

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования

АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики

Кафедра информационных и управляющих систем

Гетман А. Н.

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

«ПРОЕКТИРОВАНИЕ ИНТЕРНЕТ-МАГАЗИНА».

РАЗДЕЛ 1. «ВЕРСТКА»

Благовещенск 2021

Печатается по решению
редакционно-издательского совета
факультета математики и информатики
ФГБОУ ВО «Амурский государственный
Университет»

Лабораторный практикум «Проектирование интернет-магазина». Раздел 1. «Верстка»: учеб. пособие по дисциплине «Проектирование интерфейса человек-компьютер» для бакалавров направления подготовки 09.03.01 «Информатика и вычислительная техника», направленность (профиль) образовательной программы «Автоматизированные системы обработки информации и управления» / Амур. гос. ун-; А. Н. Гетман – Благовещенск: АмГУ - 2021. – 140 с.

Автор: Гетман Андрей Николаевич, к.т.н., доцент кафедры информационных и управляющих систем ФГБОУ ВО «Амурский государственный университет»

Учебное пособие представляет собой первую часть (раздел 1. Верстка) полноценного лабораторного практикума проектирования интернет-магазина (состоящего из 3-х разделов), написанного на языке web-программирования php с использованием СУБД mysql. При проектировании интернет-магазина использован современный паттерн (шаблон проектирования) MVC (модель-вид-контроллер).

Пособие рекомендуется студентам, изучающим курсы web-программирования с использованием языка разметки html, каскадных таблиц стилей css, языка web-программирования php, СУБД mysql и сервера Apache.

Рецензенты:

Е.Ф. Попова, к.т.н., доцент кафедры информатики и методики преподавания информатики ФГБОУ ВО БГПУ;

А.Г. Масловская, д.ф.-м.-н., профессор кафедры математического анализа и моделирования ФГБОУ ВО АмГУ.

© Амурский государственный университет, 2021

© Гетман А. Н., автор

ВВЕДЕНИЕ

Лабораторный практикум «Проектирование интернет-магазина» состоит из трех разделов:

- Раздел 1. «Верстка»;
- Раздел 2. «Программирование видимой части»;
- Раздел 3. «Программирование администраторской части».

Раздел 1. «Верстка» состоит из 7-ми лабораторных работ, посвященных последовательному проектированию html-шаблона интернет-магазина, который будет основой для программирования его видимой (пользовательской) и администраторской частей.

Особенностью первого раздела «Верстка» практикума является то, что создание html-шаблона интернет-магазина выполняется на основании дизайн-проекта, предоставляемого студентам в качестве набора страниц, выполненных в программе Фотошоп. При верстке сайта студенты стремятся наиболее точно приблизить вид страниц сайта к дизайн-проекту. Именно так и происходит реальное проектирование сайтов интернет-магазинов.

Предметной областью интернет-магазина является продажа сотовых телефонов. Для выполнения первого раздела «Верстка» лабораторного практикума необходимы начальные знания в области верстки сайтов, а именно языка разметки гипертекста html и каскадных таблиц стилей css.

Студенты, приобретая навыки программирования интернет-магазина сотовых телефонов, смогут использовать свои знания web-программирования при выполнении курсового проекта по дисциплине «Проектирование интерфейса человек-компьютер» в другой предметной области, при выполнении выпускной квалификационной работы и в дальнейшей своей деятельности после окончания вуза на предприятиях и в организациях нашей страны.

Лабораторная работа № 1

Введение. Разметка главной страницы. Создание шапки сайта.

Приступаем к созданию проекта интернет-магазина по продаже сотовых телефонов.

Вначале следует отметить, что для создания полноценного проекта интернет-магазина требуется наличие команды, в которую входят:

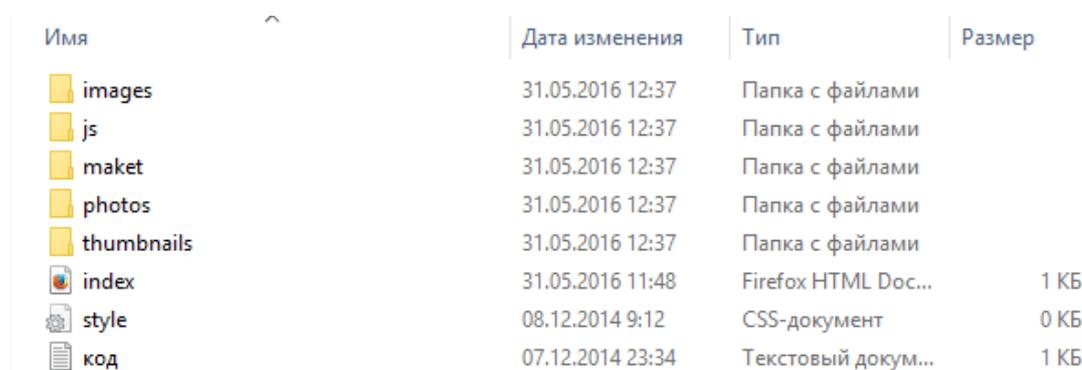
- 1) Руководитель проекта - кроме непосредственного руководства работой команды он, как правило, берет на себя переговоры с клиентом и юридическое оформление договоров;
- 2) Дизайнер сайта - занимается разработкой внешнего вида сайта (обычно дизайн сайта выполняется в программе Photoshop);
- 3) Верстальщик сайта – выполняет верстку сайта в кодах html (или xhtml) и css. Вместо работающих ссылок ставятся «заглушки» (ссылки, не ведущие ни куда – временно);
- 4) Программист – создает базу данных проекта и подключает созданный верстальщиком макет сайта к базе данных, то есть наполняет сайт реальным функционалом.

Пропустим работу руководителя проекта по поиску заказчика и выявлению его требований к будущему проекту сайта интернет-магазина. Также пропустим этапы заключения договоров. Итак, у команды есть договор с клиентом и утвержденное обеими сторонами техническое задание. Далее на основе технического задания дизайнер создает проект внешнего вида сайта. Данный проект утверждается заказчиком и является исходным пунктом для работы верстальщика.

