



CSS ALAPOK

KÉSZÍTETTE: OLÁH KATALIN



CSS

- CSS = **C**ascading **S**tyle **S**heets (rangsorolt vagy lépcsőzetes stíluslapok)
- A weblap vizuális megjelenést határozza meg a stíluslap.
- Szöveges fájl, **.css** a kiterjesztés.
- Ajánlott külső stíluslapot használni, de a weblapba is beágyazható.
- A formázási stílusokkal tudja módosítani, formázni a weblap elemeit.
- Jelenleg a CSS3 a legfrissebb stíluslap szabvány.



CSS STÍLUSLAPOK HASZNÁLATA

HÁROM LEGGYAKORIBB MÓDJA

KÜLSŐ STÍLUSLAP (CSS FÁJL) HASZNÁLATA

- Először létre kell hoznod egy CSS fájlt (pl.: style.css), amelyben a későbbiekben definiálsz a különböző stílusokat.
- A HTML és CSS fájl között ki kell alakítanod a kapcsolatot, amit a HTML fájlban határozatsz meg a következő formában:

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

- A HTML fájl <head> részébe a <title> alá kell beillesztened az előbbi HTML kódot.
- A <link> HTML elem tulajdonságai:
 - A **href** attribútum **kötelező** megadnod. Ez mutatja meg a külső CSS fájl elérési útvonalát.
 - A **rel** (relation) attribútum szintén **kötelező** tulajdonság. Ez a kapcsolat/viszony típusát határozza meg (stíluslap).
 - A **type** attribútum elhagyható. A CSS fájl típusát határozza meg (szöveges fájl)

BEÁGYAZOTT STÍLUSLAP – EMBEDDED STYLESHEET

- Beágyazott stíluslapot (embed stylesheet) a HTML oldalon belül a **head részben** definiáljuk.
- A **<style> </style>** HTML elemek közé hozhatod létre a stílusokat.
- A **type** attribútum nem kötelező megadnod. Ezzel a tulajdonsággal határozhatod meg a típusát.

```
<head>
    ...
    <style type="text/css">
        /*stílusok*/
    </style>
</head>
```

SORKÖZI FORMÁZÁS – INLINE CSS

- A sorközi formázást (inline css) a HTML oldalon belül közvetlen az adott HTML elemre kell alkalmaznod.
- Az adott HTML elemre alkalmaznod kell a **style** attribútumot.
- A style tulajdonság értéke, nem egy stílus definíció, hanem a **CSS tulajdonság-érték párokat pontosvesszővel** fel kell sorolnod, ami az adott elemre fog vonatkozni.
- Példa (egy bekezdés sorkizárt igazítású és kék színű)

```
<p style="text-align: justify; color: blue"> szöveg </p>
```



MEGJEGYZÉS KÉSZÍTÉSE

MEGJEGYZÉS - COMMENT

- CSS fájlba megjegyzéseket helyezhetsz el a `/*` és `*/` jelek közé.

```
/*megjegyzés szövege*/
```




SZÍNMEGADÁSI MÓDOK

SZÖVEGES SZÍNEK

- A színek legegyszerűbb megadási módja az angol szín nevek, például: black, orange, green, ...
- Ezek beépített angol szín nevek, amelyeket a böngészők értelmezni tudnak.
- További beépített színek:
 - <https://www.w3.org/TR/css-color-3/#svg-color>
 - https://www.w3schools.com/cssref/css_colors.asp

RGB SZÍNKÓD

- Az RGB szín additív színkeverési móddal állítható elő, amihez az `rgb()` függvényt használjuk.
- Az **R** a piros, a **G** a zöld és a **B** a kék alapszín jelenti. Mindhárom szín 0 és 255 közötti egész számot vehet fel értékül.



- RGB színkód felépítése:

```
rgb(piros, zöld, kék)
```

- Példák:

- Fekete: `rgb(0, 0, 0)`
- Fehér: `rgb(255, 255, 255)`
- Kék: `rgb(0, 0, 255)`

RGB SZÍNKÓD – ÁTTETSZŐ SZÍNEK

- Lehetőség van áttetsző színek előállítására is.
- Ebben az esetben az `rgba()` függvényt kell használnod.
 - A 4. paraméter segítségével állíthatod be a szín `alpha` csatornáját, ami azt határozza meg, hogy hány százalékban látható a szín.
 - Az `alpha` egy 0 és 1 közötti valós szám lehet.



