

Számítógép-hálózatok tárgy

CSS alapok

Óra áttekintése

- ☐ Stíluslapok szerepe, helye a kódban
- ☐ CSS stílus megadás
- ☐ Néhány css formázás
- ☐ Osztály, azonosító, látszólagos osztály
- ☐ Dobozmodell, öröklődés
- ☐ Chrome DevTools

Stíluslapok szerepe, helye a kódban

□ A logikai felosztás és a formázás elkülönítése

- design – CSS <-> tartalom – html

□ A formázásokat csoportosítottan

- külön **.css** kiterjesztésű fájlban tároljuk
- vagy a dokumentum elején `<style>` tagek között
- vagy elem paramétereként adjuk meg
- a fentiek közül a legelső felel meg leginkább a css filozófiájának, a harmadik legkevésbé

□ CSS = **C**ascading **S**tyle **S**heet

`<link href="style.css" rel="stylesheet" >`

`<style>`
`p {`
`color:white;}`
`</style>`

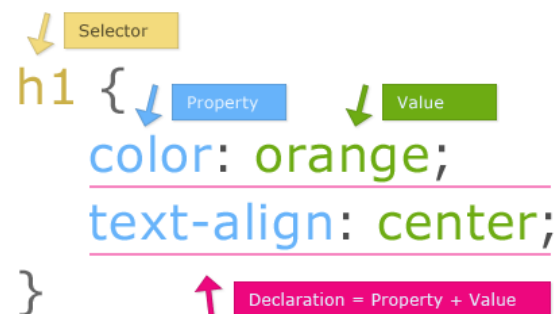
`<p`
`style="color:white">text`
`</p>`

CSS stílus megadás

- Egy CSS stílus megadás két alapvető részre különül el
 - szelektor – formázandó HTML tag (vagy osztály, id, lásd később) neve
 - deklaráció – megadott tag formázása
 - tulajdonság és megadott érték

- Alapformája:

- szelektor {tulajdonság: érték;}
- például: p {color:red;}



Csoportosíthatóság, comment

- Mind a szelektorok, mind a deklarációk csoportosíthatók

```
h2, h4, h6, p {color: #082311;}
```

```
p  
{  
color: white;  
background-color: red;  
}
```

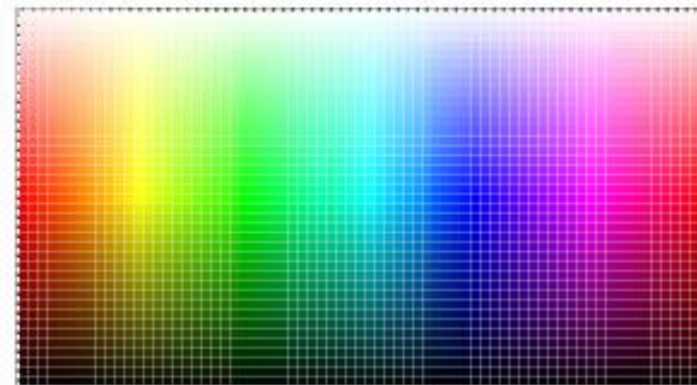
- Megjegyzés:
/ megjegyzés */*

30px: css-ben nem lehet space a mértékegység és az érték között!

Formázások stílussal

- ❑ font-style – betű stílus
 - értéke lehet: italic (dőlt), normal
- ❑ font-weight – betű vastagság
 - értéke lehet: bold (félkövér),
 - ❑ egyéb lehetőségek: normal, 100, 200, 300, 400 (=normal), 500, 600, 700 (=bold), 800, 900
- ❑ font-size – betűméret beállítás
 - értéke lehet: abszolút (px)
 - ❑ egyéb lehetőségek: %, xx-small, x-small, small, medium, large, x-large, xx-large, larger, smaller
- ❑ text-align – szöveg igazítás vízszintesen
 - értéke lehet: left (alapértelmezett), right, center, justify
- ❑ color: írásszín
- ❑ background-color: háttérszín

Színek



- Az alap színek megadhatók angol nevükkel is, egyébként RGB kóddal
- RGB kód – pl.: #89AF81 (hexadecimális)
 - első két jegy piros, majd zöld, és kék
 - 00 a leggyengébb, FF a legerősebb
 - más megadás: `rgb(255,0,0);`
- <http://www.colorpicker.com/>

Alapszínek

black (#000000)	green (#008000)
silver (#C0C0C0)	lime (#00FF00)
gray (#808080)	olive (#808000)
white (#FFFFFF)	yellow (#FFFF00)
maroon (#800000)	navy (#000080)
red (#FF0000)	blue (#0000FF)
purple (#800080)	teal (#008080)
fuchsia (#FF00FF)	aqua (#00FFFF)

