**Блочная верстка web-сайта. Часть 2**

**Блочная верстка сайта**, или верстка средствами **тегов div** (иногда называют дивная верстка). Блочный элемент представляет из себя прямоугольную зону, из таких прямоугольников создается **блочный макет сайта**. **Блочная верстка** на сегодняшний день является ведущей технологией для создания макетов и шаблонов сайтов

 Верстать надо сайт о Летающих Парасенгах.



Рисунок 1. Макет сайта

**Задание 1. Создание шапки сайта**

1. Создайте в блокноте новый документ с расширением css и сохраните его под именем **style.css**в папке **div-verstka**.

2. Создайте HTML-документ и сохраните его в той же папке под именем **index.html.** Почему именно index? Любой сервер, при заходе пользователя по адресу станет сразу же искать страницу с этим названием. Индексная страница для них всегда является главной, то есть – стартовой.

3. В самом начале HTML-документа впишите следующую строку (можно скопировать):

**<!DOCTYPE html>**

**DOCTYPE**

Определяет тип нашего документа. Любая грамотно сверстанная страница должна в самом начале содержать так называемый DOCTYPE. Нужен он для всевозможных устройств вывода информации и браузеров в том числе. Пока что все ныне существующие браузеры прекрасно обходятся и без указания DOCTYPE. Но уже грядут те времена, когда страница сайта, сразу начинающаяся с тэга <html>, просто не будет прочитана, потому как стандарты становятся жестче.

4. Следующий тэг **<head>**. В нем содержится разная служебная информация: название страницы (то, что входит в <title>), кодировка, ключевые слова для поисковых роботов, описание страницы и т.п. Вся эта информация записана в служебных тэгах **meta**, которые также не отображаются на странице в браузере.

5. Впишите на страничке следующую информацию (можно скопировать):

**<head>  
<meta charset="UTF-8">  
<meta name="description" content="Сайт о поросенках-летунах и счастливой летучей жизни." />  
<meta name="keywords" content="летать, свин-летун, пиггсы, полеты, леччики, пилоты." />  
<link rel="stylesheet" href="style.css" type="text/css" />**

Поясним:

Первый мета-тэг показывает на кодировку сайта. Желательно пользовать **UTF-8 с BOM**, чтобы случайно у пользователей не вышли вместо красивого и грамотного текста всякие некорректные символы.

Второй мета-тэг – это краткое описание сайта. Именно эта строка первой покажется в результатах поиска Яндекса или Рамблера, если они, конечно, найдут наш сайт.

В третьем мета-тэге ключевые слова для поисковиков.

И наконец последняя строка – ссылка на лист стилей **CSS**. По умолчанию лист стилей могут использовать любые устройства вывода информации будь то экран монитора, принтер.. Но если мы хотим указать для какого устройства конкретно предназначена страница, то после атрибута **type**со значением **text/сss**должны будем указать дополнительный атрибут media со значениями:

print – для принтера,

handheld – для «наладонника»,

screen – для монитора only и т.д.

Это понятно – если кто-то захочет распечатать нашу страничку, то ему вовсе ни к чему на отпечатке навигация, кнопки-картинки, рекламные баннеры и супер-красивые заголовки. Только полезная информация. И вот для такого случая пишется отдельный лист стилей, попроще.

6. Следующим парным тегом будет:

**<title>Пиггасы | Главная</title>**

7. Закройте </head>.

8. Допишите оставшиеся теги.

**<body>**

**</body>**

**</html>**

9. Сохраните файл.

Структура блоков в нашем сайте будет иметь следующий вид (рис. 2).

Поясним схему:

1. Весь сайт расположен в блоке **<div id=”container”>…</div>** (красный контур).
2. Первым в блоке container идет блок шапки сайта **<div id=”header”>…</div>** (черный контур).
3. Затем идет блок панели навигации **<div id=”nav”>…</div>** (голубой контур).
4. Ниже панели навигации расположены в две колонки блоки: с контентом **<div id=”text”>…</div>** (пурпурный контур).  
   и рядом блок новостей **<div id=”news”>…</div>** (зеленый контур).
5. Внутри блока text расположен блок со списком новых участников **<div id=”members”>…</div>** (коричневый контур).
6. После двух колонок text и news расположен блок «подвал» **<div id=”footer”>…</div>** (темно-синий контур).



