

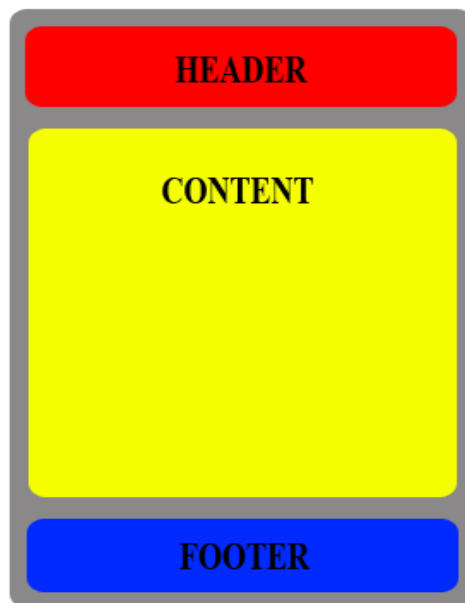
CSS

CSS layout.

Правилното подреждане на елементите в един сайт без таблица изисква не само практика и добро познаване възможностите на CSS , но и познаване на проблемите с различните browser-и и как да се справим с тях. Особено много проблеми създава Internet Explorer 6, IEv7 интерпретира css кода почти като ie6, но с малки разлики. Да неговорим за мистериозния рапър звезда (елемент който се бележи с * и обвива всички останали елементи), който за щастие изчезва с появата на ie7.

Логическо разделяне на компонентите – когато разпределяме компонентите трябва да следваме следната логика:

1. Поставяме всички компоненти в един общ- най-често <div> примерно с id="main", "wrapper", "body" името няма значение за това как ще изглежда div-а
2. Разделяне на същността на сайта на три основни части-хедър, контент, футър
пример:



като код:

```
<div id="outer-wrapper">
```

```
<div id="header">...</div>
```

```
<div id="content">...</div>
```

```
<div id="footer">...</div>
```

```
</div> <!-- END outer-wrapper -->
```

```
<!-- Сив цвят -->
```

```
<!--Червен цвят -->
```

```
<!-- Жълт цвят -->
```

```
<!-- Син цвят -->
```

3. Поставяме логото, слоган (ако има) и главната навигация в header
Поставяме съдържанието и банерите (ако има) в content
Поставяме фирмата дизайнер, лицензи, малко навигация и/или котва за горната

част на страницата, ако е прекалено дълга.

Създаване и подреждане на колони

float. Чрез float казваме на даден елемент как да „изплува“, т.е да се показва отдясно или отляво. Работи за inline и за block елементи. С float, на inline елементите може да се задава височината/дължината, но родителите им ги приемат като нямащи височина, затова добро решение е да се сложи празен елемент (tag) с clear.

*IEв6 удвоява margin-ите, когато съвпадат с посоката на float-а. Това се нарича double margin bug. Ако се добави display:inline; към съответния елемент, margin-а не се удвоява.

Ако имаме повече от един елемент със float, те се нареждат последователно, ако преди елемент с float имаме такъв без float, то елемента с float се показва преди този който няма. Когато на един ред няма достатъчно място на един ред за всички float-нати елементи, те се преминават на нов ред.

Clear. Използва се когато не искаме да „изплуват“ повече елементи в дадена посока, или в двете едновременно. Може да приема стойности left, right, both.

Left/Right-изчиства всички елементи отляво/отдясно, могат да изплуват повече елементи наляво/надясно, ако преди тях (на същият ред) има елемент с clear:left;

Пример: css2layout.html

Transperancy(прозрачност).

Много е удобно да се ползва прозрачни елементи, освен ако... трябва да използвате IEв6, За съжаление има хора който все още го използват, и използването и е трудно единствено зареди IEв6.

*Съвет- Използвайте прозрачността само когато е найстина необходима и не е възможно да наместите картинките все едно са една цяла.

Internet Explorer 6 поддържа само безцветен цвят на *.png и *.gif, но не поддържа различни цветове със прозрачност. Механизма на Microsoft за рендериране на цветове с прозрачност в IEв6 е чрез използване на филтри. Филтрите са доста навътре механизма за рендериране. За щастие се единственото необходимо е да се знае как могат да се използват чрез css и/или javascript. На мястото на **path_name** стой името на картинката за която трябва да бъде поддържана прозрачността.

```
Tagname { filter:progid:DXImageTransform.Microsoft.AlphaImageLoader  
(src='path_name',sizingMethod='scale'); }
```

sizingMethod-а може да е и 'crop'. Това не решава почти никакви проблеми, понякога не работи. Ако трябва да се поставя картинки с прозрачност една върху друга, това ще работи само за едната. Също картинки за който се използва прозрачност не могат да се използват за фон който да се повтаря. Ако има вече зададена картинка като фон и тя има прозрачност трябва да се зададе **background-image:none;** за да има ефект използването на филтър.

В интернет има доста скриптове използването на javascript за използване на филтри, но идеята е следната. Обикалят се всички картинки с разширение png и ги се заместват с филтер със src= път до картинката, естествено прила се пак със css.

!important

Нека да имаме файл index.html в който на елемент с id="test" е зададен inline стил ,примерно color:#fff. Ако нямаме достъп до този файл и не можем да променим стила, но имаме файл test.css в който можем да определяме стиловете на всички елементи от index.html би било невъзможно да определим цвета на id="test" ако нямахме правилото за !important. За да променим стил за даден елемент, с дефиниран inline стил, от друго място с по-малък приоритет е необходимо само да сложим !important към дефиницията на даденото правило за стил.

Пример:

```
#test {color:#000 !important; }
```