Начинаем работать в качестве верстальщика сайта интернет-магазина сотовых телефонов. Для выполнения лабораторных работ Вам предоставлена папка «**dopmat**» на файловом сервере «**public**» в папке Вашей группы (или в системе Moodle). Окончательный адрес папки «**dopmat**» Вам подскажет

преподаватель. В папке «**dopmat**» содержатся требуемые для выполнения лабораторных работ материалы, откройте папку «**dopmat**», далее папку «**Для лабы 1**» и скопируйте на внешний USB-носитель (флэш-накопитель) папку «**ishop**» - это будет Ваша основная рабочая папка. Расположение рабочей папки на флешке позволит не зависеть от конкретного компьютера (он может сломаться) или компьютерного класса, в котором проводятся лабораторные работы. Другие необходимые для работы файлы и коды программ выбирайте из соответствующих папок «**Для лабы N**», где **N** – номер выполняемой лабораторной работы.

Итоговое содержание папки «**ishop**» показано на рисунке 1.1.



Имя	Дата изменения	Тип	Размер
images	31.05.2016 12:37	Папка с файлами	
js	31.05.2016 12:37	Папка с файлами	
maket	31.05.2016 12:37	Папка с файлами	
photos	31.05.2016 12:37	Папка с файлами	
thumbnails	31.05.2016 12:37	Папка с файлами	
index	31.05.2016 11:48	Firefox HTML Дос...	1 КБ
style	08.12.2014 9:12	CSS-документ	0 КБ
код	07.12.2014 23:34	Текстовый докум...	1 КБ

Рисунок 1.1 – Содержание папки **ishop**

Главная страница сайта – файл **index.html**. Данный файл имеет начальный код, содержащий ДТД (декларацию типа документа), мета-теги кодировки сайта (мы будем использовать UTF-8), заголовок сайта и теги страницы (<html>) и «тела» страницы(<body>). В папке **images** содержатся файлы картинок, необходимых для оформления страниц сайта. Обычно картинки для оформления сайта выполняются в программе Photoshop, но мы будем считать, что нам их передал дизайнер сайта вместе с макетом.

Кроме того, Вам представлена папка **maket**, в которой приведены результаты работы дизайнера – файлы внешнего вида разных страниц сайта в формате **.psd** (Фотошоп). Именно такого вида сайта мы будем добиваться в

процессе верстки сайта. Для тех студентов, у которых на компьютерах не установлен Фотошоп (или его аналог), данные файлы дублированы в формате **.jpg**.

Файл стилей **style.css** пока пуст – нам его еще предстоит заполнить. Папка **js** содержит java-скрипты, необходимые нам для оформления страниц (например, в блоке поиска по сайту – кнопка «Найти» заголовка главной страницы сайта). И, наконец, папки **photos** и **thumbnails** содержат соответственно фотографии моделей телефонов (выводимых на главной странице сайта) и эскизы фотографий телефонов для слайдера страницы подробного просмотра товаров.

Файл **код.txt** содержит код java-скрипта для создания блока поиска по сайту в блоке «Хедер».

Для вёрстки нам понадобятся две программы: любой **браузер** (мы будем проверять вёрстку в нескольких) и блокнот с подсветкой синтаксиса **Notepad++**.

Откройте в просмотрщике графических файлов (вьювере) файл **главная.jpg** (из папки **maket**), содержащий будущий внешний вид главной страницы сайта - файла **index.html**. Внешний вид макета главной страницы сайта в окне браузера приведен на рисунке 1.2.

Структура разметки главной страницы сайта приведена на рисунке 1.3. Как видно из рисунка, глобально главная страница состоит из 5-ти блоков:

- 1) Блок «**main**», охватывающий всю страницу (выделен красным цветом);
- 2) Блок «**header**» (заголовок сайта с логотипом, слоганом и блоком контактов). Выделен зеленым цветом;
- 3) Блок «**menu**», представляющий собой горизонтальное меню (выделен желтым цветом);
- 4) Блок «**content**» (выделен коричневым цветом), в котором показан основной контент (содержимое) сайта. Как видно, блок контент выполнен трехколоночной версткой (чтобы убедиться в этом,

рассмотрите файлы «каталог_линейный_вид.jpg» и «каталог_табличный_вид.jpg»;

5) Блок «**footer**» - «подвал сайта» (выделен фиолетовым цветом).

Внутри данных блоков будут находиться другие блоки, которых мы коснемся тогда, когда будем оформлять соответствующий глобальный блок.