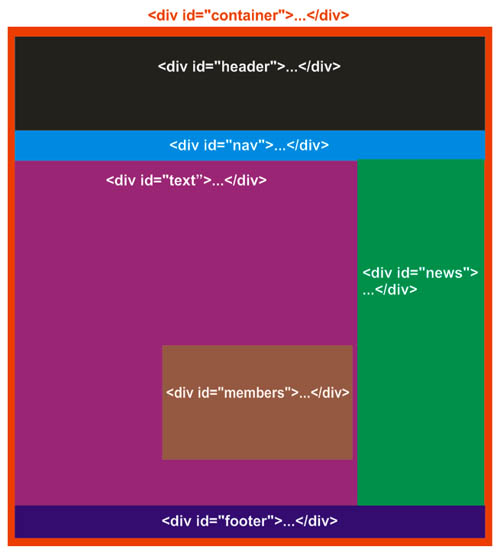


Рисунок 2. Блочная схема сайта

Сначала зададим общие правила для страницы с блочной версткой:

10. В css документе наберите следующий код (больше не копируем, набираем ручками)

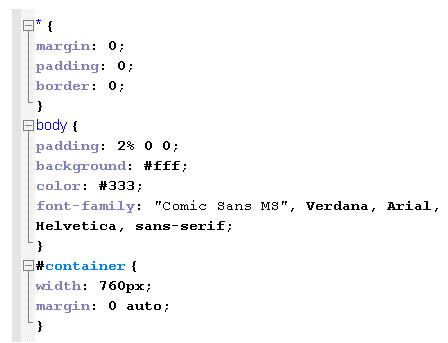


Рисунок 3

Поясним:

1. В первом правиле звездочка означает не что иное, как всю страницу. Сама звездочка – это не тэг и нигде потом в коде страницы не указывается. Браузеры прекрасно понимают ее значение и применяют данные с ней правила ко всей странице. В правиле мы указали последовательно:

Отступы – 0,

Поля – 0,

Рамка – 0.

Это нужно для того, чтобы ни поля, ни отступы, ни рамки не появлялись там, где нам не нужно. Если этого специально не указывать, то например тот же IE (Internet Explorer) выставит эти параметры по своему умолчанию. **Значения указываются либо в процентах, либо в пикселах**. **Если стоит ноль, то единицу измерения не нужно указывать.**

2. Следующим правилом мы задали для тела страницы следующие указания: поля – сверху 2%, с боков по нулям, снизу тоже ноль. Это значит, что наша страничка не будет лепиться к верхушке окна браузера, а отступит от него на 2% размера окна. Тут значения **идут подряд без запятых, и только после последнего ставится точка с запятой**.

Вообще у любого прямоугольника есть 4 стороны. И значения для них задаются по часовой стрелке, начиная сверху, затем правое, низ и левое.

Так как у нас по бокам должно быть одинаковое расстояние от края экрана (в нашем случае ноль, то есть на всю ширину экрана), то и значений мы дали всего три – **2% 0 0**. Кого смущают проценты, могут написать так: 15px 0 0. Это будет почти то же самое. Средняя цифра в этой записи – ноль – означает, что она одинакова как для правой, так и для левой стороны.

Еще раз:

а) 5px 10px 15px 20px; – сверху будет поле в 5 пикселов, справа 10, снизу 15, слева 20.

б) 5px 10px 15px; – сверху 5, с боков по 10, снизу 15.

в) 5px 10px; – сверху и снизу по 5, с боков по 10.

г) 5px; – со всех сторон по 5.

Пункты б), в) и г) здесь – это просто сокращенная запись.

Остальное несложно: фон белого цвета, цвет шрифта темно-серый (#333 или, если угодно #333333), ниже перечислены в порядке предпочтения используемые семейства шрифтов. Тут есть одно замечание: если имя шрифта состоит более чем из одного слова, то его нужно взять целиком в кавычки. Например: "**Times New Roman"**

3. В следующем правиле – слово **container**с решеткой (#). Данная решетка означает уникальность атрибута. То есть тэг **div**с данным атрибутом будет использован **только один раз** на странице.

