes elemekhez a doboz tulajdonságokat.

ennek gyakran használt értékei:

inline:folytonos elrendezésü,ez az alapértelmezett

none:az elemhez nem rendel dobozt,azaz az elemnek nincs befolyásoló hatása a weboldal elrendezésére.A visibility:hidden az hatással van,hisz ott van csak láthatatlan.

block:blokk doboz

Lista blokkszintűből inline képzése:

li{

display:inline;

}

eredmény: egymás mellé írja a pontokat:D

img{display:block;}

A képet ami inline(sorfolytonos)blokkossá alakítja,azaz vele egy sorban más nem lehet,és tartalmazza a külső és belső szegélyt….

**Doboz árnyékok:**

box-shadow:5px 5px 5px blue;

első px:pozitívnál vízszintesen jobbra,negatívnál balra

második px:pozitív le,negatív fel

harmadik px:árnyék szélének az életlenedése(0-nál a legélesebb,és mínusz nem lehet)

box-shadow:5px 5px 5px blue inset;

az inset kulcsszó változtatja belső árnyékká.

**Háttérképek:**A weblap valamennyi elemének(karakter,szó,lista,táblázat) van egy hátsó rétege,amely alapértelmezetten átlátszó,vagy színe van(background-color),vagy egy kép(background-image).

Háttérképet állítunk a body-ra,azaz az egész dokumentumra.Ez egy kis gif(50x50-es),de folymatos kitöltés van,a bal felső saroktól indulva kitölti az egész teret.

body{

background-image:url(bg.jpg);

}

A background-repeat: tulajdonsággal szabályozható,hogy milyen módon töltse ki a teret a háttér.

Értékei lehetnek:

-repeat(alapértelmezett)

-repeat-x(a háttérkép csak vizszintes ismétlése alkalmazza)

-repeat-y(a háttérkép csak függőleges ismétlése alkalmazza)

-no-repeat(a háttérkép csak egyszer lesz megismételve)

-round(egészszámú többszöröse a képnek ki kell töltse a teret,ha nem akkor átméretez)

-space(az első és az utolsó kép a két sarokhoz igazodik)

Háttérkép pocíciója:

background-position:100px 120px;

Háttérkép görgethetősége:

background-attachment:

-scroll(alapértelmezett,ha bővül a terjedelem ,akkor a háttér követi őt)

-local(a háttér egy elemhez rögzített,így ha az nem fér bele az oldalra,akkor követi őt a háttér)

-fixed(görgetástől függetlenül állandó a helyzete)

**Túlnyúlás:**A tartalom túlnyúlik a befoglaló dobozon.Blokkos és soron belüli blokkos elemeknél alkalmazható.

overflow:visible,hidden,scroll,auto,no-display,no-content

visible(akár kilóg akár nem ,akkor is megjelenik a tartalom)

hidden(a dobozon kívüleső részek nem látszanak)

scroll(görgető mechanizmus jelenik meg,mellyel a tartalom görgethető)

