

# CSS

---

Ezt nagyon nem is húznám, minden fiszemfaszom formázást meg css stílust megtalálsz a [Bibliában](#).

## Hogyan húzunk be stílusokat a HTML oldalunkra?

Erre két fő lehetőségünk van, az első és a legtisztább, hogy a `<head>` tagbe behúzzunk a css fájlunkat:

```
<link rel="stylesheet" href="styles/main.css">
```

A második pedig az, hogy a html-be a `<head>` és a `<body>` tag közé berakunk egy ilyet:

```
<head>
  <title>Zack's Univershitty</title>
  <link rel="stylesheet" href="styles/main.css">
  <meta charset="UTF-8">
</head>
<style>
.center-box {
  margin: auto;
}

.center-box p {
  margin: 0;
  text-align: center;
  font-weight: bold;
}
</style>
<body>
  <div class="center-box">
    <p>Ez lesz az oldalad alapja!</p>
  </div>
</body>
```

Jól látod, csináltunk oda egy `<style></style>` tag-et, amibe elkezdünk írni a css-t, ugyan úgy mint egy külön .css fájlba. Ezt amúgy ilyen borzasztó egyszerű oldalakra szoktuk használni, ahol max 4-5 css stílus van és ezért nem akarunk új fájlt nyitni neki, vagy mondjuk van egy fain PHP-s oldalunk, és adatbázisból akarjuk behúzni a bejelentkezett paraszt egyedi hátterét az oldalra, akkor csinálunk egy style tag-et amit php-ból generálunk, a betöltött .jpg képpel. Egyébként minden mást szépen rakosgassunk el a megfelelő .css fájllokba.

## Hogyan raksz stílus a HTML-ben megadott class="valami"-re?

Megnyitod szépen a css fájlod, és így hivatkozol rá:

```
.valami {
  margin: 0;
```

```
background-color: black;
}
```

A fenti kódban ráhivatkozunk a HTML-ünkben megadott `class="valami"`-ra, és megadtuk hogy ne legyen margója, meg legyen fekete a háttérszíne.

## Hogyan raksz ID-re stílust?

Bizony, arra is lehet ugyan úgy stílust rakni mint class-okra. Amúgy dettó ugyan úgy működik mint a class-ok, csak nem "." azaz pont karakterrel használjuk, hanem:

```
#valami {
  margin: 0;
  background-color: black;
}
```

...ja, HESSTEG azaz kettőskeresztrel használjuk.

## Hogyan hivatkozol olyanra hogy: class="valami fekete"?

```
.valami.fekete {
  margin: 0;
  background-color: black;
}
```

Miértelme van ennek? Na mondjuk tegyük fel hogy te a "fekete" nevű class-t sok helyen használsz, mondjuk pár div-re. Na most te nem akarsz minden egyes div-nek külön class nevet adni, és css-be annyiszor leírni mindbe hogy legyen fekete. Ezért te css-be létrehozol egy olyat hogy:

```
.fekete {
  background-color: black;
}
.valami {
  font-size: 50px;
}
```

Aztán a html-ed így nézki:

```
<div class="valami fekete"></div>
```

```
<h2 id="title" class="fekete">Párolgó Denevér</h2>
```

Na most látod hogy két helyen használtuk a HTML-be a "fekete" nevű class-t. Így a div-nek és a h2-nek is fekete lesz a háttere. Aztán csinálhatsz olyan speckó cuccokat is hogy például te azt szeretnéd hogy a "valami fekete" két classsal ellátott cuccod valamiben eltérjen még a többitől. Így fog kinézni a css-ed:

```
.fekete {
  background-color: black;
}

.valami {
  font-size: 50px;
}

.valami.fekete {
  color: green;
}
```

Na és így a:

```
<div class="valami fekete"></div>
```

Cuccod-nak fekete lesz a háttere, 50-es betűmérete lesz a benne lévő szövegnek, és zöld színű lesz, azért mert a fenti css-ek mind illenek rá. **FONTOS:** Ha egy darab tagen több css class-ra szeretnék hivatkozni, akkor a css-ben a class-okat mindig egymás mellé írjuk, mint a **.valami.fekete**-t.