Зачем вообще нужен контейнер при блочной верстке? А затем, чтобы поместить нашу страничку в центр экрана монитора. Для этого мы указали у контейнера отступы: сверху и снизу ноль, а с боков **auto**. Это означает, что при любом размере экрана наш сайт всегда будет строго по центру. Ширина страницы при этом равна 760 пикселям.

Возникает вопрос: а почему бы у тэга body не указать такую же ширину страницы и автовыравнивание? Дело в том, что браузеры читают данный тег (body) по-своему и отдают под него ВСЮ видимую область экрана. То есть, если бы мы даже и захотели задать автовыравнивание для body, но при этом ширину страницы задали в 760 пикселей, то страничка все равно «лепилась» бы к левому краю экрана. А нам это не нужно. Вот мы и приспособили «коробочку» под нашу страничку.

11. Из папки **html\_css\_7**скопируйте в свою папку все изображения. Мы будем их использовать для размещения на странице.

12. Далее в листе стилей запишите правило для “шапки” (header):

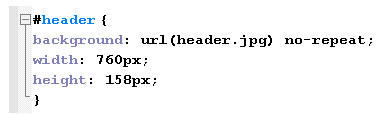


Рисунок 4

Здесь мы указали, что вся наша шапка залита фоном-картинкой с размерами 760х158 пикселей.

**url(header.jpg)**– это ссылка на картинку.

Запись **no-repeat**означает, что картинка не должна повторяться.

13. Между тегами **<body></body>** добавляем следующий код:

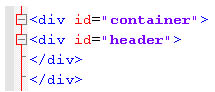


Рисунок 5

14. В результате должно получиться следующее (рис. 6).

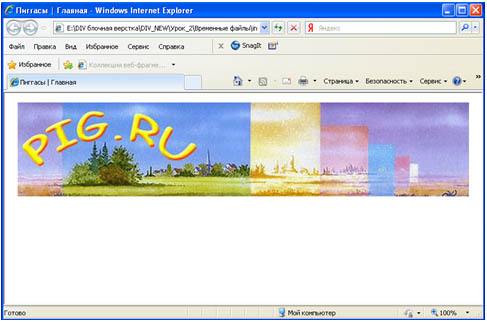


Рисунок 6

**Задание 2. Создание блока навигации**

1. Продолжим писать код для листа стилей. Следом за правилом для шапки запишем правило для блока навигации:

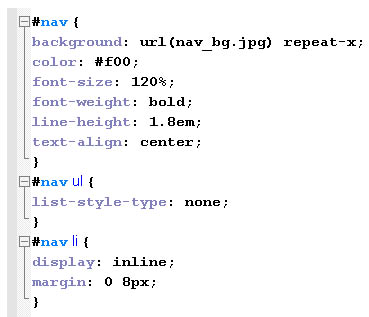


Рисунок 7

Разберем код.

Панель навигации – сразу под шапкой, горизонтальная. Для ее реализации мы воспользуемся таким элементом как маркированный список.

Блок навигации назван атрибутом **nav**. Вначале указаны общие настройки для него: background – это картинка с именем **nav\_bg.jpg** размером 8х35 пикселей. Это обычный “столбик” с градиентом от белого к серому сверху вниз. Чтобы растянуть его по всей полосе навигации, мы указали в значении слово **repeat-x,** что означает “повторить по оси х”, то есть по горизонтали.

Далее мы указали цвет шрифта ярко-красного цвета #f00 для активного раздела. Он у нас не будет ссылкой, а останется простым текстом (делать на Главной странице ссылку на самое себя – тавтология). Остальное в этом правиле несложно: размер шрифта, толщина, высота по вертикали (единица измерения em, которая равна высоте прописной буквы выбранного шрифта. Значение 1.8em показывает, что шрифт увеличен на 1.8 высоты стандартной буквы), и выравнивание текста по центру.

Следующее правило указывает, что у нашего списка не должно быть никаких маркеров. Это задается значением none. Зачем нам в меню лишние точки, кружочки или квадратики?

По-умолчанию строки списков всегда располагаются в столбик. Чтобы этого не происходило, мы далее указываем для строк списка расположение по горизонтали, то есть в линию – **display: inline;**

И еще добавляем отступы: сверху и снизу по нулям, с боков по 8 пикселей.

