

## Stíluslapok

A *stíluslapok* (CSS: Cascading Style Sheets – egymásba ágyazott stíluslapok) segítségével a weboldal szerkezete, tartalma *különválasztható* annak formájától.

„Tisztán” csak a HTML nyelv lehetőségeit használva, az oldal végső képét meghatározó utasításokat (betűtípusok, színek, hátterek, méretek, stb.) a HTML kódon belül kell leírni; legtöbbször ismétlődően. A CSS használatával ezeket az információkat áthelyezhetjük a stíluslapra, így egy sokkal egyszerűbb és kisebb HTML kódot kapunk. (Ráadásul: mivel a webböngészők gyakran tárolják a CSS stíluslapokat a gyorsítótárban, a hálózati forgalom is jelentősen csökkenhet.)

Nézzünk egy példát. Mint tudjuk, a `<h2> </h2>` HTML-elem megadja, hogy a benne lévő szöveg második szintű cím. Tegyük fel, hogy egy középre igazított, dőltbetűs, zöld, Times New Roman típusú címsort kívánunk használni. Eddigi ismereteinkkel (stílusok nélkül) ezt így lehetett megoldani:

```
<h2 align="center"><font color="green" size="+4" face="Times New Roman", serif><em>Itt a
szöveg</em></font></h2>
```

és ezt a címsor minden egyes használatakor meg kellett ismételni a HTML-kódban.

A stíluslap használatával a `h2` elem formai tulajdonságait csak egyszer kell leírni (nem is feltétlenül az oldal szövegébe), valahogy így:

```
h2 {
    text-align: center;
    color: green;
    font-size: large;
    font-family: "Times New Roman", serif;
    font-style: italic;
}
```

Az oldal kódjába pedig csak ennyi kerül:

```
<h2>Itt a szöveg</h2>
```

Ebből a kis példából talán már látszanak a stíluslapok (CSS) használatának legfontosabb előnyei:

- több lap vagy akár egy teljes webhely stílusait egy helyen lehet tárolni, így az gyorsan és könnyen frissíthető, átalakítható;
- a HTML-dokumentum mérete csökken, áttekinthetősége viszont javul, mivel nem tartalmazza a formázási információkat;
- különböző felhasználókhoz különböző stílusokat lehet rendelni: például könnyebben olvasható, nyomtatható stílusok.

Nézzük tehát, hogyan használhatók a stíluslapok.

Minden CSS utasításnak két része van, a *szelektor*, és a *deklaráció*.

- a *szelektor* tartalmazza a formázandó HTML-elem(ek) megnevezését;
- a *deklaráció* írja le a szelektorban megadott HTML-elemek formázását. Ennek általában több része van (paraméter, érték), ugyanis meg kell adni, hogy mit akarunk a szelektorban formázni (pl. szín), utána pedig azt, hogy hogyan akarjuk formázni (pl. kék).

Például:

```
P {font-family: Verdana, Arial, sans-serif,}
```

Ebben a példában a `P` lesz a *szelektor* (a HTML-ben a bekezdés jele). A *deklarációt* mindig kapcsos-zárójelek közé tesszük. Itt a deklaráció paramétere a `font-family` (betűtípus), és az értéke a `Verdana, Arial, sans-serif` felsorolás. Annak, hogy felsorolást adtunk meg, az oka a következő: a weblap szövegét ugyan a saját gépünkön írjuk meg, azt a webszerver tárolja, de a *felhasználó* gépén futó böngésző jeleníti meg. Ha azon a gépen telepítve van a Verdana betűtípus, akkor a bekezdés azzal jelenik meg. De ha nincs Verdana, akkor Arial-lal, és ha az sincs, akkor tetszőleges talpnélküli (sans-serif) típussal fog megjelenni.

Hogyan lehet a fenti módon megalkotott stílusokat a HTML dokumentumhoz kapcsolni? A csatolásra többféle módszer is van, a legpraktikusabb talán az, hogy a stílusokat például egy *stilus.css* nevű szövegfájlba leírjuk (ezt hívják *külső stíluslapnak*), és erre a HTML dokumentum *fejrésében* az alábbi utasítással hivatkozunk:

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

Ha webhelyünk valamennyi lapjába beillesztjük a fenti sort, akkor az egész webhely küllemének *egyszerre* történő megváltoztatásához elegendő a *stilus.css* fájlt módosítani.

A CSS-ben lehetőség van a deklarációk és szelektorok csoportosítására, mellyekkel csökkenthetjük stíluslapjaink méretét és így azt sokkal átláthatóbbá tehetjük.