Kód validálás

❑ W3C html validátora:

- <http://validator.w3.org/>

❑ W3C CSS validátora:

- <http://jigsaw.w3.org/css-validator/>
- CSS level választandó, css3 jelenleg
- *"Congratulations! No Error Found. This document validates as CSS level 3!"* vagy
- *"Sorry! We found the following errors (1)"* és *"11 dt Value Error : color blu is not a color value"*: elgépelés

Már nem használatos formázó megoldások

- Ezeket ma már css-el oldjuk meg
- Html4 transitional doctype-ot kellett használni hozzájuk
 - `...`
 - `<body bgcolor="red" text="white" link="blue" background="vmi.jpg">`
 - `<hr size="3" align="center" width="80%">`
 - `<p align="left">...</p>`
 - `<ul type="square">`
 - stb.



Osztály – Class

- ❑ A HTML kód body részében található minden elem osztályozható
- ❑ Használat:
 - head-ben (vagy css file-ban) létrehozzuk az osztályt:
 - ❑ `doboz.osztalynev`{formázási beállítások;} vagy
 - ❑ `.osztalynev` {formázási beállítások;}
 - body-ban használjuk:
 - ❑ `<doboz class="osztalynev">ezt formázom</doboz>`

Class - példa I.

Különböző színű paragrafusok

```
<style>
```

```
  p.ps{color:red;}
```

```
  p.kek{color:blue;}
```

```
</style>
```

```
...
```

```
<body>
```

```
  <p class="ps"> Lorem ipsum dolor sit amet, consectetur  
adipiscing elit. </p>
```

```
  <p class="kek"> Vestibulum id magna ut est auctor  
bibendum. </p>
```

```
  <p> In vel erat sit amet sapien ullamcorper volutpat vel sit  
amet lectus. </p>
```

```
</body>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Vestibulum id magna ut est auctor bibendum.

In vel erat sit amet sapien ullamcorper volutpat vel sit amet lectus.

Class - példa II.

Egy class- több html tag

```
<style>
    .ps{color:red;}
</style>
...
<body>
    <h1 class="ps">Lorem ipsum dolor sit amet, consectetur
    adipiscing elit.</h1>
    <p class="ps">Vestibulum id magna ut est auctor
    bibendum.</p>
</body>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Vestibulum id magna ut est auctor bibendum.

Class - példa III.

Span – mikor szükséges?

```
<style>
```

```
    span.zold{color:green;}
```

```
    span.ps{color:red;}
```

```
</style>
```

```
...
```

```
<body>
```

```
    <p>L<span class="zold">or</span>em ipsum dolor sit  
    amet, consectetur adipiscing elit.</p>
```

```
    <p>Ves<span class="ps">t</span>ibulum id magna ut  
    est auctor bibendum.</p>
```

```
</body>
```

Lo**r**em ipsum

Ves**t**ibulum id

Class - példa IV.

Span – több taghez

```
<style>  
    span.zold{color:green;}  
    p span.ps{color:red;}  
</style>
```

...

```
<body>  
    <h3>L<span class="zold">or</span>em ipsum dolor sit  
    amet, consectetur adipiscing elit.</h3>  
    <p>L<span class="zold">or</span>em ipsum dolor sit  
    amet, consectetur adipiscing elit.</p>  
    <p>Ves<span class="ps">t</span>ibulum id magna ut  
    est auctor bibendum.</p>  
</body>
```

Lorem ipsum dolor sit

Lo**re**m ipsum dolor sit

Ves**t**ibulum id magna ut

Class - Új tulajdonság hozzáadása

<style>

p.ps{color:red;}

</style>

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

...

In vel erat sit amet sapien ullamcorper volutpat vel sit amet lectus.

<body>

<p class="ps"> Lorem ipsum dolor sit amet, consectetur adipiscing elit. **</p>**

<p>In vel erat sit amet sapien ullamcorper volutpat vel sit amet lectus. **</p>**

</body>

Class - Öröklött tulajdonság felülírása

```
<style>
```

```
  p {background-color: gray;}
```

```
  p.ps {background-color: red;}
```

```
</style>
```

```
...
```

```
<body>
```

```
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing  
  elit.</p>
```

```
  <p class="ps"> In vel erat sit amet sapien ullamcorper  
  volutpat vel sit amet lectus. </p>
```

```
  <p>Proin egestas lectus in quam iaculis elementum. </p>
```

```
</body>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

In vel erat sit amet sapien ullamcorper volutpat vel sit amet lectus.

Proin egestas lectus in quam iaculis elementum.

Azonosítót akkor használunk (osztállyal szemben), ha egyszer szerepel egy oldalon, pl fejléc, lábléc, menü, stb

Azonosító – Id

- A HTML kód body részében található minden elemhez hozzárendelhető
- Használat:
 - head-ben (vagy css file-ban) létrehozzuk az azonosítót:
 - `doboz#azonositonev`{formázási beállítások;}
 - `#azonositonev` {formázási beállítások;}
 - body-ban használjuk:
 - `<doboz id="azonositonev">ezt formázom</doboz>`

Div és id - példa