А теперь укажем, каким образом наши менюшки будут реагировать на наведение мышки. Менюшки – это ссылки на другие разделы нашего сайта. Поэтому пропишем еще пару правил для ссылок.



Рисунок 8

В первом мы обозначили цвет ссылки в спокойном состоянии, а во втором – в активном, то есть при наведении мыши.

2. В конце таблицы стилей добавьте следующее правило:



Рисунок 9

Это общее для всех ссылок правило. Оно указывает, что все ссылки на странице с блочной версткой по умолчанию не используют подчеркивание.

3. Найдите в коде следующее место:

<div id="header">

</div>

и сразу же за ним запишите следующее:

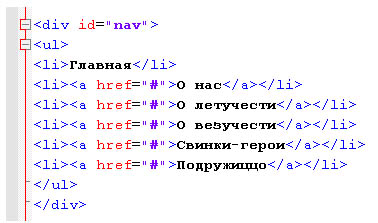


Рисунок 10

Наши разделы оформлены как пункты списка, и каждый пункт, кроме первого, является ссылкой. В данном случае вместо адреса несуществующих страниц мы просто вставили решетку (#), которая всегда возвращает нас на текущую страницу.

|  |
| --- |
| Теперь настала пора сказать пару слов о *каскадности стиля*. В правилах для блока навигации мы вначале указали настройки для всего блока **#nav**, затем для маркированного списка, обозначенного тэгом **ul**, далее показали правила для строк **li**… Каждое последующее правило получает “в наследство” характеристики предыдущего: от **nav**к **ul**, от **ul**к **li**. Все вместе они являются вложенными в тэг контейнера и получают также от него часть правил (в частности, центрирование посередине экрана и заданную ширину в 760 пикселей). Это и является своеобразным каскадом. |

 4. Сохраните файлы и просмотрите результат в броузере. У Вас должно получиться следующее (рис. 11).

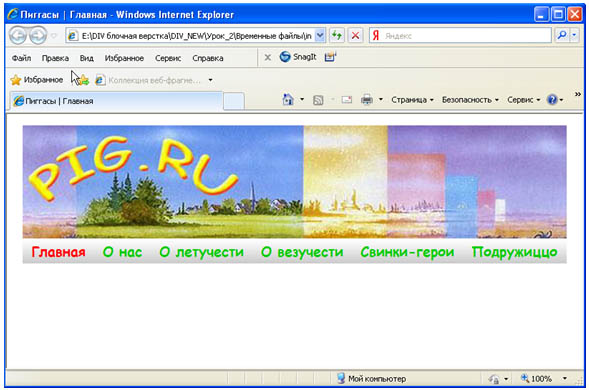


Рисунок 11

**Задание 3. Создание блока с контентом**

Наполним нашу страницу контентом. На схеме сайта (см. рис. 2) видно, что полезная площадь страницы разделена на две функциональные области:

* Основной текст (с картинками и пр.)
* Блок новостей.

Обычно, такую блочную верстку называют двухколоночной. Нам надо сделать так, чтобы текст остался слева, а новости справа.

1. Запишите на страничке сразу после этого места:

………

<li><a href="#">Свинки-герои</a></li>

<li><a href="#">Подружиццо</a></li>

</ul>

</div>

следующий код:

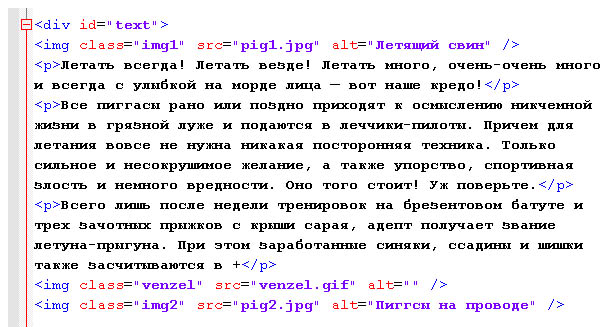


Рисунок 12

2. Сохраните файл и просмотрите в броузере. Сайт выглядит «поломанным» (рис. 13).