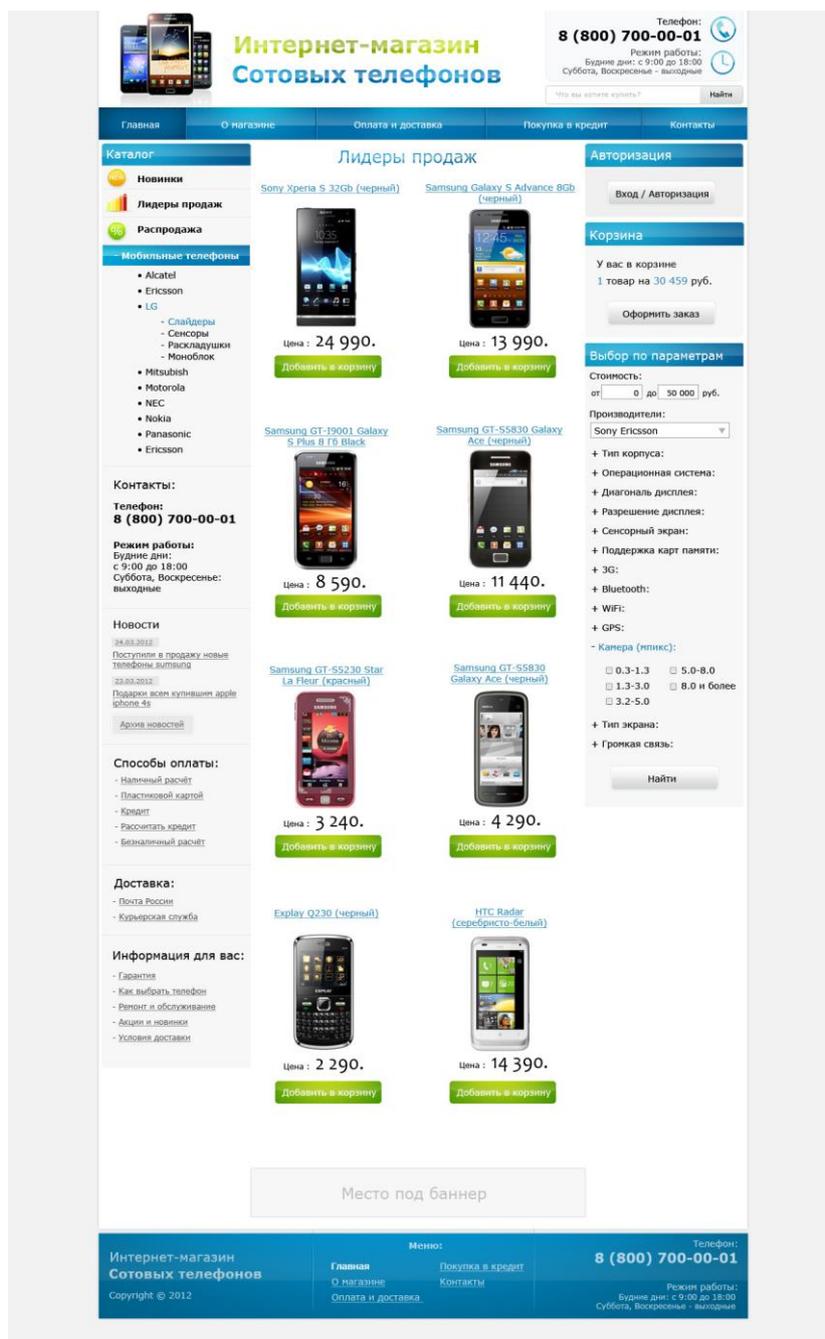


Рисунок 1.2 – Макет главной страницы сайта

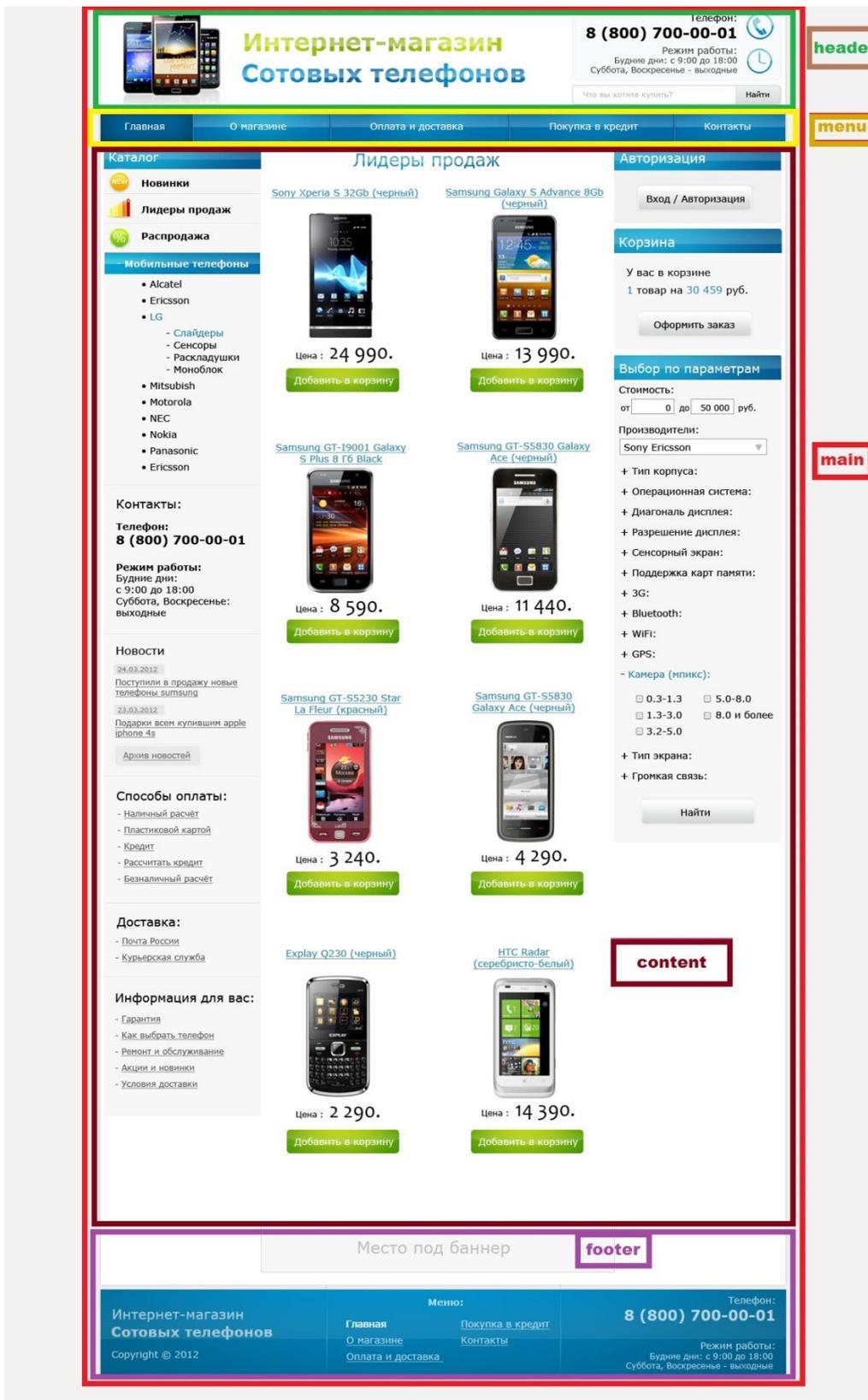


Рисунок 1.3 – Разметка блоков главной страницы

Приступаем к созданию главной страницы сайта. Сайт будет кроссбраузерным (одинаково будет выглядеть во всех браузерах), полурезиновым (будет менять свой размер в зависимости от размеров экрана браузера, однако пределы изменения размеров страниц сайта будут ограничены) и хорошо масштабируемым.

