

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

УТВЕРЖДАЮ

Проректор-директор ИК

\_\_\_\_\_ А.А. Захарова

« \_\_ » \_\_\_\_\_ 2015 г.

## **ВЕРСТКА ШАБЛОНА ВЕБ-САЙТА**

Методические указания к выполнению лабораторных работ  
по курсу **«Интернет-технологии»**  
для студентов направлений 09.03.02 «Информационные системы и тех-  
нологии»

*Составитель: И.И. Савенко*

Издательство  
Томского политехнического университета  
2015

УДК 629.76  
ББК 00000  
А00

**Савенко И.И.**

А00 Верстка шаблона веб-сайта: методические указания к выполнению лабораторных работ по курсу «Интернет-технологии» для студентов направлений 09.03.02 «Информационные системы и технологии» Института кибернетики

ТПУ / Сост.: И.И. Савенко; Томский политехнический университет. – Томск: Изд-во Томского политехнического университета, 2015. – 33 с.

**УДК 000000**  
**ББК 00000**

Методические указания рассмотрены и рекомендованы  
к изданию методическим семинаром кафедры  
АиКС ИК  
« 01 » сентября 2015 г.

Зав. кафедрой АиКС  
кандидат технических наук,

\_\_\_\_\_ *А.С. Фадеев*

Председатель учебно-методической  
комиссии

\_\_\_\_\_

*Рецензент*

Кандидат технических наук, **доцент** кафедры АиКС ИК ТПУ  
*В.Н. Вичугов*

© Составление. ФГБОУ ВПО НИ ТПУ, 2015  
© Савенко И.И., составление, 2015

## Содержание

|                                                  |    |
|--------------------------------------------------|----|
| ВВЕДЕНИЕ.....                                    | 4  |
| 1 Подготовительный этап.....                     | 5  |
| 2 Проектирование шаблона будущего веб-сайта..... | 8  |
| 3 Верстка шаблона веб-сайта.....                 | 9  |
| 1.1. Создание каркаса веб-сайта.....             | 11 |
| 1.2. Блок быстрой навигации.....                 | 17 |
| 1.3. Блок с логотипом и главным меню сайта.....  | 20 |
| 1.4. Рекламный блок.....                         | 23 |
| 1.5. Новостной блок.....                         | 24 |
| 1.6. Блок с предложениями.....                   | 26 |
| 1.7. Блок подписки на рассылку.....              | 28 |
| 4 Задание на лабораторную работу.....            | 31 |

## ВВЕДЕНИЕ

Целью данной лабораторной работы является знакомство с таким понятием как верстка. Получение студентом навыков формирования гипертекстовых документов с использованием языка разметки HTML и каскадных таблиц стилей CSS.

Разработка веб-сайта – это сложный процесс, состоящий из нескольких последовательных этапов:

- подготовительный;
- проектирование шаблона будущего веб-сайта;
- верстка;
- программирование;
- наполнение контентом;
- продвижение сайта;
- поддержка сайта.

В рамках данной лабораторной нас будет интересовать этап верстки, на нем мы остановимся более подробно, но и остальные этапы без внимания не оставим.

## 1 Подготовительный этап

Подготовительный этап является одним из самых важных этапов в процессе разработки веб-сайта, на данном этапе определяется его назначение и цели. От целей будет зависеть весь дальнейший процесс разработки. Так же на этапе подготовительного этапа осуществляется поиск и анализ существующих аналогов, которые преследуют ту же цель что определили и вы. После чего разрабатывается техническое задание (ТЗ) на веб-сайт, если присутствует заказчик, то ТЗ составляется совместно с ним. Техническое задание – это документ, который регламентирует, какие работы должны быть выполнены. Все что описано в ТЗ должно содержать объективные критерии, по которым можно определить, сделан ли тот или иной пункт работ или нет. Для общего развития приведу некоторые пункты ТЗ:

**Введение.** Раздел вводит в курс дела разработчиков веб-сайта.

**Эксплуатационная выгода.** В данном разделе описывается выгода, которую должен принести веб-сайт, но не всегда выгода сводится только к деньгам. Например, сайт визитка позволит прорекламирровать свою компанию в сети интернет, соответственно, чем больше информации о компании, тем больше клиентов она к себе привлекает.

**Функциональное назначение.** Необходимо указать перечень того, какими техническими средствами будет получена выгода, которая была описана в пункте «Эксплуатационная выгода». Если говорить о Интернет магазине, то в данном пункте можно указать следующее:

*«Веб-сайт должен обеспечить возможность создания **каталога товаров**, предоставить возможность добавлять выбранные товары в **корзину**. На веб-сайте должны присутствовать страницы с **информацией о доставке и возврате товара** ..... Модуль публикации **новостей** на сайте ...»*

**Термины и определения.** Наличие данного раздела в ТЗ дает уверенность, что заказчик и исполнитель говорит об одном и том же. Термины указываются с двух сторон, как от заказчика к исполнителю, так и от исполнителя к заказчику. Например: Хостинг, SMTP-сервер и прочее. Не нужно указывать специфичные для предметной области термины и не имеющие отношения к реализации проекта.