Рисунок 13

Тэг **div**с атрибутом **text**– это и есть область полезного содержимого странички, иначе называемое **контентом**. Как и в любом тексте тут мы видим абзацы, обрамленные тэгами **р**, а также несколько изображений (тэги **img**).

С тэгами абзаца все понятно: открылся один, за ним часть текста, закрылся; открылся следующий, за ним опять часть текста, закрылся, и т.д.

У тэгов изображений есть слово **class**. Оно означает многоразовость использования данного атрибута (в противовес индивидуальному **id**). Кроме того, **class**можно использовать и вовсе без тэга **div**! Что мы и сделали. У тэгов изображения мы просто вписали это слово сразу же за самим тэгом.

3. Запишем далее в листе стилей CSS:

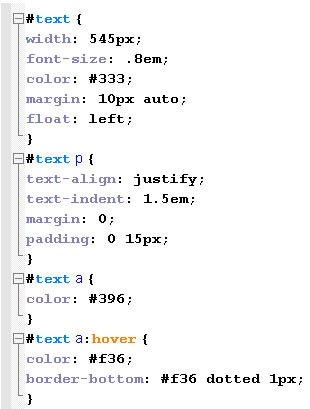


Рисунок 14

В первом правиле мы указали, что ширина у области текста будет равна 545 пикселям. Размер шрифта 0.8em (в данном правиле ноль можно не писать, .8em – обозначает тоже самое). С отступами понятно – верх и низ по 10 пикселей, по бокам на автомате. А вот последняя строчка как раз и задает местоположение нашего блока текста с левой стороны. Слово **float** переводится как “обтекание“. Но тут есть одна особенность. Читаем: “обтекание – слева”. Но ведь это сам текст находится слева! А обтекает его все остальное справа. В этом есть некая путаница. Чтобы не заплутать, просто запомните: **left** – сам объект слева, а течет все правее. И наоборот, **right** – объект справа, а течет все левее.

4. Посмотрите, что получилось – текст выровнялся по левому краю странички (рис. 15), оставив справа пустое место. В это пустое место мы потом и вставим блок новостей, присвоив ему в листе стилей значение right для атрибута float.



Рисунок 15

В следующем правиле мы для абзацев нашего текста задали выравнивание по всей выделенной площади. Слово justify как раз это и означает. Если этого не указать, то по умолчанию весь текст выровняется по левому краю.

Далее. Слово indent означает не что иное как обычную “красную строку“. Размер ее также указан в полтора размера шрифта.

Ну и, наконец, ссылочки. Для неактивной задали цвет салатовый, а для активной красный, да еще и с подчеркиванием точечной (dotted) полоской в 1 пиксель толщиной.

5. А теперь давайте укажем правила для наших картинок. Запишите в листе стилей:

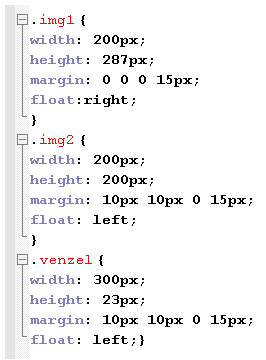


Рисунок 16

Каждую картинку мы назвали своим атрибутом **img1**, **img2**, **venzel**, указав в каждом правиле размеры картинок и отступы для них. Вообще отступы и поля проще всего подбирать опытным путем. То есть вначале просто без них кидаем блоки или картинки на страницу с блочной версткой, а потом смотрим, куда все это оно сползло и чуток корректируем, добавляя где надо и убавляя, где не надо. Кстати, значения можно указывать и с минусом. Например, минус 10px. И картинка сдвинется в противоположную сторону, хоть даже и за край экрана.

Каждой картинке мы задали обтекание в соответствии с ее расположением на странице. Первая картинка будет справа от текста, вторая — слева, и вензель тоже слева.

И еще одна вещь. Как видите, данные правила начинаются не с решетки **#**, а с точки. Это и есть признак того, что правило относится не к **id**, а к **class**.

6. Теперь сохраняемся и любуемся на то, что у нас получилось. Если все выполнили правильно, сайт будет выглядеть следующим образом (рис. 17).