В коде главной странице сайта мы будем писать только html-код страницы, а все оформление передадим в файл стилей (style.css).

Главную страницу сайта (и остальные страницы тоже) будем выполнять на контейнерах **div**.

Откройте в блокноте Notepad++ файл **index.html** и впишите код, выделенный красной рамкой (рисунок 1.4).

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script type="text/javascript" src="js/functions.js"></script>
<!--[if lt IE 9]>
<script src="http://ie7-js.googlecode.com/svn/version/2.1(beta4)/IE9.js"></script>
<![endif]-->
<link rel="stylesheet" type="text/css" href="style.css" />
<title>Интернет-магазин сотовых телефонов</title>
</head>
<body>
<div class="main">
  <div class="header">
    </div> <!-- header -->
  </div> <!-- main -->
</body>
</html>
```

Рисунок 1.4 – Основная разметка главной страницы **index.html**

Между тегами **<title>** **</title>** мы вставили название (заголовок) сайта, которое теперь будет отражаться во вкладке браузера.

Далее мы создали контейнеры для сайта и заголовка. После закрывающих тегов **</div>** мы поместили комментарии вида **<!-- header -->**. Это очень важно, даже обязательно. Нам придётся вставлять новые дивы в определённые места,

часто перед вполне определёнными закрывающими тегами. А их на главной странице будет более десятка.

Приступаем к созданию заголовка (шапки) сайта. Шапка сайта, в свою очередь, также имеет блочную структуру, представленную на рисунке 1.5.



Рисунок 1.5 – Структура блока «header»

Как видно из рисунка, блок «header» состоит из 4-х блоков, выделенных красным цветом:

- 1) Логотип сайта;
- 2) Слоган сайта («Интернет-магазин сотовых телефонов»);
- 3) Блок контактов (телефон и режим работы);
- 4) Блок поиска.

Для заголовка сайта воспользуемся четырьмя рисунками: **logo.jpg** (логотип сайта), **slogan.jpg** (слоган сайта), **phone-img.jpg** (телефон и часы в блоке контактов) и **search-btn.jpg** (кнопка «найти»). Эти рисунки уже подготовлены и находятся в папке **images**. Откройте ее и просмотрите данные рисунки.

В блокноте откройте файл **index.html** и внесите текст, выделенный красной рамкой, на рисунке 1.6.

Рассмотрим структуру внесенного кода. Первые две строчки – это картинки «лого» и «слоган». Причем картинка «лого» – логотип сайта – является гиперссылкой на корень сайта ``. Корнем сайта называется файл **index.html**, рядом с которым располагаются папки с картинками, скриптами и т.п. То есть это наш файл, с которым мы сейчас работаем.

```
<div class="header">
```

```
  <a href="/"></a>
  
  <div class="head-contact">
    <p><strong>Телефон:</strong><br />
    <span>8 (800) 700-00-01</span>
    </p>
    <p><strong>Режим работы:</strong><br />
    Будние дни: с 9:00 до 18:00 <br />
    Суббота, Воскресенье - выходные
    </p>
  </div> <!-- head-contact -->
  <form method="get" action="">
    <ul class="search-head">
      <li> <input type="text" name="search" /> </li>
      <li> <input type="image" class="search-btn" src="images/search-btn.jpg" /></li>
    </ul>
  </form>
</div> <!-- header -->
```

Рисунок 1.6 – Код заголовка главной страницы в файле **index.html**

Далее идет контейнер `<div class="head-contact">` блока контактов. В этом контейнере содержится два абзаца (параграфа) `<p>`.

В первом абзаце приведен телефон организации, во втором абзаце – описание режима работы. Для первого абзаца мы использовали тег ``. В обычном понимании браузера этот тэг придает тексту жирный шрифт, однако мы в файле стилей придадим тексту новые свойства.

Во втором абзаце использован тег `` - сам по себе это «пустой» тэг (вроде `<div>`-а, только `<div>` - это блочный элемент (он стремится занять всю строку целиком), а `` - строчный элемент, он располагается внутри отдельной строки), в файле стилей мы придадим ему вполне определённые значения текста.

Последним идет форма. Форма выполнена в виде нумерованного списка `` (`unordered list` - нумерованный список) с отдельными элементами списка `` (`list item` - пункт списка). Первым элементом списка идет поле ввода текста поискового запроса. Вторым элементом – картинка с надписью «найти».

Всем картинкам и списку присвоены свои классы. Они позволят в дальнейшем произвести требуемое оформление элементов хедера.

Посмотрите результат в браузере (рисунок 1.7).

В дальнейшем, если в тексте лабораторной работы будет встречаться текст *«Посмотрите результат в браузере»*, следует понимать, что нужно выполнить два действия:

- 1) сохранить код файла в блокноте Notepad++, нажав на кнопку с дискетой (или с несколькими дискетами  на панели инструментов блокнота Notepad++) или воспользовавшись «горячими клавишами» CTRL+S;
- 2) открыть файл в браузере или, если он был ранее открыт, обновить страницу (можно нажатием кнопки F5 на клавиатуре).

Как видно из рисунка, заголовок сайта еще далек от требуемого вида.



Телефон:

8 (800) 700-00-01

Режим работы:

Будние дни: с 9:00 до 18:00

Суббота, Воскресенье - выходные

-
-

Рисунок 1.7 - Первичный вид заголовка сайта

Теперь мы должны подключить скрипт для окна поиска. В материалах к лабораторной работе Вам дан файл «код.txt». Скопируйте код данного файла и вставьте скрипт вместо первого блока списка `...` (рисунок 1.8).