- RGB színkód felépítése:

```
rgba(piros, zöld, kék, alpha)
```

- Példák:

- Fekete (40%-ban látható, 60%-ban áttetsző):
`rgb(0, 0, 0, 0.4)`
- Fehér (20%-ban látható, 80%-ban áttetsző):
`rgb(255, 255, 255, 0.2)`
- Kék (70%-ban látható, 30%-ban áttetsző):
`rgb(0, 0, 255, 0.7)`

HEXADECIMÁLIS SZÁMMAL

- A színeket meg tudod adni 16-os számrendszerbeli színkóddal is.

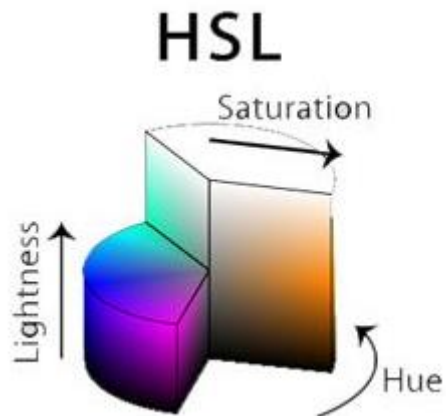
#RRGGBB

- A színkód **#** (kettőskereszt) jellel kezdődik, ami jelzi, hogy hexadecimális színkód.
- A R (red, piros), G (green, zöld), B (blue, kék) 0-tól 9-ig a-tól f-ig kaphat értéket.
- CSS3 szabványban ha páronként megegyezik az érték akkor rövidíteni lehet a színkódot #RGB alakra.
- Például:
 - Fekete: #000000, vagy #000
 - Fehér: #ffffff, vagy #fff
 - Kék: #0000ff, vagy #00f
- Új színek kikeveréséhez próbáld ki a [w3schools színkeverőjét](#).

	#ffffff
	#ecf9ec
	#d9f2d9
	#c6ecc6
	#b3e6b3
	#9fdf9f
	#8cd98c
	#79d279
	#66cc66
	#53c653
	#40bf40
	#39ac39
	#339933
	#2d862d
	#267326
	#206020
	#194d19
	#133913
	#0d260d
	#061306
	#000000

HSL / HSLA SZÍNKÓDOK

- A HSL szín-koordináta-rendszer egy másik módszert biztosít a színek kifejezéséhez.
 - H – hue – árnyalat (0 és 360 közötti egész szám)
 - S – saturation – színtelítettség (0 és 100 közötti egész szám)
 - L – lightness – fényesség (0 és 100 közötti egész szám)
 - A – alpha – láthatóság (0 és 1 közötti valós szám)



- RGB színkód felépítése:

```
hsl(hue, saturation, lightness)
```

vagy

```
hsla(hue, saturation, lightness, alpha)
```

- Példák:

- Fekete: `hsl(0, 0%, 0%)`
- Fehér: `hsl(0, 0%, 100%)`
- Kék (féláttetsző):
`hsla(240, 100%, 50%, 0.5)`



MÉRTÉKEGYSÉGEK

ABSZOLÚT MÉRTÉKEGYSÉGEK

- Az érték nem függ más elemek értékétől.
- Konkrét, fix értéket jelent.
- Gyakran használt abszolút mértékegységek:
 - px – pixel – képpont
 - in – inch
 - pt – pont $1\text{pt} = 1/72\text{in}$
 - cm – centiméter
- A szám és a mértékegység között nem lehet szóköz!



20px

RELATÍV MÉRTÉKEGYSÉGEK

- Egy másik elem (pl. szülő elem) értékétől függ, az adott elem értéke.
- Általában a betűméret, sormagasság, szélesség, magasság, margók beállításánál szoktuk használni.
- Gyakran használt relatív mértékegységek:
 - % – percent – százalék
 - em – arány – a szülő elem beállításához képesti arány (valós szám)
 - rem – arány – a gyökér (HTML) elemre beállított betűméretet veszi alapul. (valós szám)
- A szám és a mértékegység között nem lehet szóköz!
- Részletesebb leírás a mértékegységekről: cyrex.hu



1.5em



KIJELÖLŐK, SZELEKTOROK

A NÉGY LEGYAKRABBAN HASZNÁLT SZELEKTOR TÍPUS

STÍLUS ÁLTALÁNOS FELÉPÍTÉSE

- A külső és beágyazott stílus esetén, az elemek formázásához ún. **kijelölőket/szelektorokat** kell definiálnod.
- Általános felépítése:

```
szelektor_neve{  
    tulajdonság: érték;  
    ...  
}
```

- Minden szelektor rendelkezik egy **névvel**.
- A **{ }** jelek között úgynevezett **CSS tulajdonság-érték párok** segítségével határozhatod meg, hogy milyen stílusok legyenek érvényesek az adott HTML elem(ek)en.
- A CSS tulajdonság-érték párokat **pontosvesszővel** kell lezárni. Az utolsó esetén elhagyható a jel.

