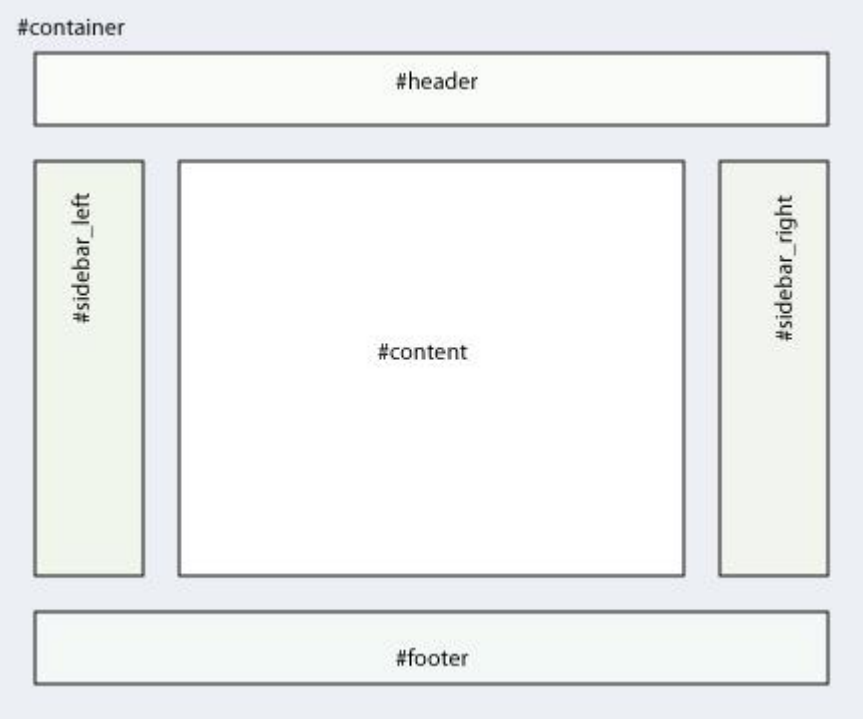
CSS layout negatív margókkal

2007-05-01

Célunk, hogy három fix szélességű oszlopból álló weboldalt készítsünk, ahol a tartalmat CSS stíluslapokkal formázzuk. A webtartalom megformázásához "sima" float-olást fogunk majd használni, de mindenekelőtt vessünk egy pillantást a következő ábrára, amely megmutatja, hogy a HTML oldalunkon mely DIV elemek lesznek majd használatosak.

float:left és float:right, úsztatás egyszerűen



Mint láthatjuk a #container elem tartalmazza az összes többi DIV elemet, amelyet majd el akarunk rendezni.
Először töltsd le a HTML kódot, hogy máris nekiláthassunk az oldal elrendezésének.

Letöltés: pure_float.html

Ha színekkel különböztetjük meg az oldal elemeit, akkor szerintem jobban nyomon követhetjük a formázás menetét.

```
#container {
        background:#EAEDF1;
}
#header {
        background:#F7FBF6;
}
#sidebar_left {
        background:#F0F3EA;
}
```

```
#sidebar_right {
        background:#F0F2EC;
}
#content {
        background:#FFFFFF;
}
#footer {
        background:#F1F7F4;
        color:#999999;
}
```

Ezután "pofásítsuk" oldalunkat néhány egyszerű smink-szerűséggel:

```
p, h1, h2, h3, h5 {
        margin:0;
        padding:10px;
}

#header h1, #footer h5 {
        text-align:center;
}

#sidebar_left ul {
        list-style:none;
}
```

És most jön a lényeg: az oldal elrendezése. Elsőként a #container elemnek adjunk szélességet, majd vigyük a lap közepére.

```
#container {
        width:700px;
        margin:0 auto;
}
```

Ezután "vesszük" az oldalsávokat és azoknak is szélesség-méretet adunk, majd float-oljuk a megfelelő irányokba.

```
#sidebar_left {
        width:150px;
        float:left;
}
#sidebar_right {
        width:150px;
        float:right;
}
```

Végezetül a középső tartalommező bal és jobb szélét is beigazítjuk.

```
#content {
        margin:0 150px;
}
```

