

CSS-től az űrlapig

Ismétlés

- ▶ Érdemes megjegyezni: a HTML nyelv nem azért készült, hogy segítségével formázzuk a kinézetét/stílusát az oldalunknak, hanem azért, hogy leírjuk vele az oldalunk tartalmát, ezeket „tag”-ek segítségével tehetjük meg. A szövegünket csoportosítani kell a <div> tag segítségével és az egyforma alapstílusú részeket class nevekkal csoportosíthatjuk, ezeken belül a speciális tulajdonságú részeknek egyéni azonosítót ID-t adhatunk. Ezeket az attribútumokat bármelyik HTML tag névbe beírhatjuk. (pl: <div class=„bekezdés”>szöveg más szöveg szöveg</div>)
- ▶ Lehetőség van egy tag-hez több class (csoportot) is rendelni ezeket szóközzel választjuk el egymástól az attribútum nevében: (pl: <div class=„egyik másik”>két csoport tulajdonságai is érvényesek erre a szövegre</div>)
- ▶ Ilyenkor a különböző tulajdonságok „összeadódnak”, ha mindkét classban egyforma tulajdonság is van, akkor nem lehetünk biztosak benne, hogy melyik azonos tulajdonság értéke melyik értéket ír felül. (tehát ezt kerüljük)

Ismétlés CSS

- ▶ A CSS azért készült, hogy segítségével formázhassuk a HTML oldalunkat. Ereje abban rejlik, hogy külső fájlban is megadhatjuk a stílusleírásokat, amiket bármennyi HTML fájlban felhasználhatunk (így létrehozhatunk stílustémákat).
- ▶ Jó szokás a kódunkat rendszeresen kommentelni és karbantartani, minél lényegre törőbben leírni (nem használatos tulajdonságokat hagyjuk el).
- ▶ CSS szintaxisa:
tagnév vagy #ID vagy .classnev {
 tulajdonsag:ertek;
}
- ▶ Több classnévvel (vagy akár ID-vel) is rendelkező elemre hivatkozhatunk egyértelműen (ezek sorrendje lényegtelen)
HTML: <div class=„egyik másik” ID=„id”>valami</div>
CSS:
.egyik.masik#id {
 tulajdonsag:ertek;
}
- ▶ Több class egyforma tulajdonságait is egyszerre megadhatjuk:
.egyik, .masik, .harmadik { tulajdonsag:ertek; }

CSS választók

- ▶ A CSS selectors (választók) segítségével azt adjuk meg, hogy milyen feltételek mellett milyen elemeket válasszon ki (és formázzon) a CSS.

pl. classnevekhez tartozó stílusok (tulajdonságok) milyen események, feltételek mellett változzanak.

```
/* Ennek segítségével megváltoztathatom egyik  
elemem tulajdonságát, ha az egérrel rámegyek a  
(classnevem csoportba tartozó) szövegre
```

```
*/
```

```
.classnevem:hover {  
    tulajdonsag:ertek;  
}
```

```
/* Ha elhagyom az egérrel a szöveget akkor a  
tulajdonság visszaáll az eredeti alapértékre
```

```
*/
```

```
.classnevem {  
    tulajdonsag:alapertek;  
}
```

- ▶ http://www.w3schools.com/cssref/css_selectors.asp

CSS választók

▶ Hasznos választók:

- ▶ , (vessző) - két különböző elem kiválasztása
pl.: Több class egyforma tulajdonságait is egyszerre megadhatjuk:
.egyik, .masik { tulajdonsag:ertek; }
- ▶ elem1 elem2 (szóköz) - kiválasztjuk azokat az elemeket amelyeknél jobboldali elem a baloldali leszármazottja
- ▶ .class - classnév kiválasztó
- ▶ #id - azonosító kiválasztó
- ▶ [attribútum megadása] - minden olyan elemet kiválasztunk amiben meg van adva ez az attribútum
- ▶ [valami]:hover - minden olyan elemet formázunk ami felett van az egerünk éppen
- ▶ * - mindent kiválasztunk

CSS megadása

- ▶ 3 féleképpen lehet html-hez rendelni:
 - ▶ Inline attribútumként (`<div style="tulajdonsag:ertek;">szöveg</div>`)
 - ▶ A `<head>` fejlécen belül `<style>...</style>` tag-ek között
 - ▶ Külső fájlból (ez a legpraktikusabb, html kódtól függetlenül is módosíthatjuk a stílus leírásokat bármikor és más html-ben is újrahasznosítható, felhasználható)

Tipp:

A böngészőnkben az URL cím végére írhatjuk a saját #azonosító , (ID-nket) és így arra a részre ugrik a böngészőnk az oldalon automatikusan ahol az ID definiálva van (ez hatáto módszer könyvjelzőknél is)

Különleges HTML5 tag-ek

- ▶ `<video>` - videó fájl (akár .mp4 akár .webm) belinkelése az oldalunkra (be kell zárni `</video>`)

Lehetséges attribútumok pl:

`width="320" height="240" autoplay controls`

- ▶ Ezen belül meg kell adni a relatív elérési útvonalat a fájlhoz, meg a típusát:

```
<source src="movie.mp4" type="video/mp4">
```

(nem kell bezárni)

- ▶ `<audio>` - zene/hang fájl belinkelése (be kell zárni `</audio>`)

Lehetséges attribútumok pl:

`autoplay controls loop`

- ▶ Ezen belül megadjuk a relatív elérési útvonalat

```
<source src=„hang.mp3" type="audio/mpeg">
```

(nem kell bezárni)

- ▶ Minden kulcsszó (tag) böngésző támogatottsága megtalálható a <http://www.w3schools.com/> oldalon, egy forráskódnál érdemes meggondolni és mérlegelni a kompatibilitási problémákat.

Tipp:

Az `` taget se kell bezárni

Különleges HTML5 tag: <svg>

- ▶ Az <svg> (Scalable Vector Graphics) segítségével saját magunk által definiált alakzatokat hozhatunk létre
- ▶ `<svg width=„,100” height=„,100”>`
- ▶ pl.1:
`<circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />`
- ▶ pl.2:
`<rect width="300" height="100" style="fill:rgb(0,0,255);stroke-width:3;stroke:rgb(0,0,0)" />`
- ▶ pl.3:
`<polygon points="200,10 250,190 160,210" style="fill:lime;stroke:purple;stroke-width:1" />`
- ▶ pl.4:
`<text x="0" y="15" fill="red" transform="rotate(30 20,40)">I love SVG</text>`
- ▶ `</svg>`

Tipp:

Figyeljünk oda a tagek szabályos bezárására, pl:

HIBÁS: `<i>szöveg</i>`

HELYES: `<i>szöveg</i>`

CSS függvények

- ▶ Meghívásuk paraméterekkel történik az érték helyén:
`.classnevem {
 tulajdonsag:fuggveny(param1,param2);
}`
- ▶ Ezeknél különösen oda kell figyelni a kompatibilitási problémákra, ugyanis a legtöbb böngésző csak a legutóbbi frissítésekben támogatja az általános függvénynevet (standard syntax).
- ▶ Pl:

```
#grad {  
background: -webkit-linear-gradient(left, red , blue); /* Safari 5.1 to 6.0 */  
background: -o-linear-gradient(right, red, blue); /* Opera 11.1 to 12.0 */  
background: -moz-linear-gradient(right, red, blue); /* Firefox 3.6 to 15 */  
background: linear-gradient(to right, red , blue); /* Standard syntax */ }  
}
```

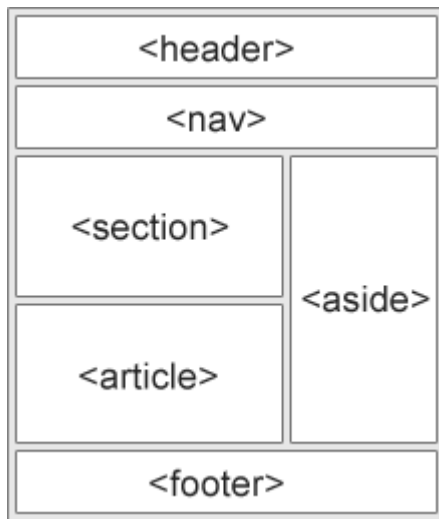
- ▶ http://www.w3schools.com/cssref/css_functions.asp