## Tag-en belüli classokra hivatkozás

Tegyük fel van neked egy ilyen HTML-ed:

```
<div class="valami fekete">
  <h2 class="text">Benti fing</h2>
</div>

<h2 class="text">Kinti fing</h2>
```

Na most te okosan ügyesen megszeretnéd stílusozni azt a div-en belüli h2-es szöveget, és kifezetten azt szeretnéd, hogy a most megírt css-ed csak erre legyen igaz. Ha most csak simán létrehozol egy:

```
.text {
  color: green;
}
```

css-t, akkor mindkettő szöveg zöld lesz, de te csak azt akarod hogy a div-en belüli legyen az. Akkor pontosítod a css-ed és explicit megmondod neki mit szeretnél:

```
.valami.fekete .text {
  color: green;
}
```

Figyeled? Tettem most egy SZÓKÖZT a classokba. Miért? Azért, mert ha szóközt teszel a classok közé, akkor azt jelenti hogy a következő class az benne lesz az előző cuccban. (VSCode-ban ha ráviszed az egered a css-be a megírt classodra, kifog adni tooltipre egy mintát hogy milyen html felépítést fog keresni akkor ez a css segítségnek). És ezt tolhatjuk tovább, ha a class="text"-es cuccon belül is van még valami, a css-ben megint szóközt írunk a text után, és írjuk a következő classunkat.

```
<div class="valami fekete">
  <h2 class="text">Benti fing<span class="subtext">(nagyon büdi)</span>
</h2>
</div>

<h2 class="text">Kinti fing</h2>
```

Na most van egy olyanunk hogy div-ben a h2 és abban egy span, és mi most közvetlen a span-t akarjuk stílusozni, akkor a css fájlunk így fog kinézni:

```
.valami.fekete .text {
  color: green;
}

.valami.fekete .text .subtext {
  color: red;
  font-weight: bold;
}
```

Ezzel a nagy h2-es szöveget zölddé tettük, a benne lévő zárójeles kis span szöveget pedig pirossá és vastagon kiemeltre.

## CSS pontok és erősségek

Huha na ezt se tudtam jobban megcímezni, lol. Szóval lesznek olyan esetek amikor felül szeretnél írni egy meglévő stílust valami egyszeri eset miatt és ezért nem akarsz még egy külön rakás css-t írni. Folytassuk az előző cuccunkat:

HTML maradjon ez:

```
<div class="valami fekete">
  <h2 class="text">Benti fing<span class="subtext">(nagyon büdi)</span>
</h2>
</div>

<h2 class="text">Kinti fing</h2>
```

És most legyen ez a css:

```
.valami .text {
  color: green;
}

.valami .text .subtext {
  color: red;
  font-weight: bold;
}
```

Ha most ezt megnézzük, ugyan azt az eredményt fogjuk kapni mint az előbb. De mivan akkor, ha kibővítem egy ilyenre a HTML-em:

```
<div class="valami">
  <h2 class="text">Benti fing<span class="subtext">(nagyon büdi)</span>
</h2>
</div>

<div class="valami fekete">
  <h2 class="text">Fekete fing</h2>
</div>

<h2 class="text">Kinti fing</h2>
```

Ha most ezt így megnézzük, mind a kettő div-ben lévő h2-es szöveg zöld lesz. Na most látjuk a másiknál ott van egy külön class hogy "fekete". Annak a használatával szinezük át a másikat feketére, így:

```
.valami .text {
  color: green;
}

.valami.fekete .text {
  color: black;
}

.valami .text .subtext {
  color: red;
  font-weight: bold;
}
```

Na és ha most megnézzitek, az egyik zöld, másik fekete. Miért is? Mint látod, a másikon van egy extra class. Na most ezt úgy számold fejben, hogy CSS-ben van egy ilyen pontozás szerű rendszer. 1db class name 1 pontot ér, példa: `class="valami"` ez egy 1 pont. `class="valami fekete"` ez már 2 pont, you get it. Na hogy kicsit jobban megértsd módosítom a HTML-t:

```
<div class="container">
  <div class="valami">
```

```
        <h2 class="text">Benti fing<span class="subtext">(nagyon büdi)
</span></h2>
    </div>

    <div class="valami fekete">
        <h2 class="text">Fekete fing</h2>
    </div>
</div>

<h2 class="text">Kinti fing</h2>
```

Most beraktam a két divet még egy "container" class-os divbe. A CSS-ünk most ez:

```
.valami .text {
    color: green;
}

.valami.fekete .text {
    color: black;
}

.valami .text .subtext {
    color: red;
    font-weight: bold;
}
```

Na most hogy csinálom meg, hogy felül írom a "valami fekete" classos szöveg színét úgy, hogy már megadtam fentebb? Egyszerű, írok egy ilyen CSS-t:

```
.valami .text {
    color: green;
}

.valami.fekete .text {
    color: black;
}

.valami .text .subtext {
    color: red;
    font-weight: bold;
}

.container .valami.fekete .text {
    color: blue;
}
```

Látod? Fogtam most a "container" class-tól kezdtem el meghatározni mit szeretnél css-elni, utána rámentem a "valami fekete" class-os div-ben lévő "text" classos szövegre, így felülírtam minden eddigi css-

t ezzel.