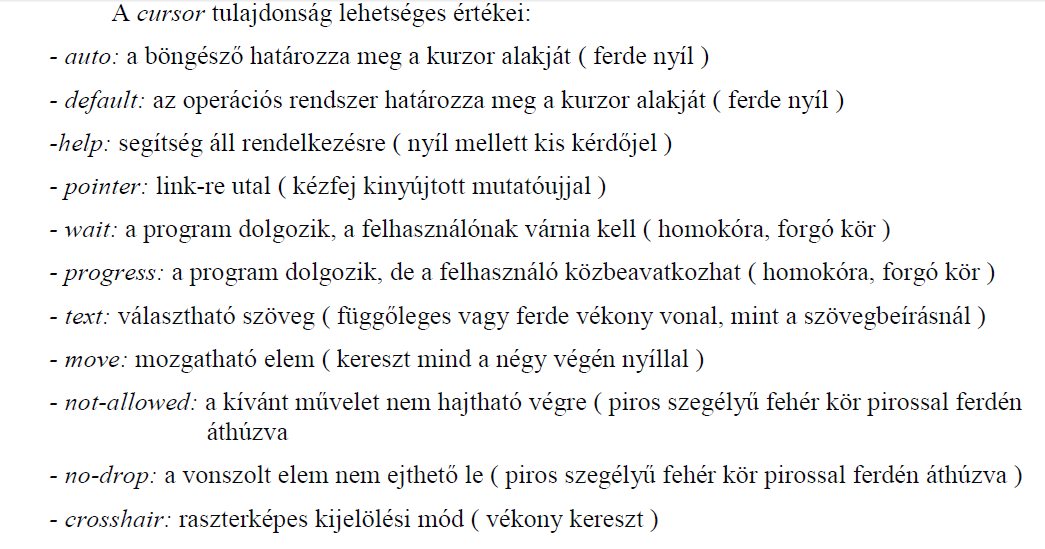
auto(görgető mechanizmus ez is)

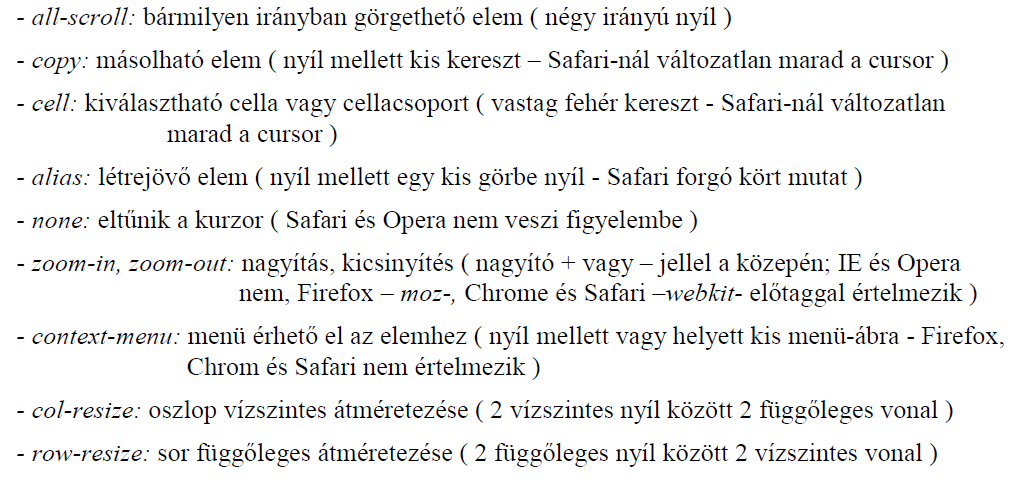
no-display(ha nem fér be akkor eltávolít,mintha display:none lenne aktiválva)

no-content(ha nem illsezkedik a dobozba akkor visibility:hidden lesz rá definiálva)

**Kurzor megjelenítés:**

cursor:





**p:active{**

**color:red;**

**cursor:wait;**

**}**

**Folyamatos átmenet:**Ha egy objektum felé visszük például az egeret ,akkor nem rögtön történik a változás ,hanem időszerűen vált.

div {

width: 100px;

height: 100px;

background: red;

transition-property: width;

transition-duration: 2s;

}

div:hover {

width: 300px;

}

transition-property: width,height,color;

transition-duration: 2s,5s,6s;

A sorrendiség tetszőleges,viszont arra vigyázni kell,hogy a property-be megadott sorrendet követnie kell a durationnak,aholis az időátmenetet rendeljük hozzá.

Opcionálisan két további paraméter is megadható:

-transition-delay: 3s,2s,3s;

itt azt adjuk meg,hogy mennyi időt késlekedve kezdje meg az átmenetet.

alapértelmezetten mindegyikhez nulla tartozik.

