

9) </FRAMESET>

10) </HTML>

В 5-й строке вместо тега <body> используется тег FRAMESET. Параметр ROWS указывает на то, что страница должна быть разбита на горизонтальные фреймы. Если же надо было разбить на горизонтальные фреймы, то надо указать COLS. В 5-й же строке указано, что фреймов должно быть 3 – на 1-й и 2-й отводится 45% высоты экрана, на третий – все, что осталось.

В 6-й, 7-й, 8-й строках непосредственно подключается содержимое фреймов, которое берется из файлов B611.HTM, B612.HTM, B613.HTM.

В настоящее время фреймы практически не используются, так как более удобным является использование слоев. Тем не менее активно используются плавающие фреймы, для задания которых используется тэг <IFRAME>. Плавающие фреймы поддерживаются не всеми браузерами. Плавающий фрейм можно расположить в любом месте страницы, для этого надо указать код

```
<iframe src="111.html" name="iframe1" width="300" height="300"
align="left"></iframe>
```

Здесь будет подключен файл 111.html.

2. CSS

Лекция 3. Каскадные таблицы стилей

3.1. Синтаксис CSS

Определение. Таблицы стилей (или каскадные таблицы стилей, CSS) – это описание правил, задающих параметры представления отдельных элементов на языке HTML

CSS появились одновременно с HTML 4.0 (Dynamic HTML). Сам термин «каскадные таблицы стилей» был предложен в 1994 году. Все объявления CSS называются *селекторами*, записываются в фигурных скобках.

Примеры использования синтаксиса.

1. Для того чтобы все заголовки первого уровня (h1) были зеленого цвета с размером шрифта 15 пикселей, надо записать

```
h1 {color: green; size: 15px;}
```

Свойства разделяются точкой с запятой

2. Одно свойство можно присвоить нескольким тэгам

```
h1, h2, h3 {color: green; size: 15px;}
```

3. Свойство наследования. Если по h3 текст должен быть зеленым, то и между em он тоже будет желтым.

```
<h3> Часть 4.<em>Очень большая</em> и хорошая</h3>
```

4. Контекстные селекторы. Если необходимо, чтобы текст между тегами em, которые между h3, был зеленым, а в других случаях каким-то другим, то надо задать

```
h3 em {color:green}
```

5. Использование классов.

Определили h1:

```
h1 {color: green; size: 15px;}
```

Чтобы некоторые заголовки были красными, необходимо задать

```
h2.red {color:red}
```

Использование:

```
<h2 class=red>Это красный заголовок</h2>
```

Замечание. В таблице стилей можно указать просто без указания родительского тэга:

```
.red {color:red}
```

6. Можно использовать маску * для задания всех селекторов

7. Еще один символ маски >. Задаёт свойства для определенных подэлементов

```
ol>li {color:green}
```

8. Для определения элемента в DHTML используется id. Поэтому можно задавать свойства через идентификаторы

```
#cont {color:yellow}
```

И когда встретится идентификатор `cont`, то текст этого элемента будет желтым.

9. Для создания прямоугольной области используется тег `<div>`.

10. Для создания строчной области используется тег ``.

11. Использование псевдоклассов. Динамические псевдо-классы: `:hover`, `:active`, и `:focus`.

Пример использования псевдо-классов. Указав следующие описание стилей:

```
a:hover { background: yellow }
```

получим, что при наведении курсора на ссылку, область под ней станет желтой.

3.2. Размещение каскадных таблиц

3.2.1. Свойства можно определять непосредственно с тегом. Например, вставка следующего тега сделает текст красным на синем фоне

```
<span style="color:red; background:blue; font: bold 1.8em Arial ">красный на синем</span>
```

3.2.2. Таблицу можно разместить между тегами `<style>` и `</style>`. Например,

```
<head>
<style>
a { color: #33ADDB;
    background-color: inherit;}
a:hover { color: #575757;
    background-color: inherit;}
h1 { font: bold 1.8em Arial, Sans-Serif;
    letter-spacing: -1px;
    margin: 0;
    padding: 0;}
h1 a { text-decoration: none;}
</style>
</head>
```

И теперь гиперссылки и заголовки первого уровня будут определяться в соответствии с указанными правилами.

3.2.3. Таблицу стилей можно разместить в отдельном файле. А подключение затем выполняется так:

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

3.3. Верстка

Наиболее популярным является деление макетов по ширине и количеству колонок. Выделяют следующие типы макетов², связанных с шириной:

- фиксированные;
- резиновые;
- эластичные;
- адаптивные;
- комбинированные.