4 LEGGYAKRABBAN HASZNÁLT SZELEKTOR TÍPUS

- Többféle szelektor/kijelölő típus létezik, de bevezetésül a 4 leggyakrabban használt szelektor típusról lesz szó:
 1. Elem kijelölő – Type Selector
 2. Osztály kijelölő – Class Selector
 3. Azonosító kijelölő – ID Selector
 4. Stílusok csoportosítása (Csoport kijelölő) – Grouping Selector

ELEMKIJELÖLŐ – TYPE SELECTOR

- Legegyszerűbb kijelölő, típus szelektornak is szokták hívni.
- A kijelölő neve egy HTML elem neve lesz.
- Megmutatja, hogy a honlapon melyik elem(ek)re vonatkozik a stílus.
- Példa:
 - A jobb oldalon látható stílus deklaráció a weblapon az összes bekezdésre vonatkozik.
 - A bekezdések betűtípusa Arial lesz és kék színű.

Elemkijelölő neve

```
p{  
    font-family: Arial;  
    color: blue;  
}
```

CSS tulajdonság-értékpár

OSZTÁLYKIJELÖLŐ – CLASS SELECTOR

- Az így definiált stílus neve . (ponttal) kezdődik.
- A név tartalmazhat angol ABC betűit, számokat, kötőjelet és aláhúzást. Számmal nem kezdődhet a stílus neve.
 - Pl. Bootstrap is kötőjelet használ a stílus neveiben.
- A weblapon azokra az elemekre fog vonatkozni, amelyek nyitó elemében hivatkozunk a **class="stílusnév"** attribútummal.
- Az osztálykijelölő többször is alkalmazható, akár különböző HTML elemekre.

OSZTÁLYKIJELÖLŐ – CLASS SELECTOR - PÉLDA

HTML fájl

```
<p class = "kiemel">
```

Ez egy idézet, amit szeretnénk kiemelni.

```
</p>
```

...

```
<img class = "kiemel"  
src = "kep.jpg">
```

CSS fájl

Osztálykijelölő neve

```
.kiemel{  
    border: 3px solid blue;  
}
```

CSS tulajdonság-értékpár

AZONOSÍTÓKIJELÖLŐ – ID SELECTOR

- A stílus neve # (kettőskereszttel kezdődik) és a névre ugyanazok a szabályok érvényesek, mint az osztálykijelölő esetén.
- A weblapon azokra az elemekre fog vonatkozni, amelyek nyitó elemében hivatkozunk a `id="stílusnév"` attribútummal.
- Egy weblapon egy példányban alkalmazhatjuk a stílust, mert az azonosítót (id) egy weblapon csak egyszer lehet használni.
- Az azonosító kijelölőt szerkezeti elemekre, illetve űrlapelemekre ajánlott használni.
 - Olyan elemekre szoktuk használni, amikre hivatkozunk majd később JavaScriptből,
 - vagy például link esetén ugrási pontot szoktunk elhelyezni vele egyoldalas weblapok esetén.

AZONOSÍTÓKIJELÖLŐ – ID SELECTOR - PÉLDA

HTML fájl

```
<div id = "fejlec">  
  <img src = "logo.jpg">  
  <nav>  
    <a href="...">menü1</a>  
    <a href="...">menü2</a>  
    <a href="...">menü3</a>  
  </nav>  
</div>
```

CSS fájl

```
#fejlec{  
  background-color: blue;  
  height: 150px;  
  color: white;  
}
```

Azonosítókijelölő neve

CSS tulajdonság-értékpár

STÍLUSOK CSOPORTOSÍTÁSA – GROUPING SELECTOR

- Ezt a kijelölőt akkor használjuk, amikor különböző elemekre ugyanazok a tulajdonságok érvényesek.
- Az egyes stílusok nevét vesszővel kell elválasztani.

Stílusnevek felsorolása, amelyeknek közösek a formázási beállítások

```
p, h1, .kiemel{  
    padding: 15px;  
    color: white;  
}
```

CSS tulajdonság-értékpár