**Animáció:**A folyamatos átmenet kiterjesztése.

Ismétlések száma:

animation-iteration-count: egész szám infinite a végtelen

animationn-delay/duration ugyan az mint a transitionnél.

div {

width: 100px;

height: 100px;

background: red;

position: relative;

/\* Chrome, Safari, Opera \*/

-webkit-animation-name: myfirst;

-webkit-animation-duration: 5s;

-webkit-animation-timing-function: linear;

-webkit-animation-delay: 2s;

-webkit-animation-iteration-count: infinite;

-webkit-animation-direction: alternate;

-webkit-animation-play-state: running;

/\* Standard syntax \*/

animation-name: myfirst;

animation-duration: 5s;

animation-timing-function: linear;

animation-delay: 2s;

animation-iteration-count: infinite;

animation-direction: alternate;

animation-play-state: paused;

}

@-webkit-keyframes myfirst {

0% {background:red; left:0px; top:0px;}

25% {background:yellow; left:200px; top:0px;}

50% {background:blue; left:200px; top:200px;}

75% {background:green; left:0px; top:200px;}

100% {background:red; left:0px; top:0px;}

}

@keyframes myfirst {

0% {background:red; left:0px; top:0px;}

25% {background:yellow; left:200px; top:0px;}

50% {background:blue; left:200px; top:200px;}

75% {background:green; left:0px; top:200px;}

100% {background:red; left:0px; top:0px;}

}

:active div{

-webkit-animation-play-state:running;

}

-animation-direction(animáció iránya)

normal(alapértelmezett,a kódolásnak megfelelően,tehát ha jobbra mozgást kódoltam,akkor jobbra mozog)

reverse(fordítva csinál mindent)

alternate(a páros ciklusszámúak nem a kódolás szerint zajlanak(reverse) ,a páratlanok pedig kódolás szerint(normal))

alternate-reverse(itt a páros a normal és a páratlan a reverse)

-animation-play-state: Két lehetséges érték

paused:stoppolt

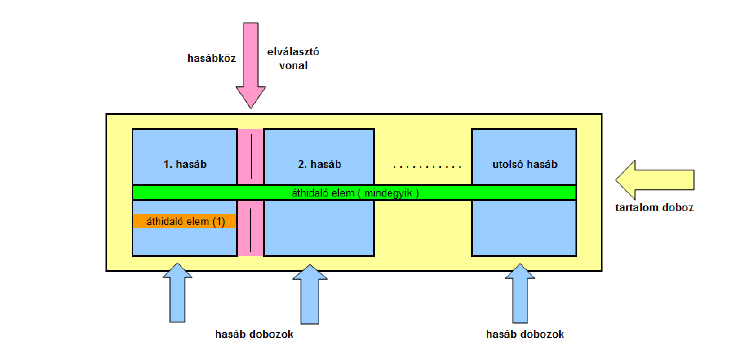
running:futtatott

-animation-name:az animáció azonosítására való

-webkit azért használatos mert nem mindegyik böngésző ismeri fel a sima szabványt.

@ keyframes animation-name{-nél adjuk meg az animációra onatkozó változásokat.

Százalékosan megadható mikor mi történjék.

**Többhasábos elrendezés:**

column-count: hasábok száma

auto(más tulajdonság szabja meg,a hasábok mérete,ugyanis mind ugyan olyan széles és hosszú)

írhatunk oda pozitív egész számot,ami a maximális számot határozza meg.

column-width: hasáb szélesség

auto(más tulajdonság határozza meg)

megadható pozitiv egész szám is

column-gap: hasábköz,aminek hossza megegyezik a hasábokéval.

auto(ennek értékét a böngésző határozza meg)

szám is lehet(javasolt érték 1em)

column-rule: hasábelválasztóvonal

column-rule-width:bármilyen pozitív érték

column-rule-style:none érték 0 szélességet hordoz(border-stylehoz hasonlóan formázható)

column-rule-color:a színe

p{

-webkit-column-count:3;