**Данные и списки.** Ключевой раздел ТЗ. В данном разделе определяются детали будущего веб-сайта, соответственно накладываются ограничения, что позволит объяснить заказчику, что в ТЗ написано именно так, дополнительные задачи решаются за отдельные деньги. Обычно данный раздел занимает больше половины всего ТЗ. Раздел «Данные» содержит перечень сущностей, которые используются в проекте. Это очень близко к описанию таблиц в базе данных. В пункте «Функциональное

назначение» мы определили, что веб-сайт должен иметь возможность публикации новостей, тогда в этом пункте необходимо дать определение, что такое новость в понимании заказчика. Например: *новость должна состоять из следующих полей: «Заголовок», «Текст новости», «Дата публикации», «Анонс».* На примере «Новостей», в разделе «Списки» необходимо описать по какому принципу «Новости» группируются в списки, например: *На веб-сайте должна быть возможность отображения «Популярных новостей», в списке должно отображаться 10 самых популярных новостей, новость считается популярной, если она имеет максимальное количество просмотров за день.* Соответственно, для реализации последнего требования необходимо в раздел данные добавить поле «Количество просмотров».

Как вы заметили, два данных пункта очень сильно зависят друг от друга и позволяют детально судить о функционале сайте и о ходе процесса разработки.

**Описание страниц.** Раздел позволяет понять, какую информацию отображать и как её располагать на страницах. Данный раздел достаточно короткий, так как мы всегда можем сослаться на раздел «Данные и списки».

Вот пример: *На странице редактирования новости при нажатии на кнопку «Сохранить» происходит сохранение изменений новости без перезагрузки страницы с использованием технологии AJAX. Перед сохранением осуществляется проверка полей на корректность как показано на рисунке 6.1.»*

The screenshot shows a web form titled "Редактирование новости" (Editing news). At the top, there is a grey error box with a red 'X' icon and the text "Новость не была сохранена" (News was not saved). Below this, the form has several sections: "Заголовок" (Title) with the value "Астероидный Южный Треугольник"; "Текст" (Text) with a paragraph of text; "Анонс" (Preview) which is an empty text area highlighted in red, with a callout box pointing to it that says "Поле не может быть пустым" (Field cannot be empty); "Изображение" (Image) with a placeholder box labeled "100 x 100", a file path "C:\WINDOWS\Greenstone.bmp", and a "Обзор" (Browse) button; a checked checkbox "Удалить изображение" (Delete image); and a "Сохранить" (Save) button at the bottom.

Рисунок 6.1 – Страница редактирования новостей

Для заметки, для быстрого проектирования эскиза будущей страницы можно использовать бесплатное программное обеспечение Pencil Project.

Как вы уже заметили, замечательно, если в описании страницы присутствует эскиз страницы. Всегда необходимо следить о том, что описание соответствует эскизу, а изображение на эскизе не противоречит описанию в разделе «Данные и

списки». Как в данном примере, информации о «Изображении» в разделе «Данные» нет, в дальнейшем это может вызвать проблемы.

**Требование к надежности.** В данном разделе необходимо уточнить, будет ли разрабатываемый веб-сайт высоконагруженным, так как это вполне может потребовать специфических действий по настройке серверов или написанию кода, и как вы понимаете, лучше это определить до разработки, чем после. Так же надежность подразумевает под собой и наличие механизма создания резервной копии БД, если данный механизм необходим, то также необходимо определить периодичность создания резервной копии и куда будет осуществляться её копирование (удаленный или не удаленный сервер).

**Требование к хостингу.** В данном разделе необходимо определить хостинг провайдера, это связано с тем, что в зависимости от провайдера определяются и технологии на котором будет разрабатываться веб-сайт: интерпретаторы, библиотеки, пакеты, объему дискового пространства, памяти и т.д.. Приведу пример: мы определили, что наш веб-сайт должен осуществлять резервное копирование БД на удаленный сервер, соответственно создание и отправка резервных копий на удаленный сервер будет осуществляться с использованием утилиты `rsync` (с ней вы знакомились на курсе СПО), которая является стандартной для Unix-подобных систем, но хостинг провайдер не предоставляет доступ к данной утилите. Соответственно требование ТЗ не может быть выполнено и об этом нужно позаботиться заранее.

**Наполнение контентом.** Перед тем, как сдать проект заказчику, нужно протестировать весь функционал, добавив предварительный контент, который покажет заказчику, что сайт полностью функционирует. Как раз в этом разделе обговаривается объем добавляемого контента, примером является создания учетной записи администратора (что тоже является своего рода контентом, без этого заказчик даже не попадет в часть администрирования сайта).

Описание этого раздела предостережет нас от разного понимания того, кто должен залить 500 фотографий и наполнить каталог товарами.

**Сдача и приемка.** В данном разделе описываются технические условия, при наступлении которых должен состояться расчет за работу. Как вариант, перенос на хостинг разработанного веб-сайта только после сто процентной оплаты работы.

Я привел только некоторые пункты ТЗ, и такое ТЗ не охватывает все стороны веб-сайта, но для очень большого числа проектов оно станет хорошим описанием.

## **2 Проектирование шаблона будущего веб-сайта**

После того, как ТЗ закончено и согласовано со всех сторон, ТЗ передается группе дизайнеров, которые осуществляют разработку шаблона веб-сайта в таких программных продуктах как Photoshop или GIMP. Этот этап можно разбить на несколько вложенных этапов:

- генерация идей дизайна. Формируется расширенная версия макета главной страницы;
- разработка предварительного шаблона дизайна главной страницы;
- согласование шаблона с заказчиком и внесение изменений в шаблон;
- разработка внутренних страниц.

В свою очередь, программисты начинают подготавливать каркас, на котором будет разрабатываться веб-сайт - будет ли использоваться CMS или Framework. И после того, как шаблон готов верстальщики приступают к своей работе.