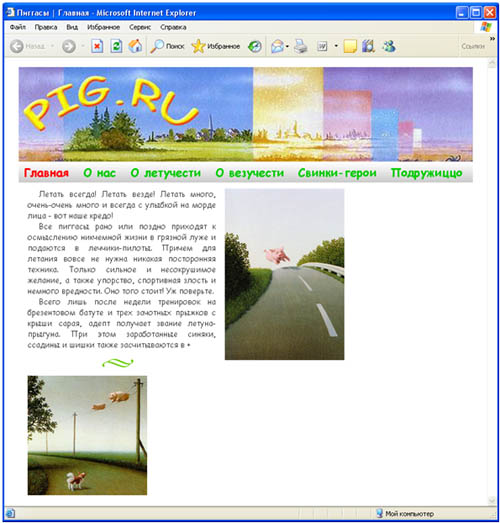


Рисунок 17

В блоке текста мы разместим список новых участников. Вообще за это отвечает какой-нибудь php-скрипт, но так как мы делаем простую статичную страницу, то оформим эту штуку обычным списком. Только на этот раз не *маркированным*, а *нумерованным*. Задается такой список тэгом **ol**.

7. На страничке сразу после вот этого места:

……………..

<img class="venzel" src="venzel.gif" alt="" />

<img class="img2" src="pig2.jpg" alt="Пиггсы на проводе" />

вставите следующий код:



Рисунок 18

Появился новый атрибут **members**. Так мы назвали наш список. Тэг **h2**– это заголовок второго уровня. Первый уровень, как легко догадаться, это чаще всего название сайта или страницы.

Далее у нас идут подряд два нумерованных списка. Зачем два? На макете мы разместили новых участников в две колонки с вертикальной нумерацией в каждой. Можно было бы, конечно, занести всех в один общий список, сделать ширину одной ячейки ровно вполовину отведенного под список места, и тогда ячейки автоматом переносились бы на новую строчку. Но тогда нумерация выглядела бы так:

**1 2**

**3 4**

**5 6**

и так далее. Согласитесь – коряво как-то.

Поэтому мы сделали просто два списка подряд. Но! Каждому в листе стилей задали обтекание слева. Об этом чуть позднее. Сразу после списков мы положили картинку волнистой линии.

Ну, а теперь откроем наш лист стилей и запишем еще кусочек красивых правил.

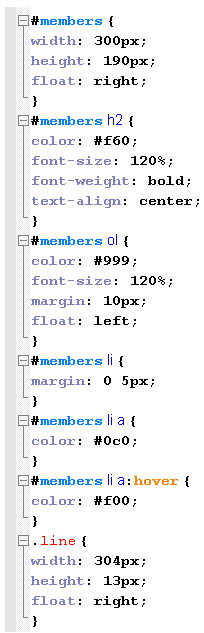


Рисунок 19

Расшифруем. Для начала мы задали всему блоку размер. Причем на этот раз не только ширину, но и высоту. Для чего это нужно? Если, например, в нашем списке было бы чуток меньше народу, то картинка волнистой линии “поджимала” бы список снизу, а нам надо, чтобы она была почти под срез картинки с летящим поросенком слева. Поэтому мы задали жесткий размер как по ширине, так и по высоте. И обтекание справа. Ну, это уже понятно – список должен быть правее указанного рисунка.

Далее задали правило для заголовка второго уровня. Здесь нам уже все знакомо.

Для самого нумерованного списка с тэгом **ol**мы указали обтекание слева. То есть оба наших списка (помним, что в коде страницы их два подряд) будут идти не в столбик, а один подле другого рядышком.

Все имена в списке оформлены в виде ссылок (на странички профиля участников). Для них мы задали только цвета, без всяких подчеркиваний. Но почему-то на страничке они все равно подчеркиваются при наведении мышки, да еще точечной линией! Вот тут как раз и сработал так называемый **каскад**– список-то лежит в зоне действия атрибута **text**и поэтому просто перенял от него часть правил.

Последнее правило здесь для рисунка линии. Оно так же, как и предыдущие картинки, оформлено классом.

8. Сохранились. А теперь смотрим, что получилось (рис. 20).