```
<ul class="search-head">
<li>
<input type="text" name="search" id="quickquery" placeholder="Что вы хотите купить?" />
<script type="text/javascript">
//
placeholderSetup('quickquery');
//]]&gt;
&lt;/script&gt;
&lt;/li&gt;
&lt;li&gt; &lt;input type="image" class="search-btn" src="images/search-btn.jpg" /&gt;&lt;/li&gt;
&lt;/ul&gt;</pre></div><div data-bbox="338 768 658 788" data-label="Caption"><p>Рисунок 1.8– Код тела скрипта</p></div><div data-bbox="89 825 908 875" data-label="Text"><p>Посмотрите результат в браузере (рисунок 1.9). Браузер может заблокировать выполнение скрипта – разблокируйте его в этом случае.</p></div><div data-bbox="484 893 513 911" data-label="Page-Footer"><p>13</p></div>
```



Интернет-магазин Сотовых телефонов

Телефон:
8 (800) 700-00-01

Режим работы:
Будние дни: с 9:00 до 18:00
Суббота, Воскресенье - выходные

-
-

Рисунок 1.9 - Результат включения скрипта в код страницы

Как видно из рисунка 1.9, в поле ввода появилась строка «Что вы хотите купить?». Можно внести текст в поле ввода, затем стереть его – слова «Что вы хотите купить?» должны появиться снова. Значит, java-скрипт работает. Функционал скрипта будет подключаться позднее, на этапе программирования.

Для проверки правильности написания кода на рисунке 1.10 приведен окончательный (для данной лабораторной работы) код файла **index.html**.

Вообще мы будем выполнять лабораторные работы следующим образом: вначале мы пишем код web-страницы, убеждаемся в наличии всех необходимых элементов, а затем приступаем к написанию стилей оформления документов.

Приступаем к форматированию оформления внешнего вида шапки сайта, для чего создан файл **style.css**. Откройте в блокноте файл **style.css** и напишите в коде файла следующий код (рисунок 1.11).

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script type="text/javascript" src="js/functions.js"></script>
<!--[if lt IE 9]>
<script src="http://ie7-js.googlecode.com/svn/version/2.1(beta4)/IE9.js"></script>
<![endif]-->
<link rel="stylesheet" type="text/css" href="style.css" />
<title>Интернет-магазин сотовых телефонов</title>
</head>
<body>
<div class="main">
  <div class="header">
    <a href="/"></a>
    
    <div class="head-contact">
      <p><strong>Телефон:</strong><br />
      <span>8 (800) 700-00-01</span>
      </p>
      <p><strong>Режим работы:</strong><br />
      Будние дни: с 9:00 до 18:00 <br />
      Суббота, Воскресенье - выходные
      </p>
    </div> <!-- head-contact -->
    <form method="get" action="">
      <ul class="search-head">
        <li>
          <input type="text" name="search" id="quickquery" placeholder="Что вы хотите купить?" />
          <script type="text/javascript">
            //<![CDATA[
            placeholderSetup('quickquery');
            //]]>
          </script>
        </li>
        <li> <input type="image" class="search-btn" src="images/search-btn.jpg" /></li>
      </ul>
    </form>
  </div> <!-- header -->
</div> <!-- main -->
</body>
</html>

```

Рисунок 1.10 – Окончательный код файла **index.html**

```
*{
    margin:0; padding:0;
}
img{
    border:none;
}
body{
    background:#f2f2f2;
}
```

Рисунок 1.11 – Код файла **style.css**

Рассмотрим код файла стилей. Для начала мы скинули в ноль все отступы всех блоков (параметры **margin** и **padding** для универсального селектора ***** - то есть для всех элементов оформления страницы). Далее убираются рамки у картинок – селектор **img**. Дело в том, что некоторые браузеры, в первую очередь Internet Explorer (и особенно в версиях меньше 8-й) отражают цветную рамку вокруг гиперссылки – в данном случае это картинка – логотип сайта. В результате в браузере IE вокруг логотипа сайта исчезнет рамка.

Теперь разберемся с цветом фона страницы (так называемой подложкой или фона сайта - **background**) – светло-серо-бежевый цвет (задается значением **#f2f2f2**). Данный цвет можно определить в программе Adobe Photoshop или в одной из специальных программ, предназначенных для определения цвета небольшого фрагмента изображения путем выделения цветового поля элементом типа «Пипетка» и отображением кода цвета в специальном окне.

Посмотрите результат в браузере. За границей сайта подложка должна принять светло-серо-бежевый цвет. Содержимое сайта прижмется к левому и верхнему краям браузера. Кроме того, исчезли точки у пунктов списка формы – теперь поле ввода текста «Что вы хотите купить» и кнопка «Найти» прижались к левому краю браузера вплотную.

Добавьте в файл **style.css** код в соответствии с рисунком 1.12.

```

.main{
  background:#fff; min-width:980px; max-width:1200px; margin:0 auto;
}
  .header{
    overflow:hidden;
  }
    .logo, .slogan{
      float:left;
    }
    .logo{
      margin:8px 0 0 30px;
    }
    .slogan{
      margin:40px 0 0 30px;
    }
  }
}

```

Рисунок 1.12 – Код файла **style.css** (продолжение)

Обратите внимание на стиль написания кода файла **style.css**. Он выполнен «ступеньками» - вначале «главный класс» (**main**), затем – с отступом вправо – все стили элементов, входящих в контейнер **main** (в данном случае – элемент **header**), затем - с отступом вправо - стили элементов, входящих в элемент **header**. И так далее. В таком представлении кода файл стилей читается очень легко, так же легко осуществлять поиск стиля для нужного элемента.

Фон блока **main** – белый (**#fff**). Минимальная ширина страницы – 980 пикселей, максимальная – 1200 пикселей (именно это даёт эффект так называемой «полурезиновой вёрстки»). Отступы (**margin**) – сверху и снизу – 0, по краям – автоматически (**auto**).