### 3 Верстка шаблона веб-сайта

Верстка – процесс формирования веб-страницы в текстовом редакторе, в нашем случае мы будем рассматривать верстку с использованием языка HTML (Hypertext Markup Language) и CSS (Cascading Style Sheets). Язык разметки гипертекстовых документов (HTML) представляет собой набор тегов, которые сообщают браузеру о структуре веб-документа, а каскадные таблицы стилей используются для отображения HTML-документов (цвет, шрифт и т.д.). В процессе верстки вам понадобятся лекции по HTML и CSS, потому что мы не будем говорить про назначение всех тегов HTML и CSS свойств, только некоторых из них, целью изучения является сам процесс верстки.

Верстальщик в процессе верстки должен предусмотреть следующие требования к веб-документу:

- **Кроссбраузерность** – страница должна одинаково отображаться в различных браузерах;
- **гибкость верстки** – возможность легко добавлять/удалять информацию на сайте;
- **скорость обработки** – веб-документ должен быть быстро обработан браузером;
- **валидность** – веб-документ должен соответствовать стандартам;
- **семантическая корректность** – логичное и правильное использование элементов HTML.

В зависимости от предоставленного верстальщикам шаблона веб-страницы, выбирается один из двух подходов к верстке, каждый имеет свои достоинства и недостатки:

- **табличная верстка** – ранее была основным и единственным методом верстки, но и сейчас применяется для ускорения вёрстки в ложных случаях. Табличная верстка подразумевает, что структура документа будет определена при помощи таблицы. Особенность в том, что пока таблица не загрузится полностью, её содержимое не начнет отображаться, прежде чем показать содержимое таблицы, браузер должен вычислить необходимые размеры ячеек, их ширину и высоту.
- **блочная верстка** – верстка основана на использовании универсального тегу <div>, который может выполнять много функций. Данный тег является кирпичиком блочной верстки, для которого применяются различные стили, что позволяет делать из него все что угодно (в рамках CSS). При блочной верстке, конечно же, используются таблицы, но только в качестве своего прямого назначения. Как вы понимаете, обильное использование тега <div> снижает читабельность исходного текста веб-страницы. В пятой версии HTML были добавлены новые

теги, К примеру, теги <header> и <footer> были добавлены в спецификацию HTML с целью снизить доминирование тега <div> и придать больше смысла разметке.

Что же касается дизайна шаблона, то он тоже может быть трех видов:

- **фиксированный дизайн** – дизайн, в котором ширина шаблона задана в пикселях и не меняется в зависимости от содержимого.
- **резиновый дизайн** – дизайн, в котором ширина шаблона обычно занимает все пространство окна веб-браузера и задается в процентах.
- **адаптивный дизайн** – дизайн веб-страниц, обеспечивающий отличное восприятие на различных устройствах, подключенных к интернету. Является самым сложным типом дизайна, как для дизайнера, так и для верстальщиков, но облегчает работу программистам, потому что нет необходимости создавать отдельный сайт, например для мобильных телефонов (не всегда это возможно). Целью адаптивного веб-дизайна является универсальность веб-сайта для различных устройств. Существуют различные фреймворки позволяющие облегчить процесс создания адаптивного веб-сайта.

Как было сказано выше, этап верстки мы рассмотрим более подробно и на примере фиксированного дизайна с использованием блочной верстки. Исходное изображение веб-сайта представлено в приложении 1. Хотелось обратить Ваше внимание, что каждый верстальщик выбирает для себя удобный для него способ верстки. Не бывает двух одинаково сверстаных страниц, даже если верстался один и тот же шаблон, поэтому ниже приведенный пример верстки, является только примером и является сугубо личным подходом.



ограничивать ширину каждого раздела, если задний фон был одного цвета, то достаточно бы одного общего для всех частей блока верхнего уровня.

Обратите внимание на изображение 1, пунктиром показы отступы, которые необходимо учесть во время верстки.

Перед версткой обратите внимание, что тег `body` по умолчанию имеет внешние отступы, чтобы от них избавиться, необходимо переопределить это свойство в файле со стилями.