Рисунок 20

**Задание 4. Создание блока новостей**

1. Открываем нашу страничку**index.html**. Находим вот это место в блочной верстке:

**……….**

<li><a href="#">Васек</a></li>

</ol>

</div>

<img class="line" src="line.gif" alt="" />

и следом сразу вставляем вот этот кусочек:

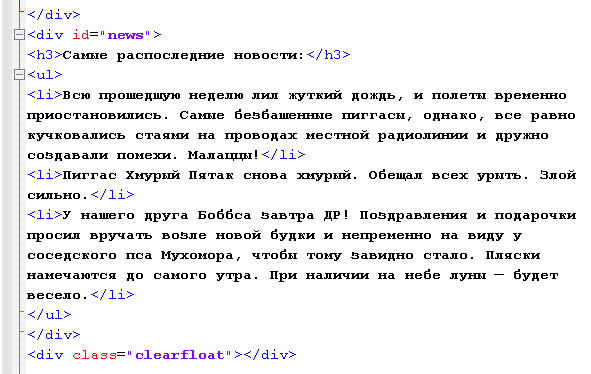


Рисунок 21

Что мы тут видим? Во-первых, закрывающий тэг **</div>**, который показывает, что область основного текста закончилась. Далее идет маркированный список новостей с заголовком 3-го уровня (**h3**).

В самом низу мы добавили “очистку обтекания” – clearfloat (для нее также в листе стилей пропишем свои правила, а именно очистку с обоих сторон). Для чего это? Вообще наша верстка является “плавающей“. Вон сколько у нас уже обтекаемых элементов. При этом разные браузеры по-разному воспроизводят такую верстку (о том, что IE по-своему “видит” поля и отступы уж и не говорю). В Opera и Firefox можно было бы обойтись и без очистки и сразу ниже писать код футера. Но в IE в таком случае появляются странные подергивания фона у футера при наведении на ссылки мышки. Браузер словно хочет подпрыгнуть и заполнить все пустующие места какие есть. Поэтому мы просто добавим эту строчку.

2. А теперь открываем наш лист стилей и допишем следующий код:

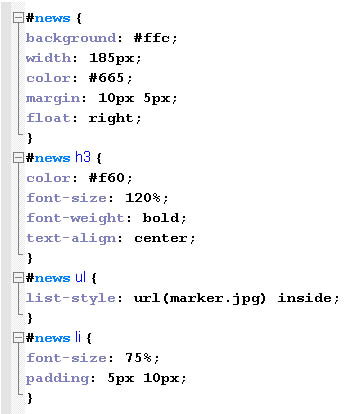


Рисунок 22

Здесь мы для начала поменяли фон для блока новостей, чтобы визуально отделить колонку от основного содержимого. Затем задали ширину блока, цвет для шрифта и обтекание справа. С заголовком получилось хитро. Если вернуться чуток назад, то можно заметить, что у заголовка 2-го уровня (у списка участников), также задан размер шрифта в 120%.

Вообще по-умолчанию браузеры сами определяют размеры для заголовков в зависимости от их ранга. То есть по логике вещей, заголовок 3-го уровня должен быть мельче 2-го. А у нас наоборот! Это произошло потому, что заголовок списка участников находится в блоке основного текста, для которого мы задали размер всего шрифта в 0.8em.

Следующим правилом мы задали для списка стиль маркеров. По-умолчанию любой браузер отображает маркеры в виде черных кружков. Но можно задать и другое отображение – окружностей, квадратов, без маркеров, или вообще свой собственноручно нарисованный маркер. Как раз так мы и поступили. Нарисовали симпатичную зеленую галку с размерами 14х17 пикселей и указали в стиле ссылку на нее – **url(marker.jpg)**. Следом приписали слово **inside**. Что это значит? Маркеры по-умолчанию не входят в сам блок текста. Когда мы задаем поля для текста, то маркеры не слушаются и вылезают за края. Это не всегда хорошо. Поэтому мы принудительно указали им быть inside, то есть внутри колонки с текстом.

3. Сохраняемся и смотрим, что вышло! (рис. 23)



Рисунок  23

**Задание 5. Создание «подвала» web-страницы**

Нам осталось только оформить “подвал” (footer), да сделать еще пару-тройку замечаний.

