

```
a:link {color: rgb(0,0,255)}
a:visited {color: rgb(127,127,127)}
a:hover {color: rgb(255,255,0)}
```

A példa szerint a linkek színe zöld, a látogatott linkek színe szürke lesz, és ha egy link felett megállunk az egérrel, akkor az sárgává válik.

Fontos tudni, hogy a sorok rendje lényeges. Az *a:hover* sornak feltétlen az *a:visited* és *a:link* után kell állnia, mert egyébként nem érvényesül a hatása.

Saját bekezdésstílus

Vizsgáljuk meg a következő tartalmú stíluslapot.

```
p {font-size: 28pt; color: #00FF00}
p.tanar {color: #FF0000}
p.diak {color: #0000FF}
```

Az első sor rendelkezik arról, hogy a bekezdések 28pont méretű karakterekből álljanak és zöldek legyenek. Ha csak ennyit tartalmazna a stíluslap, akkor a lap minden bekezdése ilyen formátumú lenne. Az osztályaink felirat 28 pont méretű és zöld lesz. A következő két sor lényegében egy-egy saját bekezdésstílust ad meg. A példa szerint egy tanar és egy diak nevű stílust definiáltunk, ami csak a megfelelő bekezdés színéről rendelkezik. A *p.tanar* stílusú bekezdés színe a példa szerint piros lesz, de felmerül a kérdés, hogy mi lesz a karakterméret. A jelölésből kiindulva tekinthetünk úgy a tanar és a diak nevű stílusainkra, mintha azok a *p* stílusnak lennének a gyerekei. Ez annyit fog eredményezni, hogy a két saját stílusunk karaktermérete a szülőtől fog öröklődni, azaz 28pt méretű lesz.

Nézzük meg, hogyan jelöljük a html kódban, hogy egy szöveget saját stílussal szeretnénk formázni.

```
<p>Osztályaink</p>
<p class="tanar">Bán Anikó</p>
<p class="diak">Holczer Viktor</p>
<p > &nbsp;</p>
<p class="tanar">Dobos Zsuzsa</p>
<p class="diak">
Balázs Gábor <br>
Dékány József </p>
```

A két osztályfőnök nevénél a *p.tanar* stílust szeretnénk alkalmazni a stíluslapról. Ezt a `<p>` elem `class` paraméterének megadásával tehetjük meg a lap html kódjában. A diákok névsoránál hasonlóan jelezhetjük, hogy a diak stílust szeretnénk használni.

Álelemek

Egy érdekes lehetőség, hogy a stíluslapokon az úgynevezett álelemek segítségével a tartalom olyan részeire is meghatározhatunk formázást, amelyekre a html kódban nem tudunk hivatkozni. Például a stíluslapokon a *first-letter* álelem segítségével a szöveg első betűjére hivatkozhatunk. Egészítsük ki az előző példa stíluslapját a következő sorral.

```
p:first-letter {font-size: 50pt}
```

Ez azt fogja eredményezni, hogy minden bekezdés első betűje 50 pont méretű lesz. Fontos, hogy ha az előző példa szerinti html kódú az oldalunk, akkor a Dékány név első betűjének kivételével minden név első betűje 50 pont méretű lesz. Ennek oka az, hogy az utolsó két nevet nem önálló bekezdésbe tettük, hanem csak sortőrést helyeztünk közéjük.

Érdemes megemlíteni, hogy valójában a linkek látogatottsági állapottól függő színezése is az álelemek segítségével valósul meg.

Eseménykezelők

Egy egyszerű weblapon a hivatkozásra való kattintás után töltődik be az újabb lap. Programozási eszközökkel készíthetünk például legördülő menüt, és az egyes menüpontokhoz rendelhetünk különböző feladatokat. A html nyelv viszont beépítetten tartalmaz néhány lehetőséget különböző események figyelésére. Figyelhetjük például, hogy egérrel egy elem fölé értünk, esetleg egy elemre duplán kattintottunk, és megadhatjuk, hogy mi történjen az egyes események bekövetkezése esetén. A legtöbb html elem támogatja az eseménykezelők alkalmazását. Például megoldhatjuk, hogy egy `<p>` elem hatókörébe eső szöveg érzékeny legyen az egérrel való kattintásra. Pontosabb részletekért olvassuk el a **Programok a weblapon** című fejezet **Eseménykezelők** alfejezetét.

Feladatok

37. Milyen színű lesz az alma és a körte című szó a következő kódrészlet szerint?

```
<p> <font color="#FF0000"> Alma <font color="#00FF00"> Körte </font> </font>
</p>
```

38. Mit mondhatunk a minta szerinti kódrészletről?

```
<p> <font color="#00FF00"> Csülök <b> </font> Sonka </b> </p>
```