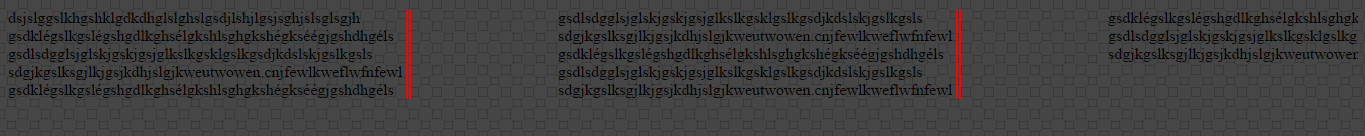
-webkit-column-width:50px;

-webkit-column-rule-width:5px;

-webkit-column-rule-style:double;

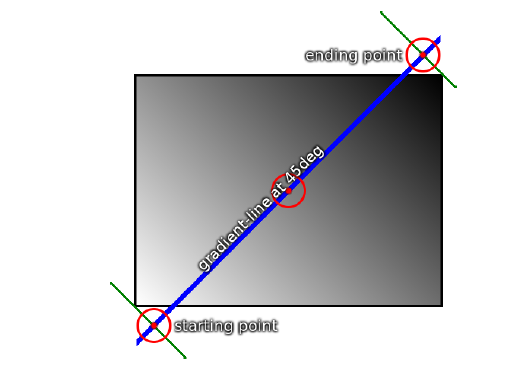
-webkit-column-rule-color:red;

-webkit-column-gap:300px;}



**Színátmenetek:**Akár a háttérkép is létrehozható vele,és sokkal kisebb helyigényű mintha egy képet szerkesztenénk photoshoppal.

lineáris megoldás:



body{

background-image:linear-gradient(45deg,yellow,blue);

}

első paraméter a dőlésszög.

második az alulról kiinduló szín

harmadik a felülről kiinduló szín

background-image:linear-gradient(to bottom,red,white);

felülről lefelé haladva (tehát nem a sarokból) pirosból fehér lesz az alján.

background-image:linear-gradient(to top,red,white);

Ha nincs pozíció megadva,akkor a böngésző egyenletesen osztja el,és nyilván lehet több színt is használni.

background-image:linear-gradient(yellow,blue,white);

**Síkbeli transzormációk(2d):**

X vagy Y tengely mentén eltolás,elforgatás,kicsinyítés,nagyítás elferdítés hajtható végre.

transform: tulajdonság lehetséges értékei

-translateX(…):Vízszintesen tolás,pozitív szám esetén jobbra,negatív esetéb ballra(px ben való megadás)

-translateY(..):Függőlegesen valő eltolás,pozitív szám esetén lefelé,negatívnál felfelé.

összevont alak is létezik:

div:hover{

-webkit-transform:translate(50px,50px);

}

-scaleX():Kicsinyítés vagy nagyítás vízszintes irányba

-scaleY():Kicsinyítés vagy nagyítás függőleges irányba

-webkit-transform:scale(1.5,1.5);

scaleX(-1): az x tengelyre tükröz a scaleY(-1) az y tengelyre tükröz,a kettő együtt pedig mindkettőre.

-rotate: (elforgatás)

rotate(45deg);

origó körül való forgatás,plussz érték az óramutató járásával megyezező irányú.

Egy elemen egyidejűleg több transzformációs művelet is végrehajtható!

div:hover{

-webkit-transform:translate(300px,100px) rotate(145deg) scale(1.5,1.5);

}

-skew: (elferdítés)

skexX():Vízszintes tengely mentés fordítás degben megadva.

skewY():Függőleges tengely mentén fordítás degben megadva

skew() :A kettőt egyszerre megadva.

Transzformáció origójának a kezdetének a megadása:

transform-origin:

x tengely esetén right,center,left szavak

y tengely esetén top,center,bottom szavak használhatóak.