Подвал, он же “футер” – это весьма важная часть сайта с блочной версткой, хотя и мало кто туда добирается, особенно при очень длинных страницах. Там обычно дублируются ссылки на разделы сайта, пишутся копирайты и контактная информация. Очень часто можно наблюдать такую картину: верхнее горизонтальное меню под “шапкой” практически без изменений дублируется в левой колонке. Выглядит это, по меньшей мере, бестолково. Уж лучше продублировать основные разделы сайта в подвале. Пользы будет несомненно больше.

Футер не должен доминировать над шапкой, но и не должен теряться, делая страницу неуравновешенной. Мы сделаем его немного контрастным по цвету, но небольшой высоты.

1. Запишите в коде страницы сразу после этого места:

……………..

…намечаются до самого утра. При наличии на небе луны — будет весело.

</ul>

</div>

<div class="clearfloat"></div>

Следующий код:

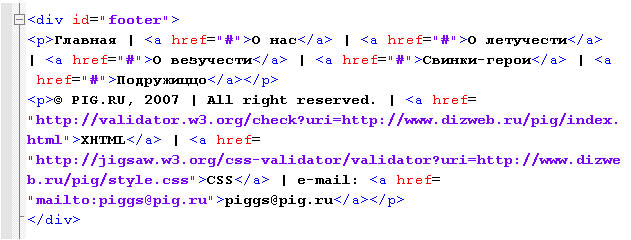


Рисунок 24

***Примечание:*** знак копирайта в html-коде осуществляется записью **&#169**

Здесь у нас слово «Главная» не является ссылкой (об этом мы уже говорили), а следом идут обычные ссылки на другие разделы сайта. Всего в футере два абзаца. В первом ссылки на разделы, а во втором как раз и есть всякие копирайты, ссылка на “мыло” и еще две интересные ссылочки. Это ссылки на валидаторы — конторы, которые проверяют вашу страницу на соответствие стандартам. Ссылки устроены так, что нажимая на них, вы сразу даете команду этим самым валидаторам проверить ваши странички (которые уже лежат в сети конечно же). И тут же получаете ответ. Сие придает вашим страничкам солидность и всеобщее наше верстальское одобрение.

Вот практически и вся страница. Проверьте только, чтобы в конце кода у вас было два </div> подряд. Вот так:

……….

href="mailto:piggs@pig.ru">piggs@pig.ru</a></p>

</div>

</div>

</body>

</html>

Второй **/div**— это закрытие области контейнера.

2. Теперь допишем в листе стилей оставшийся код:

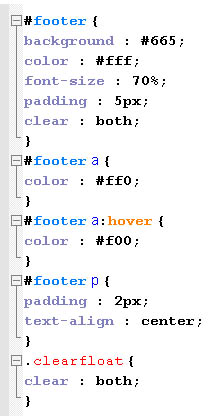


Рисунок 25

Здесь мы задали фон серо-зеленого цвета, цвет текста – белый. А вот нижняя строчка в первом правиле (**clear: both;**) означает, что с обеих сторон подвала не должно быть ничего. То есть он у нас занимает всю ширину нижней части контейнера.

Ссылки у нас здесь ярко-желтые, а в активном состоянии – красные. Для текста мы выбрали размер мельче всех на странице – 70%. Как я и сказал, футер– важная часть, но не настолько, чтобы бросаться в глаза.

Выравнивание для текста мы задали по центру.

Самое последнее правило как раз служит для блока очистки, заданного атрибутом **clearfloat**, о котором мы говорили, верстая блок новостей.

3. Сохраняемся, смотрим нашу блочную верстку (рис. 26).



Рисунок 26

4. Все вроде бы хорошо, но как-то неуютно (особенно на белом фоне экрана). Добавим-ка мы всей странице тонкую светло-серую рамку! Откроем лист стилей, найдем вот это место:

#container {

width : 760px;

margin : 0 auto;

}

И добавим следующую строчку:

http://web-verstka.ru/webverstka/kurs_html_css/images_html_css/8.27.jpg

Рисунок 27

Должно получиться следующее:

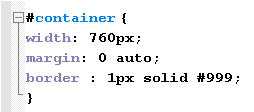


Рисунок 28

5. Вот, собственно, и все! (рис. 29)



Рис. 29