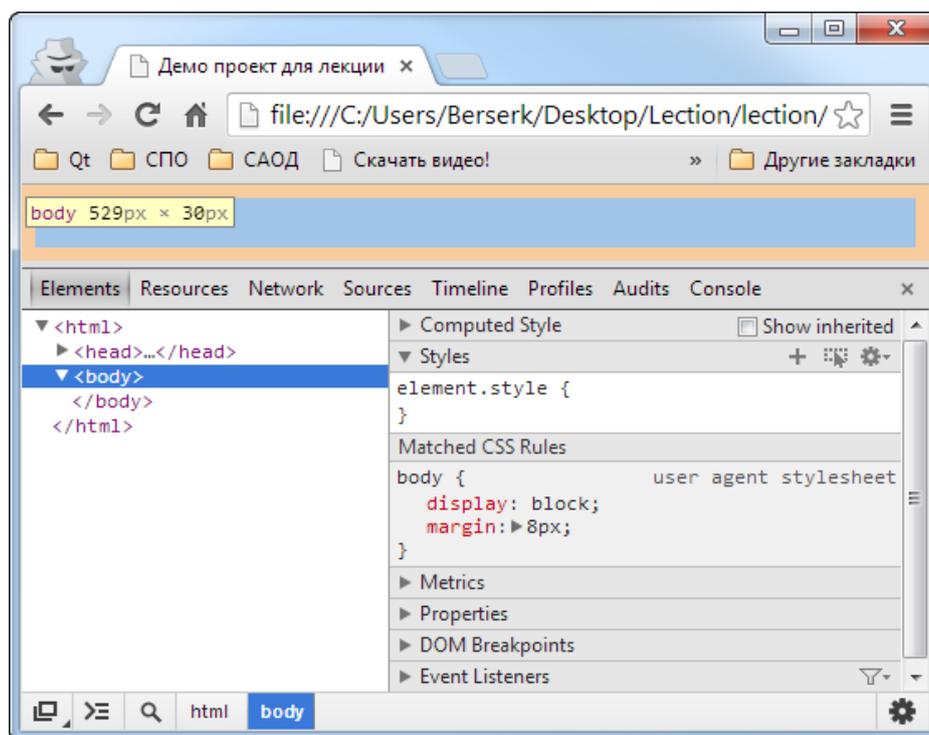


Рисунок 2 – Отступы у тега `<body>`

Ниже приведен код документа и результирующее изображение. Для наглядности будут использованы яркие цвета блоков, чтобы вы могли видеть структуру.

## HTML-страница

```
<html>
<head>
  <title>Демо проект для лекции</title>
  <link href="style.css" rel="stylesheet" type="text/css"/> <!--подключаем стиль-->
</head>
<body>
<!--Родительский блок необходим для того, чтобы задать цвет (так как цвет тянется по всей ширине
браузера)-->
<div class="bound-top-navigation">
  <!--Блок навигации. Класс wrapper (фиксирует ширину раздела сайта) вынесен в отдельный стиль,
он будет повторяться -->
  <div id="top-navigation" class="wrapper">
    </div>
  </div>
<div class="bound-logo-main-menu">
  <div class="wrapper logo-main-menu"> <!--Применяем сразу 2 стиля к одному элементу-->
    <div id="logo"></div>
```

```

        <div id="main-menu"></div>
    </div>
</div>
<div class="bound-advertisement-content">
    <div id="advertisement" class="wrapper"></div>
    <div id="content" class="wrapper">
        <!--Блоки находятся рядом друг с другом, обратите внимания на стили данных блоков-->
        <div id="block-news"></div>
        <div id="block-offers"></div>
    </div>
</div>
<div class="bound-footer">
    <div id="footer" class="wrapper">
        <div id="block-subscription"></div>
        <div id="block-why-us"></div>
        <div id="block-follow-us"></div>
        <div id="block-address"></div>
    </div>
</div>
<div class="bound-below-footer"></div>
</body>
</html>

```

## CSS

```

body {
    margin: 0;
}

.wrapper {
    width: 920px; /*задаем фиксированную ширину контенту*/
    margin: 0 auto; /*выравниваем блок по центру страницы*/
}

/*TOP NAVIGATION*/
.bound-top-navigation {
    background: #f2f2f2;
}

#top-navigation {
    height: 28px;
    background: #f2f2f2;
    font-size: 12px;
}

/*LOGO AND MAIN MENU*/
.logo-main-menu {
    height: 102px;
    background: khaki;
}
#logo {
    width: 280px;
    height: 102px;
    background: #ffb6a5;
}

/*ADVERTISEMENT AND CONTENT*/
/*ADVERTISEMENT*/
.bound-advertisement-content {
    background: #f2f2f2;
}

```

```

#advertisement {
    width: 980px;
    height: 335px;
    background: #8ef254;
}
/*CONTENT*/
#content {
    height: 400px;
    background: aquamarine;
    clear: both;
}

```

Блок новостей и блок предложений в шаблоне расположены рядом друг с другом, как вы знаете, блочный элемент заполняет всю ширину веб-браузера, даже если ему задать фиксированную ширину, соответственно следующий за ним блок будет расположен снизу. Обратите внимание на свойство float, оно позволяет сделать блок плавающим, т.е. не занимать все пространство.

```

#block-news {
    width: 320px;
    height: 400px;
    background: #f2af2d;
    float: left; /*делаем блок плавающим и прижимаем влево, следующий ПЛАВУЮЩИЙ
блок прижмется к нему*/
}
#block-offers {
    width: 600px;
    height: 400px;
    background: #8773f2;
    float: left; /*необходимо указать, так как этот блок является следующим за пла-
вающим (иначе он прижмется к родительскому элементу)*/
}

/*FOOTER*/
.bound-footer {
    background: #e6e6e6;
}

#footer {
    height: 296px;
}

#block-subscription {
    width: 320px;
    height: 296px;
    background: #f25f68;
    float: left;
}

#block-why-us {
    width: 200px;
    height: 296px;
    background: #f2c332;
    float: left;
}

```

```
#block-follow-us {  
  width: 200px;  
  height: 296px;  
  background: #94f2e5;  
  float: left;  
}  
  
#block-address {  
  width: 200px;  
  height: 296px;  
  background: #f28a22;  
  float: left;  
}  
  
.bound-below-footer {  
  height: 110px;  
  background: #454545;  
}
```

В результате, мы получили необходимую нам структуру, далее можно приступить к верстке отдельных блоков определенных дизайнером.