Стиль класса **.header** выключает «схлопывание» цвета фона под картинкой логотипа. Классы **logo** и **slogan** выравниваем по левому краю. Если посмотреть макет главной страницы сайта в программе Adobe Photoshop, то можно увидеть, что логотип отступает от верха страницы примерно на 8 пикселей, а слева – на 30 пикселей. Именно эти *отступы (по часовой стрелке: сверху-справа-снизу-слева)* и прописаны в классе стилей параметром **margin:8px 0 0 30px**. Аналогично отступы определяются в программе Adobe

Photoshop и кодируются для класса **slogan** (в отличие от класса **logo** у **slogan-a** отступ сверху у него 40px).

Посмотрите результат в браузере (рисунок 1.13).



Рисунок 1.13 - Результат применения стилей для логотипа и слогана

Итак, логотип и слоган теперь выровнены примерно так, как это представлено в макете сайта. Теперь займёмся правой частью шапки сайта. Вначале разберемся с блоком контактов. На рисунке 1.14 приведен код стилей для блока контактов (класс **.head-contact**) и блока поиска (класс **.search-head**).

Первая строка стилей класса **.head-contact** выравнивает блок по правой части страницы, затем дает нам цвет фона блока контактов, а также фоновый рисунок (телефонная трубка и часы – содержится в файле **phone-img.jpg**) с отступами от краев. Вторая строка говорит о наличии рамки вокруг блока контактов (рамка достаточно хорошо видна при большом увеличении картинки главной страницы в папке **maket**). Кроме того, в нижней части рамка имеет *закругления* с радиусом примерно 10 пикселей (задаётся параметром **border-radius** и также по часовой стрелке начиная с верхней грани блока **div**). Третья строка дает нам выравнивание текста по правой стороне и отступы текста от краев блока.

```

.head-contact{
    float:right; background:#f8fafb url(images/phone-img.jpg) 250px 8px no-repeat;
    border:2px solid #edf2f6; border-radius:0 0 10px 10px; text-align:right; margin:0 6px 0 0;
    padding:5px 65px 5px 15px;
    }
.head-contact p{
    font-size:12px; font-family:Verdana, Geneva, sans-serif; line-height:1.3; margin:0 0 3px 0;
}
.head-contact p strong{
    font-size:14px; font-weight:normal;
}
.head-contact p span{
    font-size:21px; font-weight: bold;
}

.search-head{
    float:right; list-style:none; background:url(images/bg-search.jpg) no-repeat left top;
    width:305px; height:29px; margin:5px 6px 5px 0;
    }
#quickquery{
    border:none; margin:6px 30px 3px 10px; width:210px; font-size:12px;
}
.search-head li{
    float:left;
}
.search-btn{
    margin:8px 0 0 0;
}

```

Рисунок 1.14 – Код файла **style.css** - блоки контактов и поиска (продолжение)

Далее идет описание для абзацев (параграфов) **<p>**, заключенных в контейнер **<div class="head-contact">**, а также для абзацев, оформленных тегами **** и ****. Обычные абзацы (у нас это строки «Будние дни: с 9:00 до 18:00» и «Суббота, Воскресенье - выходные») оформляются шрифтом Verdana, размером 12px, отступом 3 px снизу и межстрочным отступом с множителем 1,3.

Абзацы контейнера **<div class="head-contact">** с тегом «**strong**» (строки «Телефон» и «Режим работы:») оформлены текстом размером 14 px нормальной ширины, а с тегом «**span**» (строка с номером телефона – 8(800) 700-00-01) – размером 21px жирного (**bold**) шрифта.

Форма поиска в хедере оформлена в виде списка со стилем **<ul class="search-head">**. Класс для этого стиля приведен в последнем блоке файла стилей **style.css** (рисунок 1.14). Списки выровнены по правому краю хедера, подложкой (**background** - ом) для них является файл **bg-search.jpg**, задающий цвет поля ввода и кнопки «Найти». Размер блока: длина – 305px, высота – 29px, отступы – по 5px сверху и снизу и 6px справа. Данные размеры определялись путем подбора и сравнения результата с макетом (в сильно увеличенном формате).

Поле ввода (**селектор #quickquery**) имеет длину 210px, ширину 12px, без рамки и с отступами 6px сверху, 30px справа, 3px снизу и 10px слева.

Список **.search-head li** выравнивается по левому краю (нужно только помнить, что форма имеет размер блока 395px и выравнивание происходит именно внутри этого блока).

Кнопка «Найти» имеет свой класс **.search-btn**, в котором задан только верхний отступ 8px.

Вы можете «поиграть» с размерами и отступами, наблюдая за изменениями в браузере.

Посмотрите результат в браузере (рисунок 1.15).

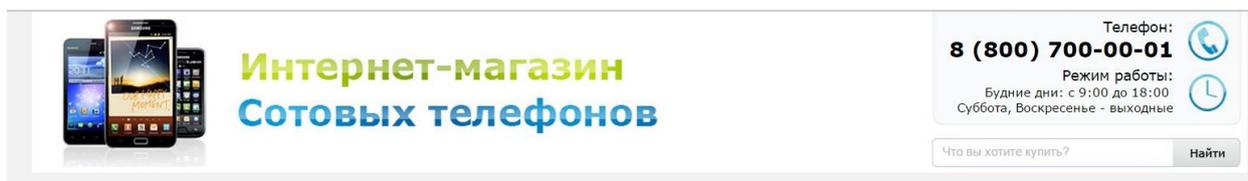


Рисунок 1.15 – Внешний вид шапки сайта

Таким образом, мы закончили написание кода шапки сайта.

Самостоятельно. Разобраться с приведенным кодом, ответить на любые вопросы преподавателя по оформлению блока хедера при защите лабораторной работы.

Лабораторная работа № 2

Создание главного (верхнего) меню, разметки для контента и блока айстопперов

Прошрое занятие мы закончили написанием кода шапки сайта. Переходим к созданию главного (верхнего) горизонтального меню.

Можно открыть скриншот любой страницы и увидеть, что главное меню располагается под шапкой сайта (рисунок 2.1).



Рисунок 2.1 – Шапка сайта и главное меню

Меню будет выполняться в виде горизонтального списка, при этом нет необходимости заключать меню в контейнер `<div>`, что достаточно удобно: контейнеры `<div>` довольно капризны и могут «рассыпаться» при

определенных условиях (несоблюдение форматов текста внутри контейнера, превышение количества пунктов меню и т.п.).