```
<style>  
  #fejlec {background-color: lightgray;}  
  #fejlec h1,h2,h3 {color:red;}  
</style>
```

*Azért, mert az
oldalon egyedi
előfordulású*

Szürke háttérű fejléc

Menü

Almenü

```
<body>  
  <div id="fejlec">  
    <h1>Szürke háttérű fejléc</h1>  
    <h2>Menü</h2>  
    <h3>Almenü</h3>  
  </div>  
</body>
```

Azért div, mert egy nagyobb,
oldal-szintű egység.

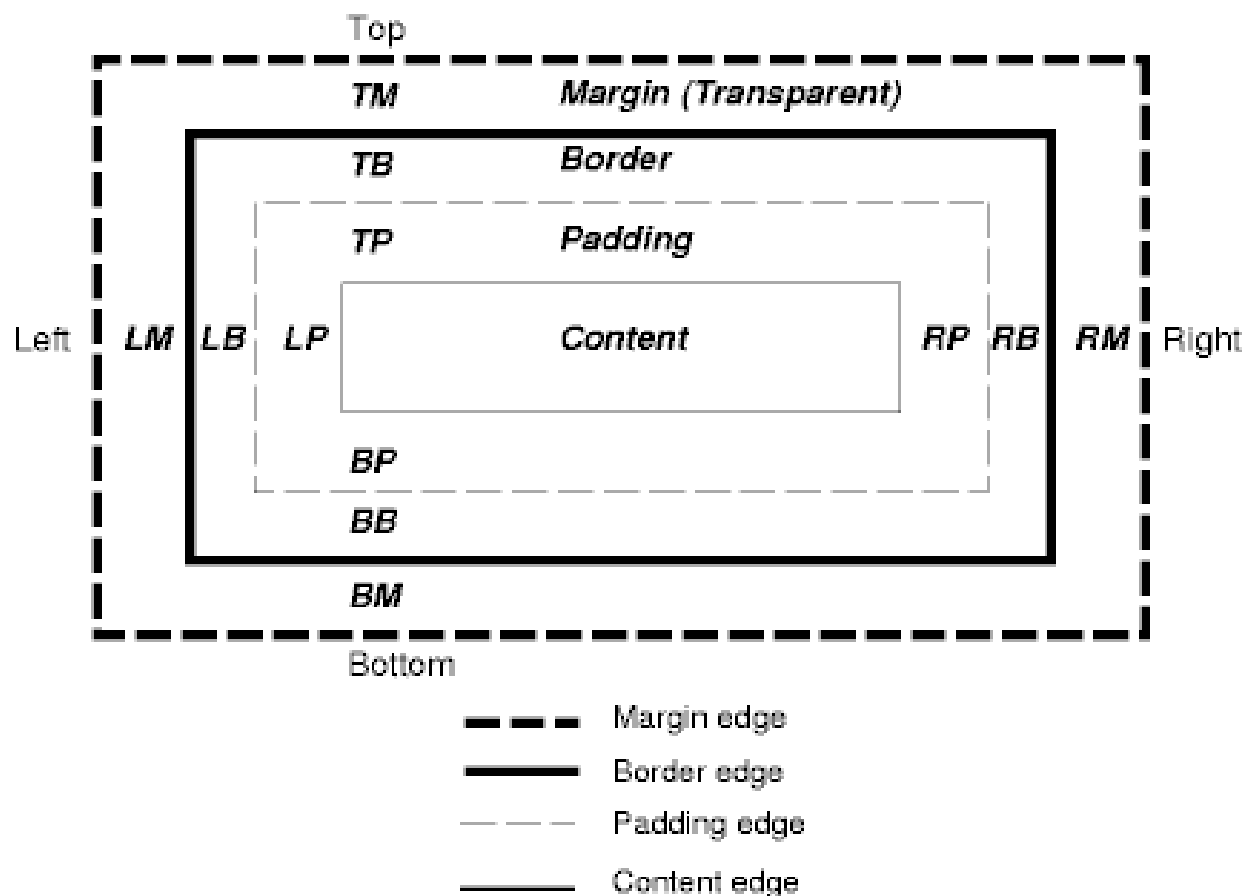
Dobozmodell

- A HTML dokumentum dobozokra tagolódik
- A legnagyobb doboz a body
- Ezen belül különböző tag-ek dobozokat alkotnak
 - h1, p
 - ul, ol, stb.
- Bizonyos dobozokon belül további dobozok (gyerek dobozok) vannak, pl.:
 - ul -> li -> li.osztalynev
- Az egyes dokumentumrészek öröklik a tartalmazó (hierarchiában felettük álló) szülő dobozok tulajdonságait
- Például ha a body háttere kék, akkor a p, h1, li, stb. háttere is kék

Preferenciák

1. Ha valamelyik szabály után !important jelölés van, az mindent üt. Pl. color: fuchsia !important;
2. Ez után a specificitás kerül figyelembevételre.
 - a) Leginkább egy id-val azonosított, konkrét elemre vonatkozó szabály specifikus.
 - b) Ezt követi egy adott class-ra vonatkozó szabály
 - c) Legkevéssé egy adott element típusra vonatkozó szabály specifikus
3. Ha a specificitás megegyezik, a deklaráció sorrendje dönt. Olyan ez mint a végrendelet: a legutolsó számít. 😊

Dobozmodell tulajdonságai



Dobozmodelles példák