Фиксированный макет располагается по центру окна браузера, а его ширина ограничивается заданными размерами в пикселах.

При создании **резинового макета** задается в процентах ширина колонок таким образом, что макет занимает всю свободную ширину окна браузера.

Эластичный макет по своему виду может не отличаться от фиксированного или резинового макета. Размер элементов задаётся в em, привязанных к размеру шрифта. Значение em можно использовать не для всех элементов, оставляя ширину некоторых фиксированной.

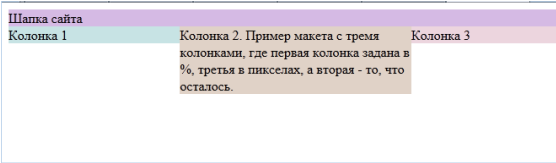
Адаптивный макет подстраивается под разрешение монитора и окна браузера, меняя при необходимости ширину макета, число колонок, размеры изображений и текста. Для этого заготавливается несколько стилевых правил или файлов под разный диапазон разрешений, выбор правил происходит через скрипты или CSS3, которые и определяют нужную для этого информацию о пользователе.

² Более подробно о верстке можно прочесть на сайте <http://htmlbook.ru>.

Комбинированный макет предполагает использование разной ширины для отдельных частей страницы, например, шапку и подвал делают резиновыми, а контент фиксированным.

Пример макета с тремя колонками, где первая колонка задана в %, третья в пикселах, а вторая – то, что осталось.

```
<html>
<head>
<style>
.header { background: #D5BAE4; }
.layout { position: relative; }
.layout DIV { position: absolute; }
.col1 { background: #C7E3E4; width: 30%; }
.col2 { background: #E0D2C7; left: 30%; right: 200px; }
.col3 { background: #ECD5DE; right: 0; width: 200px; }
</style>
<title>Три колонки</title>
</head>
<body>
<div class="header">Шапка сайта</div>
<div class="layout">
<div class="col1">Колонка 1</div>
<div class="col2">Колонка 2. Пример макета с тремя колонками, где
первая колонка задана в %, третья в пикселах, а вторая - то, что
осталось.</div>
<div class="col3">Колонка 3</div>
</div>
</body>
</html>
```



3.4. Параметры CSS, управляющие положением на странице

position: absolute | fixed | relative | static | inherit

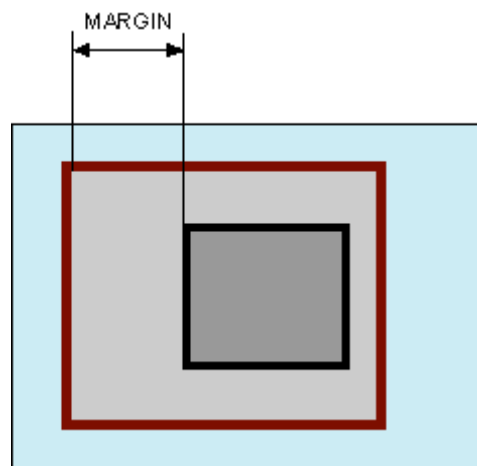
Устанавливает способ позиционирования элемента относительно окна браузера или других объектов на веб-странице. **absolute** указывает, что используются абсолютные координаты. **relative** указывает, что используются относительные координаты. **inherit** наследует значение родителя.

left задает положение относительно левого края контейнера. **top** задает положение относительно верхнего края контейнера. Задаются в процентах или пикселях.

z-index указывает на то, какой элемент должен располагаться выше при перекрытии. Измеряется в единицах.

visibility определяет видимость элемента. Принимает значения `visible` (видимый), `hidden` (скрытый), `inherit` (наследуется от родительского элемента).

margin устанавливает величину отступа от каждого края элемента. Отступом является пространство от границы текущего элемента до внутренней границы его родительского элемента.



Разрешается использовать одно, два, три или четыре значения, разделяя их между собой пробелом. Если задано одно значение, то будут установлены четыре одинаковых отступа. Если два, то первое значение – отступ сверху и снизу, второе – слева и справа. Если определены три значения, то первое значение задает отступ от верхнего края, второе — одновременно от левого и правого края, а третье — от нижнего края. Четыре параметра определяют отступ от верхнего, правого, нижнего и левого края.

padding устанавливает значение полей вокруг содержимого элемента. Полем называется расстояние от внутреннего края рамки элемента до воображаемого прямоугольника, ограничивающего его содержимое. Также как и у `margin` возможны несколько вариантов значений.