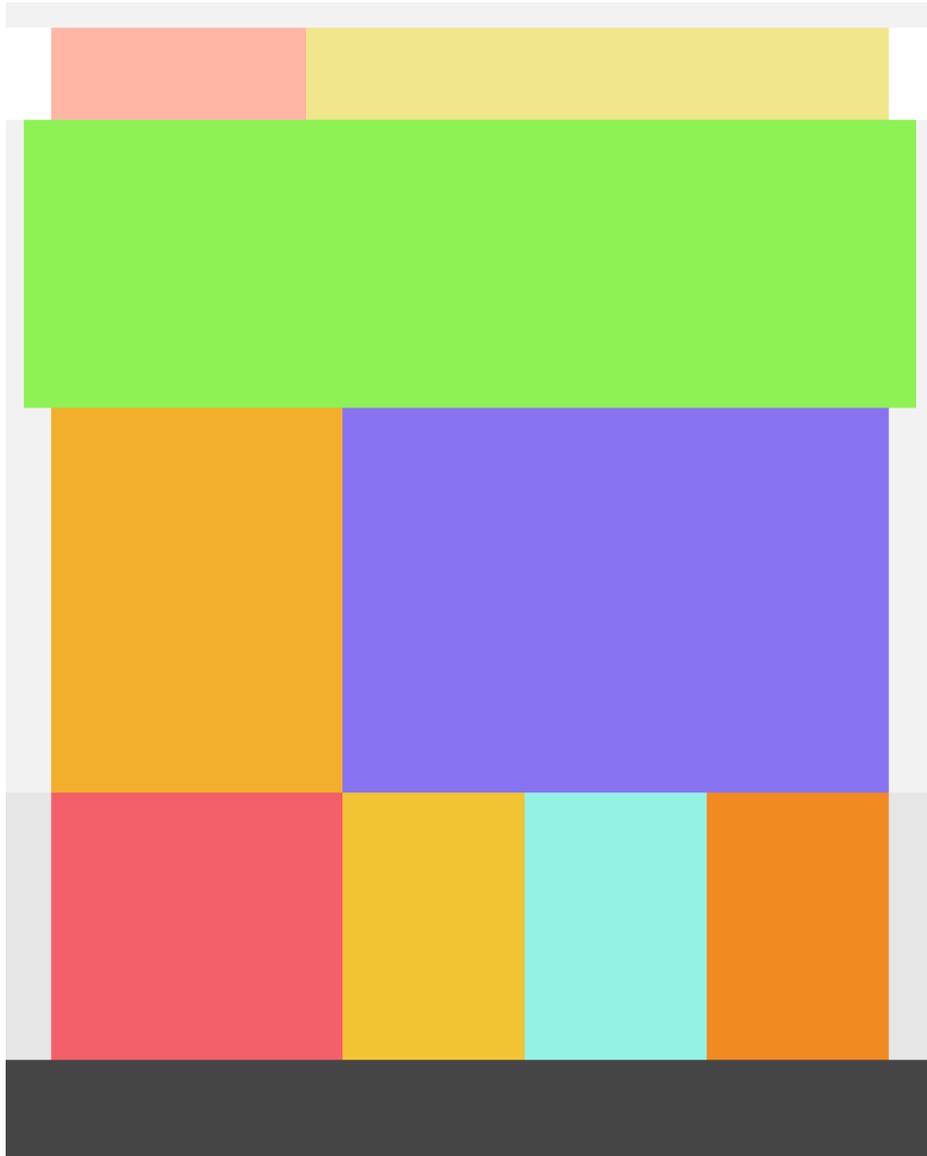


Рисунок 3 – Результат верстки

## 1.2. Блок быстрой навигации

Что касается блока навигации, его мы разобьем на два блока: блок с датой в левой части и блок с меню в правой части. Ниже приведен модифицированный участок кода, отвечающий за блок навигации.

```
<div id="top-navigation" class="wrapper">
  <div id="block-date">Friday, September 6, 2013 12:34</div>
  <div id="block-nav-menu">
    <ul>
      <li><a href="#">Support</a></li>
      <li><a href="#">FAQs</a></li>
      <li><a href="#">Sitemap</a></li>
    </ul>
  </div>
</div>
```

Если внимательно посмотреть на шаблон, то можно заметить что дата и меню располагаются по центру блока с навигацией, центрирование можно сделать при помощи внутренних отступов (padding). Задав одинаковое расстояние сверху и снизу, а также уменьшив высоту блока на суммарную величину отступа снизу и сверху, так как высота блока определяется как сумма внутренних, внешних отступов и размер границы слоя.

```
.bound-top-navigation {
  background: #f2f2f2;
}

#top-navigation {
  height: 14px; /*уменьшили высоту блока, на суммарный размер отступов*/
  background: #f2f2f2;
  padding: 7px 0; /*Добавляем отступы сверху и снизу, убираем отступы справа и слева*/
  font-size: 12px;
  color: #060606;
}

#block-date {
  float: left; /*Блок с датой размещаем слева*/
}

#block-nav-menu {
  float: right; /*Блок с меню размещаем справа*/
}
```

Обратите внимание на обращение к элементу ul, у него нет ни класса ни уникального идентификатора, но мы знаем, что он расположен внутри тега с идентификатором #block-nav-menu, данный стиль будет применен ко всем спискам расположенным внутри этого блока.

```
#block-nav-menu ul{
  margin: 0;
  padding: 0;
```

```
}
```

Как вы знаете, список это блочный элемент, чтобы расположить его элементы рядом воспользуемся свойством `float`. Мы используем маркированный список, но маркеры не предусмотрены в дизайне, соответственно чтобы убрать их необходимо воспользоваться свойством `list-style`. Чтобы создать разделитель между элементами списка, можно воспользоваться одним из вариантов:

- при помощи границ (`border-right`);
- при помощи установки фона с изображением разделителя (`background`).

Обратите внимание на псевдо свойство `last-child`, которое позволяет найти последний элемент списка, нам это понадобится, чтобы убрать разделитель у него.