**KURVA FONTOS:** Neten látsz olyant hogy:

```
.valami .text {  
    color: green !IMPORTANT;  
}
```

Na most oda cseszel valami mögé egy !IMPORTANT-t jelzőt, az instant a legfelsőbbrendű CSS lesz és mindent felülír, még akkor is ha valami 15 classos dolgod van és itt csak 1 class-ba adtad meg. De ez baromság, undormány, átláthatatlanná teszi a css-t, csak szopni fogsz vele később, és én vagy a következő munkáltatód fogja elhegedülni a segged ezért. Ha valami css-t felül akarsz írni valamivel, azt tedd a fentebb látható módon, hogy több classsal adod meg, vagy kívülebbről kezdted a deklarálást.

## Inline style

CSS-en kívül egyből a HTML-be is megtudod stílusozni az elemet. De egyébként ez se ajánlott, ezt max akkor használjuk amikor pl PHP-ből generálsz HTML-t és mondjuk minden div-be külön background-image-t akarsz rakni vagy ilyesmi, de mi kézzel fixen sose csináljunk ilyet, adjuk meg css-ből. Példa:

```
<div style="background-color: black;">  
    <h2 style="color: red;">Fekete fing</h2>  
</div>
```

Na most itt beszíneztem a div-et fekete háttérűre, a szöveget meg zöldre. Ezt a `style="ideacssedírod"` html attribútummal éred el, szal csak simán bele írod azokat a css stílusokat amiket a CSS-be is írnál. Ha ezt használod, ez felsőbb rendű lesz a css classoknál. Szal ha te előtte css-be megadtad hogy legyen kék a háttérbe, de amúgy ott egy "style" attribútumba hogy legyen fekete, akkor fekete lesz.

## Tag-ek stílusozása

Na ne hogy azt hidd, csak úgy tudsz stílusozni, hogy classokat aggatsz mindenre, mivan akkor ha neked van 200 `<p>`-t és te mindegyikre akarsz egyedi stílust? Mindegyikre írsz egy class-t? AHHH, a gondolatától is lerohad az ujjam.

```
<div class="valami">  
    <h2>Benti fing<span>(nagyon büdi)</span></h2>  
</div>
```

Van ilyen HTML kódod, és én azt mondom stílusozd meg a "valami" class-on belüli h2-n belüli span-t. Mit csinálsz? Ezt:

```
.valami h2 span {  
    color: red;
```

```
}
```

Így most megkerested a "valami" class-on belüli h2-n belüli span-t! Ha megnézed, én itt most a css-be se pontot, se #-t nem raktam a nevek elé, hanem csak közvetlen beírtam a tag nevét hogy azt keresem. **Na most ezzel érdemes azért vigyázni**, mert ez a cula úgy működik, hogy mondjuk van egy ilyen HTML-ed:

```
<div class="valami">
  <span>Random szöveg</span>
  <h2>Benti fing<span>(nagyon büdi)</span></h2>
  <p>Hehehe <span>Ez meg egy P-ben belüli random szöveg</span></p>
</div>
```

Na most én azt szeretném ha csak a "Random szöveg" span-t stíluzosd meg, logikusan elkezdted azt írni a css-be hogy:

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

És akkor elhitted hogy majd a div-en belüli közvetlen span lesz piros, a többi nem... Hát jön a lefagyás amikor megnyitod a böngészőt és látod hogy a "valami" class-os div-en belüli **ÖSSZES** span piros lett! De mi a bánatért? Azért, mert ez úgy működik, hogy ha csak sima tag-et keresel és azt designolod, akkor a megadott tag-en belüli összes megadott tag-et megfogja találni és ráhúzza azt a stílust! Nem csak azt, ami logikusan jönne, nem, ő belemegy a többi tagbe is, és ha azonbelül van mondjuk az itt megadott span, ráfogja húzni a piros színt. Úgy hogy ezzel csak óvatosan, van haszna, de ritkán használjuk.