```
p{  
background-color: aqua;  
margin: 70px;  
border: dashed 3px red;  
padding:50px; }
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut euismod purus vel malesuada auctor. Phasellus libero augue, varius quis hendrerit quis, auctor a lorem. In volutpat magna sit amet suscipit condimentum. Donec ut lobortis quam. In malesuada risus ipsum, eget accumsan neque tristique eu.

```
p{  
border: 3px solid rgb(256,0,0);  
border-radius: 30px;  
padding:20px;  
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut euismod purus vel malesuada auctor. Phasellus libero augue, varius quis hendrerit quis, auctor a lorem. In volutpat magna sit amet suscipit condimentum. Donec ut lobortis quam. In malesuada risus ipsum, eget accumsan neque tristique eu.

Látszólagos osztály

□ Linkek látszólagos osztályai:

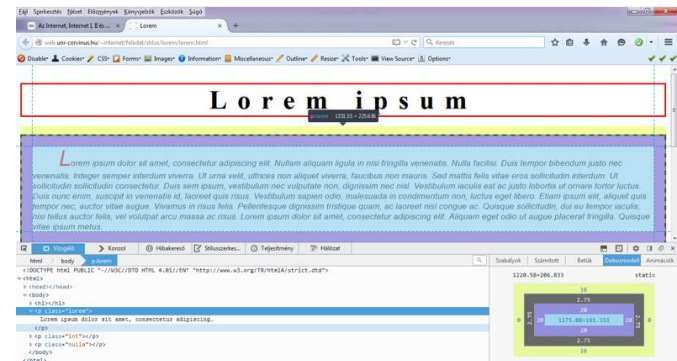
- a:link – link stílusa
- a:visited – látogatott link stílusa
- a:active – aktív link stílusa
- a:hover – a link stílusa, mikor az egérrel rámutatunk

Böngészőbeli fejlesztőeszközök CSS íráshoz – F12

□ Chrome



□ Firefox



Böngészőbe integrált fejlesztői eszközök – részletes ismertető:
<http://web.uni-corvinus.hu/~internet/hasznos/DevTools/DevTools.html>

Inline vs. blockszintű elemek

□ **Inline** (szövegszerű) elemek

- alapértelmezetten inline szintű pl: ``, `<a>`, ``
- ha több ilyen elemet teszünk egymás után, akkor azok szövegszerűen, egymás mellett jelennek meg, amíg azt a sor hosszúsága engedi
- CSS-sel nem tudjuk megadni a méretét: mindig olyan szélesek és magasak lesznek, mint a bennük lévő tartalom.

□ **Block** (dobozszerű) elemek

- alapértelmezetten block szintű pl : `<div>`, `<p>`, `<h1>`-`<h6>`, `<form>`, ``, ``, ``
- egymás alatt jelennek meg, és vízszintesen teljes szélességben kitöltik a rendelkezésükre álló helyet, míg a magasságuk a bennük lévő tartalomhoz igazodik
- CSS-sel szabadon beállíthatjuk a szélességüket (`width`), a magasságukat (`height`), külső és belső margójuk méretét (`margin`, `padding`), valamint adhatunk nekik keretet is (`border`)

Ha érdekel a css – további anyagok

- ❑ Ezen az órán csak kis részével foglalkoztunk a CSS anyagnak. Ha több is érdekel:
- ❑ CSS feladatok megoldással: <http://web.uni-corvinus.hu/~internet/feladat/stilus/stilus.html>
- ❑ Fóliákon több és részletesebb anyag: http://web.uni-corvinus.hu/~internet/webf_foliak.html
- ❑ <http://web.uni-corvinus.hu/~internet/hasznosak.html>
„Stíluslapok” részben hasznos linkek