Первый способ используется чаще, но он может не всегда подходить под дизайн, который определил дизайнер, результат работы следующего участка кода:

```
#block-nav-menu ul li{
    float: left;
    list-style: none;
    padding: 0 7px;
    border-right: solid 1px #060606; /*создаем границу справа*/
}

#block-nav-menu li:last-child{
    border-right: none; /*Убираем границу справа у последнего элемента*/
}
```

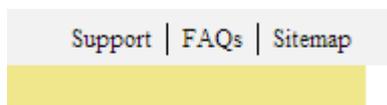


Рисунок 4 – Результат создания разделителя через границу

В нашем случае второй способ подходит больше, так как результат соответствует дизайну, но для реализации пришлось создать изображение разделителя размером: шириной `1px`, а высотой `7px`.

```
#block-nav-menu ul li{
    float: left;
    list-style: none;
    padding: 0 7px;
    background: url("images/nav_sep.gif") right 3px no-repeat; /*размещаем изображение разделителя справа и опускаем его на 3 пикселя вниз, указываем что фон повторять нет необходимости*/
}

#block-nav-menu li:last-child{
    background: none; /*Исключаем разделитель у последнего элемента*/
}
```

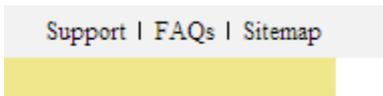


Рисунок 5 – Результат создания разделителя через фоновое изображение

```
#block-nav-menu li a {  
  text-decoration: none;  
  color: #060606;  
}  
#block-nav-menu li a:hover {  
  text-decoration: underline;  
}
```

Результат верстки верхнего меню представлен на рисунке



Рисунок 6 – Готовое меню навигации

Перейдем к верстке следующего блока, который содержит логотип и главное меню сайта.

### 1.3. Блок с логотипом и главным меню сайта

В данном блоке стоит уделить внимание именно созданию меню, так как размещение логотипа является простой задачей. Меню можно представить в виде маркированного списка (без маркера), как можно увидеть на рисунке, приведённом ниже, ширина элемента меню зависит от размера содержимого. Чтобы решить эту задачу необходимо добавить внутренние отступы со всех сторон. Так же, можно заметить, что элементы меню бывают двух типов, активные (при наведении или при нахождении на текущей странице – зеленого цвета) и пассивные (светло-серые).



Рисунок 7 – Верстка главного меню

Чтобы закруглить края необходимо использовать свойства `border-radius`, в котором необходимо указать радиус.



Рисунок 8 – Результат верстки блока

Измененный исходный код HTML документа приведен ниже

```
<div class="bound-logo-main-menu">
  <div class="wrapper logo-main-menu"> <!--Применяем сразу 2 стиля к одному элементу-->
    <div id="logo"></div> <!--Размещаем картинку с логотипом-->
    <div id="main-menu">
      <ul>
        <li class="active-element">Home</li>
        <li>Features</li>
        <li>Product</li>
        <li>Download</li>
        <li>Contacts</li>
      </ul>
    </div>
  </div>
</div>
CSS код

/*LOGO AND MAIN MENU*/
```

```

.logo-main-menu {
    width: 980px;
    padding-top: 19px; /*Зададим отступ сверху, чтобы логотип размещался по центру
блока*/
    height: 83px; /*соответственно уменьшим высоту блока (так как добавили от-
ступ)*/
    background: #fff;
}
#logo {
    margin-left: 30px;
    width: 205px;
    background: #92b800;
    float: left;
}

#main-menu {
    float: right;
    height: 83px;
}

```

В связи с тем, что элементы списка необходимо сделать плавающими при помощи свойства float и значения left, родительский элемент «ul» не может определить свой размер, для этого мы будем использовать свойство overflow и значение hidden. Это свойство управляет отображением содержания блочного элемента, если оно целиком не помещается и выходит за область заданных размеров и автоматически определяет размер блока.

```

#main-menu ul {
    margin-top: 18px; /*необходимо добавить отступ сверху, чтобы прижать меню к
нижнему блоку*/
    overflow: hidden;
    padding: 0;
    margin-bottom: 0;
}
#main-menu ul li{
    float: left;
    list-style: none;
    background: #f2f2f2; /*задаем серый цвет для всех элементов списка*/
    padding: 24px 32px 21px 32px; /*добавляем внутренние отступы, как показано на
рисунке Рисунок 7*/
    margin-right: 1px;
    /*Добавляем закругления на углах блока*/
    border-top-left-radius: 20px;
    border-top-right-radius: 20px;
}

```

Для начала определим класс, который будет предназначен для активных элементов главного меню – зададим зеленый цвет и белый шрифт. Познакомимся еще с одним псевдо тегом (селектором) hover, который осуществляет выборку элементов, на который сейчас наведена мышка, чтобы применить один стиль сразу для двух классов их необходимо перечислить через запятую.

```

#main-menu ul li:hover,
#main-menu ul li.active-element {
    background: #92b800;
    color: #fff;
}

```

}

#### **1.4. Рекламный блок**

~~Верстку рекламного блока мы не будем рассматривать, так как блок представляет собой картинку, а поместить картинку внутри блока у Вас не составит труда.~~

## 1.5. Новостной блок

Для того чтобы приступить к верстке следующего блока, нужно представить, как будет осуществляться разбиение новости на составные части (рисунок 9). Затем, разработанные стили будут применяться одновременно ко всем новостям, которые будут находиться внутри этого блока.