Például a szelektorok vesszővel elválasztott listába csoportosíthatók:

```
H1, H2, H3 {font-family: verdana, sans-serif}
```

Hasonló módon a deklarációk is csoportosíthatók:

```
H1
{
  font-family: helvetica, sans-serif;
  font-size: 120%;
  font-style: bold;
}
```

A stílusok további finomítási lehetőségét jelenti az *osztályok* (class) használata. A HTML elemeket lehetőségünk van osztályokba sorolni. Ekkor az adott osztályba sorolni kívánt elemhez egy `class` tulajdonságot rendelünk hozzá, amelynek az értéke az osztálynév lesz. A stíluslapban ehhez az osztályhoz határozhatunk meg megjelenést. Ha például szeretnénk kiemelni egyes bekezdéseket a szövegünkéből, félkövér betűkkel megjelenítve azokat, akkor a következőképpen járhatunk el a stílusdeklarációban:

```
P {font-family: Verdana, Arial sans-serif;}
.kover {font-style: bold;}
```

A deklaráció használata és hatása:

```
<body>
  <p class="kover">Ez a bekezdés félkövér.</p>
  <p>Ez már nem félkövér, mivel nincs hozzárendelve az osztály.</p>
</body>
```

Aki látogatott már weboldalakat, tudja, hogy a linkek más-más színnel jelennek meg, ha már voltunk az általuk mutatott oldalon, illetve ha még nem. Hogyan tudjuk ezt stílusokkal megoldani?

```
A:link {color: blue}           /* nem látogatott link */
A:visited {color: red}        /* látogatott link */
A:active {color: lime}        /* aktív link */
```

(A példából az is látszik, hogyan tudunk megjegyzéseket elhelyezni a CSS fájlban.)

Természetesen a CSS ennél sokkal többet „tud”. Kiváló lehetőség a tetszésünket megnyerő weblapok lementése (pl. IE-vel, vagy Firefox-szal) gépünkre: az ekkor keletkező *weblap\_elemei* könyvtárban általában megtaláljuk az oldal kinézetéért felelős *css* fájlt. Lehet tanulmányozni...

**Egy példa:**

```
HTML
    {
    background-color: #FFFFFF;
    }

BODY
    {
    background-color: #FFFFFF;
    margin : 0px 0px 0px 0px;
    border : 0px 0px 0px 0px;
    padding : 4px 4px 4px 4px;
    }

TD.balmenu
    {
    margin : 0px 0px 0px 0px;
    background-color : #EEEEDD;
    padding : 2px 4px 5px 4px;
    font-family : Tahoma,Arial,sans-serif;
    vertical-align : top;
    text-align : left;
    font-size : 12px;
    width : 18%;
    height : 100%;
    }

TD.foablak
    {
    padding : 2px 8px 15px 8px;
    color : #000000;
    background-color: #ECEBE5;
    font-family : Tahoma,Arial, sans-serif;
    vertical-align : top;
    width : 81%;
    text-align : justify;
    font-size : 11px;
    }

P
    {
    color : #000000;
    font-family : Tahoma,Arial, sans-serif;
    text-align : justify;
    font-size : 11px;
    }

.center
    {
    color : #000000;
    font-size : 11px;
    text-align : center;
    }

.menutetel
    {
    padding-top : 0px;
    text-align : left;
    font-size : 11px;
    }

.menucim
    {
    color : #000000;
    text-align : left;
    font-family :Tahoma,Arial,sans-serif;
    font-weight : bold;
    font-size : 11px;
    }

H1
    {
    color : #000077;
    text-decoration : none;
    font-family : Verdana,sans-serif;
    font-weight : bold;
    font-size : 20px;
    font-variant: small-caps;
    }
```

```
H2
    {
    color : #000077;
    font-family : Verdana;
    font-weight : bold;
    font-size : 13px;
    }

H3
    {
    color : #000077;
    font-family : Verdana;
    font-weight : bold;
    font-size : 11px;
    }

A
    {
    color : blue;
    text-decoration : none;
    }

: hover
    {
    color : blue;
    text-decoration : underline;
    background-color : #d0d0d0;
    }

: active
    {
    color : red;
    text-decoration : none;
    }

: visited
    {
    color : #FF0080;
    text-decoration : none;
    }

UL
    {
    color : #000000;
    font-family : Verdana,Tahoma,Arial;
    font-size : 13px;
    }

.karika
    {
    display : inline;
    list-style-position : inside;
    list-style-type: disc;
    margin-top : 0px;
    text-align : left;
    color : #005511;
    font-family : Verdana;
    font-size : 11px;
    }

.szam
    {
    display : inline;
    list-style-position : inside;
    list-style-type: decimal;
    margin-top : 0px;
    text-align : left;
    color : #005511;
    font-family : Verdana;
    font-size : 11px;
    }
```