При этом верстка меню будет вестись «резиновая» (то есть размер пунктов меню будет подстраиваться под количество этих пунктов) и кроссбраузерная (правильно отображаться в большинстве современных и не очень современных браузеров).

Верстку меню сайта мы выполняем в главной странице нашего сайта – файле **index.html**. Откройте в блокноте **Notepad++** файл **index.html** и после тега `</div>` (рисунок 2.2), являющегося закрывающим парным тегом контейнера `<div class="header">` добавьте текст в красной рамке.

```
42
43     </form>
44 </div> <!-- header -->
45 <ul class="menu">
46     <li><a href="#">Главная</a></li>
47     <li><a href="#">О магазине</a></li>
48     <li><a href="#">Оплата и доставка</a></li>
49     <li><a href="#">Покупка в кредит</a></li>
50     <li><a href="#">Контакты</a></li>
51 </ul>
```

Рисунок 2.2 – Код главного меню в файле **index.html**

Посмотрите результат в браузере (рисунок 2.3).



Рисунок 2.3 – Внешний вид главного меню

Как видно из скриншота, мы получили классический список, причем вертикальный. Именно так любой браузер выведет список по умолчанию. Если посмотреть на код страницы, то можно увидеть, что мы создали стиль оформления списков, присвоив ему класс «**menu**» (`<ul class="menu">`). Однако этот класс никак пока не описан в таблице стилей, поэтому браузер выводит страницу по своим встроенным в него стилям оформления списков - «по умолчанию».

Всё оформление главного меню мы будем осуществлять в таблице стилей для нашего сайта – в файле **style.css**. Открываем файл стилей **style.css** в блокноте и создаем стиль для класса `<ul class="menu">`:

Для создания фона элементов меню рассмотрим подробно макет верстки главной страницы (рисунок 2.4).



Рисунок 2.4 – Элемент главного меню сайта

Лучшим вариантом будет подробно рассмотреть структуру ячейки в файле макета, выполненным в формате фотошопа (файл **главная.psd**). Файл **главная.psd**, открытый в Фотошопе, показан на рисунке 2.5.

Ячейка (состав ячейки, а именно – слоёв Фотошопа, выделен на рисунке красной рамкой) состоит из нескольких слоев: текст (белый шрифт), фон меню (двухцветный), подсветка (белый цвет, накладывается на фон и располагается по центру), а также полосы, отделяющие края ячейки (слева – более светлая, а справа – более темная).

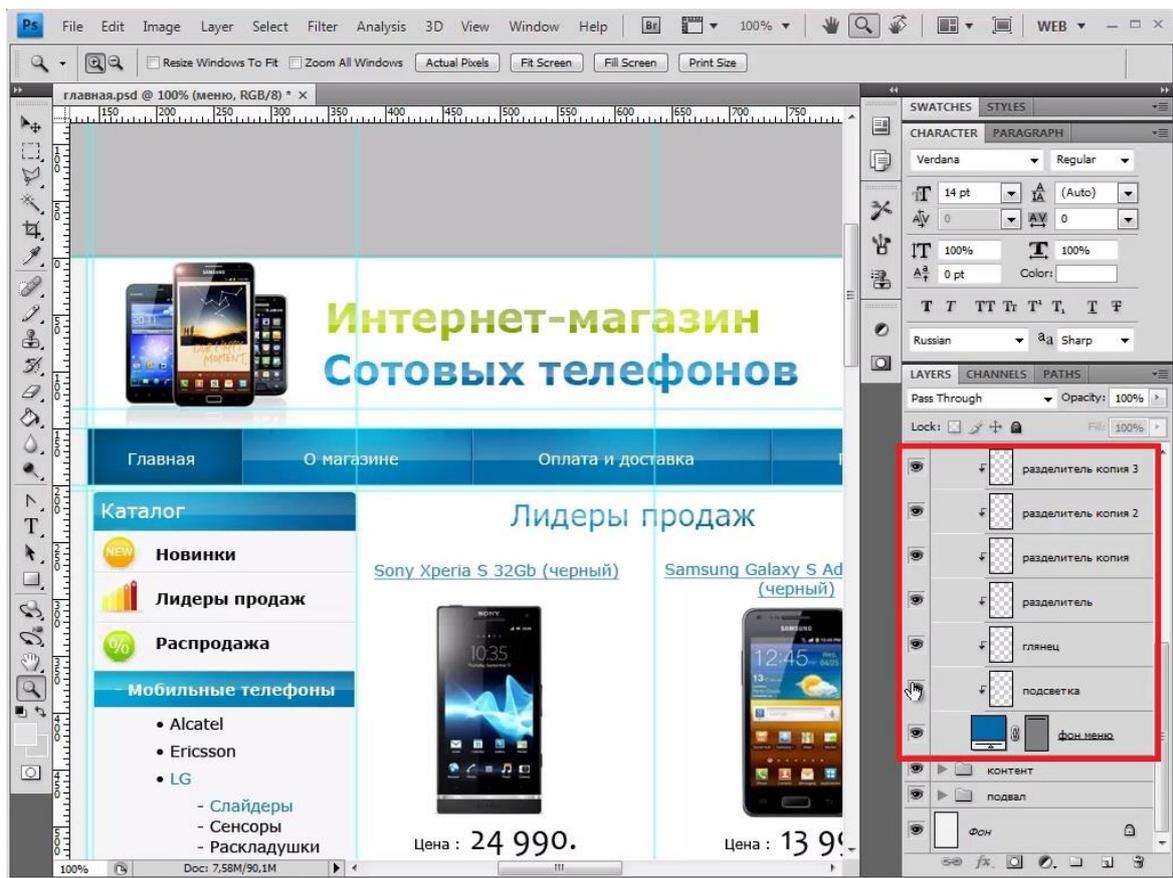


Рисунок 2.5 – Вид главной страницы в Adobe Photoshop

Фоном наших ячеек будет файл шириной 1px с именем **menu-bg.jpg** (он расположен в папке **images**), а файл подсветки ячейки – файл **menu-bg-light.png** (данный тип файла обладает замечательным свойством – прозрачностью, что позволяет «наложить» на другое изображение, не перекрывая его).