HTML színek

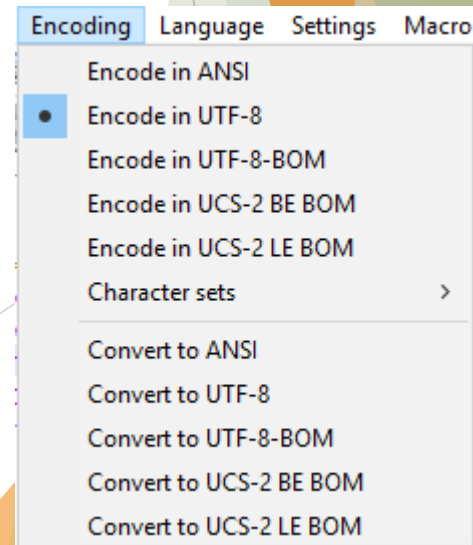
- ▶ A színeket megadhatjuk névvel:
Minden böngésző által támogatott nevek:
http://www.w3schools.com/html/html_colornames.asp
- ▶ Hexadecimális kóddal: (#RRGGBB, ahol minden pár 00-ff -ig)
pl: piros= #ff0000
- ▶ Vagy függvénnyel:
-(rgb(R,G,B); ahol minden szín 0-255 -ig terjed decimálisan)
pl: `color:rgb(255,0,0); /* Ezt minden böngésző támogat */`
-meg lehet adni átlátszósággal együtt is:
pl: `color:rgba(255,0,0,0);`
`/* Ahol 'a' paraméter az átlátszóság ami egy tizedes szám 0.0-
tól(teljesen átlátszó) 1.0-ig(teljesen telített) *//* IE9+, Firefox 3+,
Chrome, Safari, and in Opera 10+ támogatja */`
-meg lehet még adni százalékos értékekkel is
pl: `color:rgba(100%,0%,0%,1);`
- ▶ CSS segítségével:
`div.background {
background: url(kepem.jpg) repeat;
opacity: 0.6;
filter: alpha(opacity=60); /* For IE8 and earlier */ }`
- ▶ http://www.w3schools.com/tags/ref_colorpicker.asp

HTML elrendezés (layout)

- ▶ `<div>` classok segítségével definiálhatunk különböző szekciókat a honlapunkon
- ▶ Létrehozhatunk aloldalakat, amik a pl. oldalsó menüből kiválaszthatóak és így link segítségével egy aloldalunkra juthatunk. A `.css` fájlunk mindegyik oldalra ugyanazokat a layout téma stílus tulajdonságokat írja le, tehát ezt mindegyik `.htm` oldalban belinkeljük (`<link rel="stylesheet" href="stilus.css">`)



Tipp:
Figyeljünk a fájlunk UTF-8 kódolására



CSS3

- ▶ A CSS3 segítségével különleges stílusformázásra van lehetőségünk, pl:
 - ▶ Kerekített/kép dobozszegély
 - ▶ Gradiens színátmenetek
 - ▶ Árnyék effekt
 - ▶ Különleges betűtípusok (font (.ttf,.woff))
 - ▶ 2D/3D transzformációk
 - ▶ Átmenetek/animációk (@keyframes segítségével definiálhatunk egyéni animációkat, a kulcspozíciókat kell megadni %-os formában, ezeknél meghatározzuk a tulajdonságváltoztatásokat, majd az animáció hosszát animation-duration segítségével másodpercben adjuk meg)
(vagy transition segítségével másodperc pontosan szabályozhatjuk, hogy milyen gyorsan változzon be egy tulajdonságváltozás (pl forgás, méretváltozás))

Tipp:

z-index segítségével megadhatjuk, hogy melyik elem melyik elem felett helyezkedjen el (z koordináta felénk irányul)(position attribútumot be kell állítani)

Űrlapok

- ▶ Űrlapokat (Form) használunk akkor, hogyha a felhasználótól adatokat szeretnénk összegyűjteni/bekérni.
- ▶ `<form>` tag-ek között található ez a rész
- ▶ `<input>` ez lesz a legfontosabb elemünk, definiálhatjuk elküldő gombként (submit), üres beviteli mezőként (text) és radio gombként (radio) is (checked attribútum megadható) Itt definiálni kell a `name=„sajatnev”` attribútumot, mivel a php ez alapján azonosítja az adatot, alapértéket is adhatunk neki: `value=„alapertek”`
- ▶ A bevitt adatokat elküldjük egy feldolgozó szervernek, php szkriptnek
- ▶ A PHP általános szerveroldali szkriptnyelv dinamikus weblapok készítésére
- ▶ A `<form>` tag-ben attribútumként definiáljuk a php szkript elérési útvonalát (`action="action_page.php"`) és a feldolgozás irányát (`method=„get”` vagy `„post”`)
- ▶ Egyéb elemek:
http://www.w3schools.com/html/html_form_elements.asp

PHP

- ▶ PHP szkriptek futtatásához és értelmezéséhez webszerverre van szükségünk. Dinamikus weboldalakat hozhatunk létre a segítségükkel.

A WAMP LAMP XAMPP szoftvercsomagok, van bennük:

- ▶ Windows vagy Linux oprendszerhez
- ▶ Apache HTTP szerver
- ▶ MySQL adatbázis kezelő
- ▶ PHP/Perl értelmező