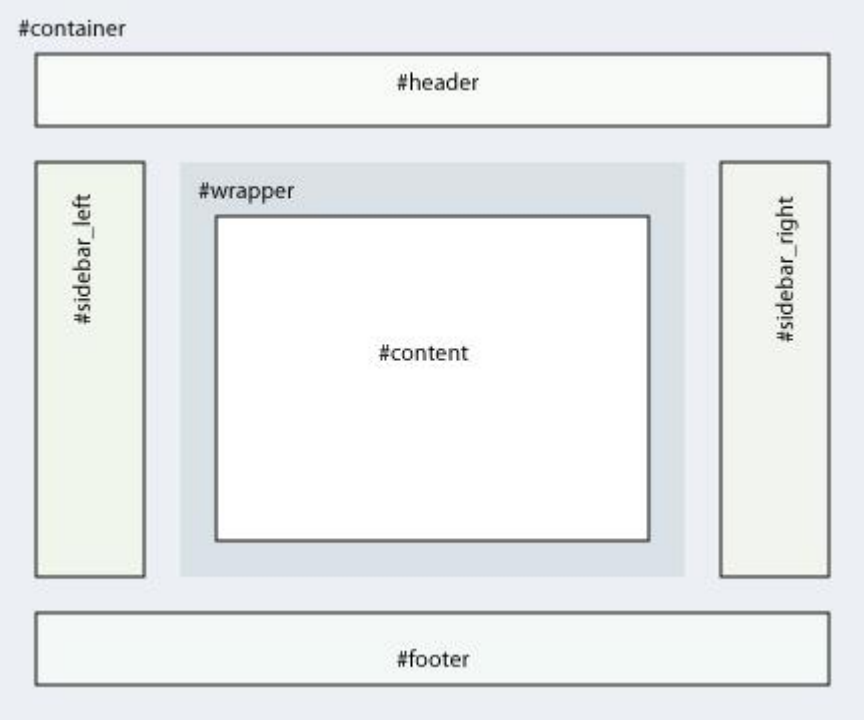
Vessünk még egy pillantást a CSS fájlunkra (pure_float_style.css), és ha úgy találjuk, hogy megfelelően kapcsolódik a kezdeti HTML dokumentumhoz, akkor minden OK. Az eredmény megtekintéséhez katt ide: mintaoldal kicsit több tartalommal (pure_float_with_more_content.html)

Továbbra is úsztatás egyszerűen, de célként a tartalom az oldalsávokhoz képest hamarabb töltődjön be. Megoldás: a negatív margók használata!

Mire is gondolok? Arra, hogy a fenti példa esetében konkrétan az oldalsávok hamarabb betöltődnek, mint a tartalom. Nem szeretném túl filozofálni a kérdést, de azt gondolom, hogy az oldal legfontosabb része az maga a tartalom. Ezért logikus, ha azt szeretnénk, hogy a tartalom előbb töltődjön be, mint például az oldalsávok.

Hogyan rendezhetjük el az oldalunkat CSS-el akkor, ha a tartalom, azaz a #container "jön" előbb és majd csak utána következnek az oldalsávok DIV elemei?

Mielőtt letöltenénk a következő megoldás kódjait, nézzük meg az itt következő képen, hogy vajon ugyanazok a DIV elemek szerepelnek-e, mint az előző példában.



Megállapíthatjuk, hogy most a #wrapper elem tartalmazza a #content elemet. De az igazi különbséget nézzük meg a HTML kódban (negative_margin.html), de töltsük le mellé a CSS fájlt is (negative_margin_style.css).

Mint láthatjuk a most letöltött HTML kódban a tartalom "megelőzi" az oldalsávokat. Terjedelmesebb tartalomnál lehet fontos, hogy ez töltődjön be a böngészőbe előbb, mint az oldalsáv; de lássuk, hogyan tudjuk oldalunk részeit CSS-el elrendezni.

A "layouts"- szal kommentezett részt nézzük, mert minden más megegyezik az előzővel. Elsőként adjunk szélességet a #container elemnek, majd vigyük középre a már ismert módon. Ezután float-oljuk a #wrapper elemet balra, majd ennek is adjunk szélességet.

```
#wrapper {
        float:left;
        width:100%;
}
```

Most “vegyük” a tartalmat és állítsuk be a szélességét.

```
#content {
        margin:0 150px;
}
```

És most következnek a negatív margók használata.

```
#sidebar_left {
        float:left;
        width:150px;
        margin-left:-700px;
}
#sidebar_right {
        float:left;
        width:150px;
        margin-left:-150px;
}
```

Majd zárjuk le a float-olást, ill. az úsztatást a #footer elem segítségével.

```
#footer {
        clear:left;
        width:100%;
}
```

Ennyi. A kész eredményt megtekintheted itt több tartalommal: mintaoldal több tartalommal (és a hozzá tartozó css fájl is letölthető)

A két HTML oldalt, amit a fenti két példában készítettünk összehasonlítottam FireFox-ban, IE6-ban, Opera-ban és a Netscape böngészőjében és az eredmény az, hogy küllemre megegyeznek, viszont a negatív margók esetében a tartalom hamarabb betöltődik a böngészőbe, mint az oldalsávok. Ez pedig sok esetben egyszerűen hasznos dolog.
Jó gyakorlást és sok hasznos CSS megoldást. Letöltheted az összes mintaoldalt egy ZIP fájlba csomagolva (8kb) – csslayoutnegativemargin.zip