Рисунок 9 – Разбиение новости

HTML и CSS коды блока новостей представлены ниже. Чтобы не загромождать страницу исходным кодом, ниже приведен только участок верстки первой новости, чтобы создать вторую необходимо продублировать данный участок HTML-кода, стиль к ним применится автоматически.

```
<div id="block-news">
  <h1>News & Events</h1>
  <div class="news">
    <div class="news-image">
      
    </div>
    <div class="news-info">
      <div class="news-date">Tue, June 7, 2011</div>
      <div class="news-content">
        Name libero tempore csoluta nobis eieligendi opro cumque nihil
        impedit quo
      </div>
    </div>
  </div>
</div>
```

В таблице стилей были использованы свойства, которые Вам знакомы и не требуют комментариев.

```
/*NEWS*/
#block-news {
  width: 320px;
  height: 400px;
  float: left; /*делаем блок плавающим и прижимаем влево, следующий ПЛАВУЮЩИЙ
блок прижмется к нему*/
}

#block-news .news {
  overflow: hidden;
```

```

    margin-top: 15px;
}

#block-news .news-image {
    float: left;
    width: 90px;
    height: 90px;
    margin-right: 20px;
}

#block-news .news-info {
    float: left;
    width: 175px;
    height: 90px;
}

#block-news .news-date {
    font-size: 14px;
    color: #000700;
    margin-top: 8px;
}

#block-news .news-content {
    font-size: 14px;
    color: #84807d;
    margin-top: 8px;
}

```



Рисунок 10 – Результат верстки

Следующим по очереди будет блок с предложениями, который расположен справа от блока с новостями.

## 1.6. Блок с предложениями

Принцип верстки этого блока похож на верстку блока с новостями, акцент сделаем на кнопку «Read More». В данном разделе мы рассмотрим несколько способов верстки данной кнопки. Первый способ самый простейший представить кнопку в виде фона для блока, на котором поместить текст:



Рисунок 11 – Кнопка без текста

```
<div class="offer-bound">
  <h1>What We Offer</h1>
  <div class="offer">
    <div class="offer-image">
      
    </div>
    <div class="offer-info">
      <div class="offer-title">This is one of free website templates created
by www.template.azhark.com</div>
      <div class="offer-content">This Pro Soft template is optimized for
1024x768 screen resolution. It has several pages: Home Page. Features. Support.
Downloads. Contacts. (note that contact us from - doesn`t work).</div>
      <div class="button">Read More</div> <!--Кнопка представляет собой слой,
с задним фоном-->
    </div>
  </div>
</div>
```

```
.button {
  padding: 10px 25px 16px 28px; /*Добавляем отступы – выравниваем текст внутри
кнопки*/
  width: 74px;
  height: 16px;
  float: right; /*Смещаем кнопку вправо*/
  font-weight: bold;
  font-size: 14px;
  background: url("images/bg-button.png") 0 0 no-repeat; /*Устанавливаем изобра-
жение кнопки фоном*/
}
```

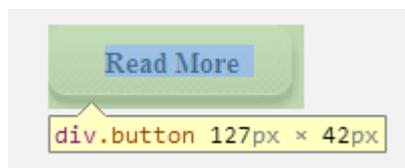


Рисунок 12 – Результат верстки

Данное решение является кросс-браузерным и вызовет меньше проблем при верстке, но она не является универсальным. Если придется поместить другой текст

внутри кнопки, то ширина кнопки не будет меняться, поэтому рассмотрим следующий способ:

Данный способ подразумевает верстку полностью при помощи CSS, особенность заключается в том, что мы создали изображение шириной в один пиксель и высотой с размер кнопки, которое содержит только фон кнопки (вырезали из шаблона) и поставили его на задний фон. Чтобы изображение дублировалось по всей ширине блока, необходимо добавить атрибут `repeat-x` у свойства `background`.



Рисунок 13 – Результат верстки

Далее необходимо закруглить края изображения и наложить тень, самое слабое место в данном подходе, это тень, к сожалению, в старых браузерах, таких как Firefox 3.5 или IE 6,7 тень будет отсутствовать, но так как процент использования этих браузеров очень низок, этим можно пренебречь.

Как говорится в лекциях, не все браузеры придерживаются стандарту установленным W3C и начали придумывать свои свойства. Например, чтобы тень отображалась корректно в старых версиях Firefox необходимо использовать свойство `-moz-box-shadow`, для Safari это свойство будет `-webkit-box-shadow`. Но сейчас, в последних версиях браузеров достаточно использовать `box-shadow`, но и про кроссбраузерность забывать нельзя.

```
.button {  
    padding: 8px 25px 10px 28px;  
    height: 17px;  
    float: right;  
    font-weight: bold;  
    font-size: 14px;  
    background: url("images/bg-button-gradient.png") 0 0 repeat-x;  
    border-radius: 8px;  
    -webkit-box-shadow: 0px 3px 5px #A29F9F;  
    -moz-box-shadow: 0px 3px 5px #A29F9F;  
    box-shadow: 0 3px 5px #A29F9F;  
}
```

Можно шагнуть еще дальше, и вообще не использовать фоновое изображение для кнопки, можно задать фон при помощи стандартных свойств CSS, которые позволяют создавать градиент, но этот случай мы рассматривать не будем.

## 1.7. Блок подписки на рассылку

В блоке с подпиской на рассылку используется текстовое поле ввода, которое абсолютно отличается от стандартного поля ввода. Поэтому самым оптимальным способом верстки будет являться следующий:

Необходимо из шаблона вырезать графическое представление поля ввода (рисунок 14) и использовать его в качестве фона для блока, который будет содержать в себе поле ввода. Самое главное убрать абсолютно все стили для поля ввода: границы, отступы, фон, тогда оно будет абсолютно не видимым, а внешний вид будет задаваться фоновым изображением блока.