Jobb alsó sarok szerinti transzformáláshoz:

-webkit-transform-origin:right bottom;

**Animációs keverés**:Olyan animáció,hogy egy kocka forogva begördül az oldal közepére.

div {

width: 100px;

height: 100px;

background-color: red;

-webkit-animation-name: myfirst;

-webkit-animation-duration: 5s;

-webkit-animation-play-state: running;

}

@-webkit-keyframes myfirst{

0% {transform:translateX(0px) rotate(0deg); }

/\*25% {transform:scale(0.5,0.5) rotate(1220deg) transleta(5px,5px);}

50% {transform:scale(1,1) rotate(1220deg) transletaX(30px);}

75% {transform:scale(1.5,1.5) rotate(1220deg) transletaX(30px);}\*/

100% {transform:translateX(500px)rotate(4000deg);}

}

Azért elég csak a nulla és a 100%-ot megadni mert a 100%-ig kell 4000 fokot forognia,és ott csak forogni kell,közben nem kell finomítgatni a lépéseken.

**Térbeli transzformációk:**Csak látszólag 3 dimenziós,egyébként nem 3 dimenziós objektumokat maniplálunk.

Ha z értéke o ,negatív szám vagy none,akkor nincs térbeliség.

**css alapbeállítása:**Egyes böngészők alapértelmezetten néhány tulajdonságot másképpen jelenítenek meg.

4 féle módszer van,amivel css-en belül ezekre egységes beállítást szabhatunk.

Itt a leggyakrabban használtat mutatom:

body, p, h1, h2, h3, table, img, ol, ul, form{

margin:0;

padding:0;

}

**Legördülő menü:**Ezt script nyelvekkel is szokás megoldani.

htmlben egy lista létrehozása:

<nav>

<ul class="menu">

<li><a href="css/jatek.html">Magunkról</a></li>

<li><a href="css/jatek.html">A játék</a>

<ul>

<li><a href="css/jatek.html">gép</a></li>

<li><a href="css/jatek.html">alkatrész</a></li>

<li><a href="css/jatek.html">összerakás</a></li>

</ul>

</li>

<li><a href="css/jatek.html">Partnereink</a></li>

<li><a href="css/jatek.html">Egyebek</a></li>

</ul>

</nav>

css formázás:

.menu {

height:40px;

width:500px;

background:green;

border-radius:10px;

}

.menu li{

position:relative;

list-style:none;

float:left;

height:40px;

}

.menu li a{

display:block;

padding:0 15px;

margin:6px 0;

line-height:30px;

text-decoration:none;

font-family:Tahoma, Geneva, sans-serif;

font-weight:bold;

font-size:15px;

color:white;

}

.menu li:hover > a{

color:lime;

}

.menu ul{

position:absolute;

padding:0;

top:40px;

left:0;

background:black;

border radius:0 0 10px 10px;

opacity:0;

}

.menu li:hover > ul{

opacity:1;

}

.menu ul li{

height:0;

overflow:hidden;

padding:0;

}

.menu li:hover > ul li{

height:40px;

overflow:visible;

}

**Konkrét példagyűjtemény:**

-Itt a háttér színét és a szöveg színét adjuk meg,ha esetleg történne valami a css betöltésekor vagy a képek betöltésekor.

<body bgcolor='#DAE2E3' text='#444444'>

</body>

- Tegyünk fel, hogy szeretnénk teljesen az oldal közepére zárni egy 440\*400 pixeles divet, amibe később az oldalunk tartalma kerül. Táblázatos módszerrel ez viszonylag egyszerű, egy három cellás táblázat – ez az, amit el kellene kerülni!

<**div** id="container">

</**div**>

Ekkor a lap szélesség nem fix értékben van megadva.

div#container {

**margin-left**: 10%;

**margin-right**: 10%;

}

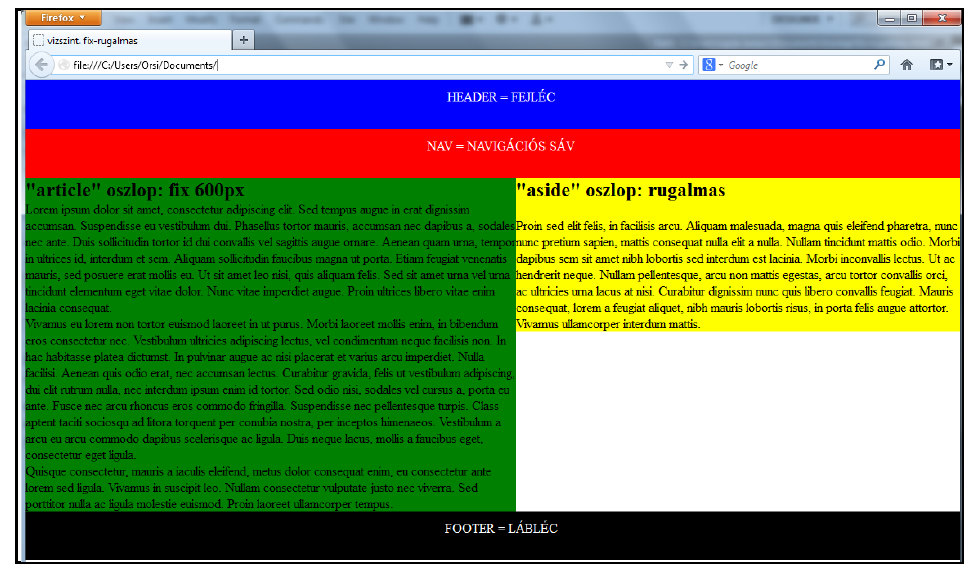
-<meta name='viewport' content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no"/>

content="width=device-width:mindig az aktuális eszköz vagy böngésző szélességéhez fog igazodni.

initial-scale=1.0,:kezdeti zoom az oldalra

maximum-scale=1.0,:meddig nagyíthassa a felhasználó az oldalunkat.

user-scalable=no:a felhasználó egyáltalán tudja-e nagyítani az oldlaunkat.Tud-e rajta zoomolni vagy sem.



html span tag szerepe: A SPAN elemet érdemes egy soron belüli változtatásokra használni.

<p>Anyámnak <span style="color:lightblue">világoskék</span> szeme van.</p>

-margin:0 0 0 0 ; óramutató járásával megegyezően ,felső,jobboldal,alul,baloldal

text-decoration:none; ezzel a szöveg formázást kapcsoljuk ki.

marginokkal meghatározzuk,hogy milyen távolság legyen az egyes bekezdések között.

padding pedig még a doboz,de nem a tartalom rész,és ez után jön a border(hozzáadódik az adott elem szélességéhez).

width:480px akkor a tartalmi részt állítjuk.

**HTML div:**

Divnek alapból 100%-os a szélessége,tehát teljesen ki fogja tölteni a képernyőt.

width:300px

ha egy 10px paddinget hozzárakunk,akkor az hozzáadódik a méretéhez.

mindkét oldalához hozzámegy a 10 px ezért a szélessége innentől kezdve 320px lesz a divnek.

Ha 5px bordert hozzárakunk,akkor az is hozzáadódik a szélességhez(a magassághoz is).a MARGIN A BORDEREN KÍVÜL LESZ AZ MÁR NEM ADÓDIK HOZZÁ A SZÉLESSÉGHEZ.

A weboldalak struktúráját divekkel szoktuk meghatározni.

float: tulajdonság(lebegtetés,lebegtetjük egymás mellett a dobozokat)