Рисунок 14 – Текстовое поле разработанное дизайнером

С созданием кнопки у нас нет проблем, так как ранее мы придумали универсальный стиль, который можно с легкостью применять для любого блока.

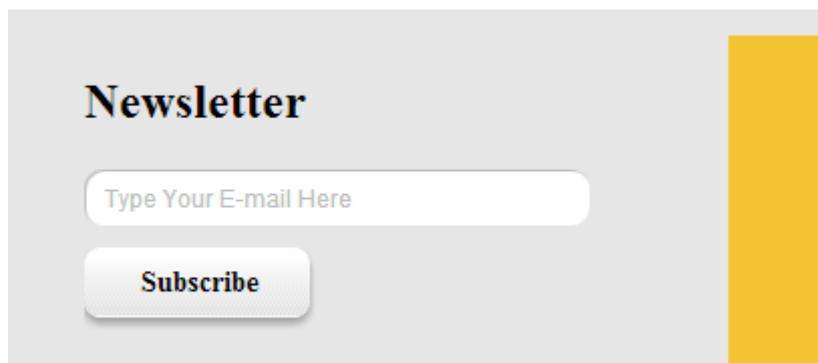


Рисунок 15 – Результат верстки

```
<div id="block-subscription">
  <h2>Newsletter</h2>
  <form method="POST" class="subscription-form">
    <div class="bg-input">
      <input class="form-input-text" type="text" name="e-mail" value="Type
Your E-mail Here" /> <!--Создаем поле ввода, value указывает текст внутри поля
ввода по умолчанию-->
    </div>
    <div class="button subscribe-button">Subscribe</div>
  </form>
</div>
```

```
#block-subscription .form-input-text {
  background: none;
  padding: 7px 10px;
  width: 251px;
  font-size: 12px;
  font-family: Arial,Helvetica,sans-serif;
  color: #c0c0c0;
```

```
    /*height: 14px;*/  
    border: none;  
    margin: 0;  
}  
  
#block-subscription .subscribe-button {  
    margin-top: 10px;  
    margin-bottom: 10px;  
    float: left;  
}  
  
#block-subscription .bg-input {  
    background: url("images/bg-newsletter-input.png") 0 0 no-repeat;  
}  
  
#block-subscription .subscription-form {  
    overflow: hidden;  
}
```

Мы с Вами подошли к заключительной части, чтобы шаблон был до конца сверстан, необходимо доработать три блока, что я предлагаю сделать Вам самостоятельно. На данном этапе веб-страница выглядит следующим образом:

## Best Professional Soft

Sed ut perspiciatis unde omnis natus error volup accusantium doloremque laudantium, totaem aperiam, eaque ipsa quae ab illo inventore.

[MORE INFO](#)

### News & Events



Tue, June 7, 2011

Name libero tempore csoluta nobis eieligendi opro cumque nihil impedit quo



Tue, June 7, 2011

Name libero tempore csoluta nobis eieligendi opro cumque nihil impedit quo



Tue, June 7, 2011

Name libero tempore csoluta nobis eieligendi opro cumque nihil impedit quo

### What We Offer



This is one of free website templates created by [www.template.azhark.com](http://www.template.azhark.com)

This Pro Soft template is optimized for 1024x768 screen resolution. It has several pages: Home Page, Features, Support, Downloads, Contacts. (note that contact us from - doesn't work).

[Read More](#)

### Why Do You Need It?!



This is one of free website templates created by [www.template.azhark.com](http://www.template.azhark.com)

This Pro Soft template is optimized for 1024x768 screen resolution. It has several pages: Home Page, Features, Support, Downloads, Contacts. (note that contact us from - doesn't work).

[Read More](#)

### Newsletter

[Subscribe](#)

Рисунок 16 – Результат верстки

#### **4 Задание на лабораторную работу**

1. Изучить рассмотренный пример в методическом пособии в браузере. Чтобы посмотреть исходный код страницы в веб-браузере Chrome необходимо нажать F12.
2. Доработать верстку шаблона (последние три блока);
3. Сверстать полученный от преподавателя шаблон (срок выполнения лабораторной работы 2 недели), убедиться, что шаблон отображается одинаково во всех популярных браузерах. Подготовить отчет, описывающий подробный процесс верстки шаблона веб-страницы.

Приложение 1  
(обязательное)  
Исходный шаблон для верстки



Учебное издание

САВЕНКО Игорь Игоревич

## Верстка шаблона веб-сайта

Методические указания к выполнению лабораторных работ  
по курсу «Интернет-технологии»  
для студентов направлений 09.03.02 «Информационные системы и тех-  
нологии»

Отпечатано в Издательстве ТПУ в полном соответствии  
с качеством предоставленного оригинал-макета

Подписано к печати 00.00.2013. Формат 60x84/16. Бумага «Снегурочка».  
Печать XEROX. Усл. печ. л. 9,01. Уч.-изд. л. 8,16.  
Заказ 000-13. Тираж 100 экз.



Национальный исследовательский Томский политехнический университет  
Система менеджмента качества  
Издательства Томского политехнического университета сертифицирована  
NATIONAL QUALITY ASSURANCE по стандарту BS EN ISO 9001:2008



ИЗДАТЕЛЬСТВО  ТПУ . 634050, г. Томск, пр. Ленина, 30  
Тел./факс: 8(3822)56-35-35, www.tpu.ru