értékei left vagy right

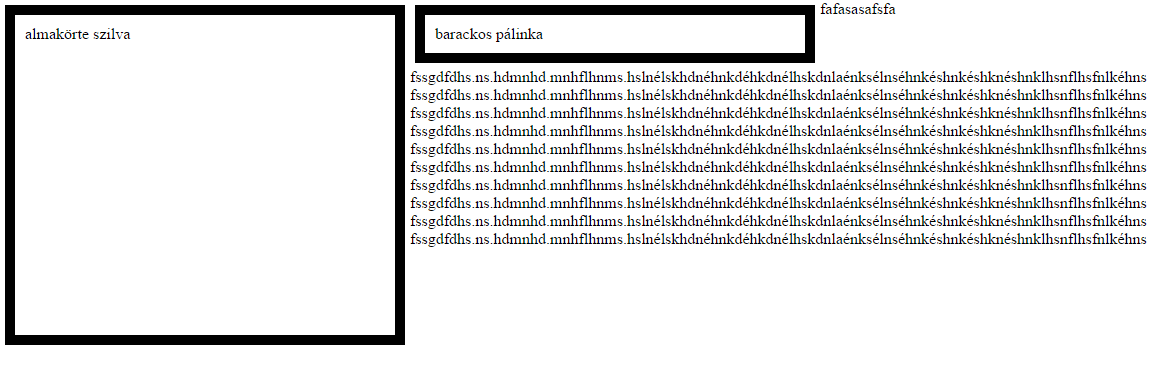
clear: azért kell mert a divek mellé befolyik a többi dolog is,amikre lejjebb lenne szükség,ezért létrehozunk egy üres divet,amire megszüntetjük a floatolást,és ekkor a további elemek nem fognak felfelé floatolni.

div.clear{

clear:both;

}/\*a clear classba tartozó divre módosítunk\*/

Látható a képen,hogy a két div,ami be lett lebegtetve(float)jobbra,azok egymás mellett jobra lebegnek(közöttük a margin miatt van hézag),és a lebegtetés miatt a mellettük lévő helyre belebeg a többi szövegrész,amit alulra szeretnénk.



<div id="jobb">

almakörte szilva

</div>

<div id="bal">

barackos pálinka

</div>

#jobb{

width:360px;

margin:5px 5px;

padding:10px 10px;

border-color:black;

border-width:10px;

border-style:solid;

float:left;

height:300px;

}

#bal{

width:360px;

margin:5px 5px;

padding:10px 10px;

border-color:black;

border-width:10px;

border-style:solid;

float:left;

}

Megoldás arra,hogy a két float mellé ne lebegjen semmi,hogy

<div id="jobb">

almakörte szilva

</div>

<div id="bal">

barackos pálinka

</div>

<div class="clear">

</div>

betettünk egy üres divet,amit css-ben formázunk

css:

div.clear{

clear:both;

}

innentől kezdve a harmadik div elválasztást képvisel és az eredmény:



Látható,hogy a harmadik div úgymond leválasztja a szöveget,és nem floatol a szöveg az első két divhez.

középre igazítás:

margin:0 auto;

**Oldal szélesség csökkenésére dinamikus rendeződés:**

@media only screen/\*megadtuk a min és a max szélességet amikor már más megjelenítést alkalmazunk\*/

and (min-width : 480px)

and (max-width : 800px) {

div#wrapper { width: 600px }

header { font-size: 320% }/\*itt az összehúzás miatt a betűméretet is csökkentjük,hogy ne törje ketté a kicsinyítés\*/

section { width: 408px }

aside { width: 150px }

footer { width: 560px }

}

@media only screen

and (min-width : 320px)

and (max-width : 480px) {

div#wrapper { width: 400px; }

header { font-size: 200%; }

section { width: 400px; padding: 0; margin: 0; border: 0 }

aside { display: none }

footer { width: 360px }

}

@media only screen

and (min-width: 240px)

and (max-width: 320px) {

div#wrapper { width: 300px; }

header { font-size: 200%; text-align: center }

header span { display: block }

nav { text-align: center }

nav a { margin: 0 3px }

nav a:first-child { margin: 0 3px }

section { width: 300px; padding: 0; margin: 0; border: 0 }

aside { display: none }

footer { display: none }

}

Akármennyi media screent állíthatünk,tehát a világ összes képernyőméretére végezhetünk beállítást,csak az sok media screent igényel.