Опишем в файле **style.css** класс **.menu** (рисунок 2.6).

```
.menu {
    background:url(images/menu-bg.jpg) top repeat-x;
}
```

Рисунок 2.6 – Код класса **.menu** в файле стилей

Код файла стилей задаёт подложку нашего меню файлом **menu-bg.jpg** с повторением по оси x. Посмотрите результат в браузере (рисунок 2.7).

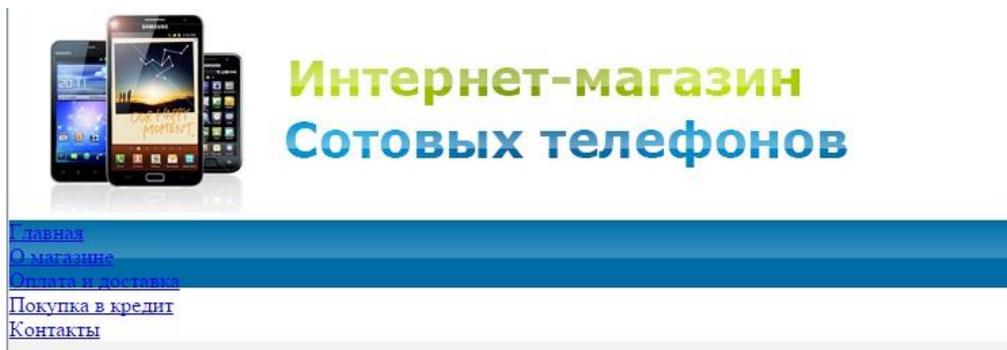


Рисунок 2.7 - Внешний вид меню после подключения файла подложки

Далее делаем ячейки блочными, задаем им ширину и высоту. В результате код стилей должен получиться в соответствии с рисунком 2.8.

В описании класса **.menu** использовано отображение списков в виде строк таблицы **display:table-row**. Отдельные элементы списка меню описываются свойством **.menu li**.

Данный стиль представляет внешний вид каждого элемента списка в виде ячейки таблицы: **display:table-cell**;, подсвечивает ячейку прозрачным файлом **menu-bg-light.png**, который имеет более светлый фон в середине и темнеет по краям, выровнен по центру и описывает левую границу ячейки шириной 1px светлым оттенком: **border-left:1px solid #2b99d0**, а правую границу – более тёмным оттенком: **border-right:1px solid #035786**. Текст внутри ячейки выравнивается по центру (**text-align:center**;) , размеры ячейки: ширина – автоматически (**width:auto**;) – это значит, что при изменении количества пунктов меню (добавлении нового пункта или удалении старого) размеры ячеек автоматически «подравняются». Если не будут вмещаться в одну строку- будут написаны в две.

```

.menu{
background:url(images/menu-bg.jpg) top repeat-x; display:table-row; list-style:none;
}

.menu li{
display:table-cell; background:url(images/menu-bg-light.png) center top no-repeat;
width:auto; text-align:center; height:50px; border-left:1px solid #2b99d0;
border-right:1px solid #035786; vertical-align:bottom;
}

.menu li:hover{
background:#01496f url(images/menu-bg-light.png) center top no-repeat;
border-left:1px solid #01496f; border-right:1px solid #01496f;
}

.menu li:first-child{
border-left:none;
}

.menu li:last-child{
border-right:none;
}

.menu li a{
display:block; width:1000px; height:50px; vertical-align:middle;
display:table-cell; font:14px Verdana, Geneva, sans-serif; color:#fff;
text-decoration:none;
}

```

Рисунок 2.8 – Код файла стилей **style.css** для главного меню

Края главного меню не имеют перехода цвета (оттенка) со светлого к более темному и наоборот. Для удаления переходов слева и справа использованы стилевые оформления **.menu li:first-child** и **.menu li:last-child**, удаляющие стили границ слева и справа.

При подведении курсора к пункту меню меняется стилевое оформление соответствующего пункта меню, за что отвечает класс **.menu li:hover**. Оформление стиля для данного класса меняет только цвет подложки (**background:#01496f**) и цвет правой и левой границ ячейки (**#01496f**).

Последний стиль для класса **.menu li** а определяет оформление гиперссылок в пунктах меню, которыми является текст, написанный в ячейках главного меню. Свойство **display:block**; говорит о том, что ссылки имеют «блочную» конструкцию. Ширина и высота блока, в котором располагается гиперссылка задается свойствами **width:1000px; height:50px;**, вертикальное выравнивание – свойством **vertical-align:middle**; - по центру. Свойство **display:table-cell**; заставляет текст меню располагаться в ранее заданных ячейках (**table-cell** можно перевести как «ячейка таблицы»), что сужает поле их расположения. Далее описано шрифтовое оформление текста в ячейках меню.

Результат посмотрите в браузере (рисунок 2.9).



Рисунок 2.9 – Результат оформления меню сайта

Переходим к созданию разметки для контента. Под главным меню располагается основное содержимое сайта, а ниже будет подвал (footer) сайта. Основное содержимое мы оформим в виде четырех контейнеров **<div>** в соответствии с рисунком 2.10.



Рисунок 2.10 – Схема центральной части сайта

Как видно из рисунка, контейнер, который охватывает весь контент (включая левое и правое меню) – это `<div id="contentwrapper">`. Он выделен красным цветом. Остальные контейнеры подписаны.

Впишите в код главной страницы сайта `index.html` строки в соответствии с рисунком 2.11. Добавляемый код заключен в красную рамку.

```

<ul class="menu">
  <li><a href="#">Главная</a></li>
  <li><a href="#">О магазине</a></li>
  <li><a href="#">Оплата и доставка</a></li>
  <li><a href="#">Покупка в кредит</a></li>
  <li><a href="#">Контакты</a></li>
</ul>
<div id="contentwrapper">
  <div id="content">
    контент
  </div> <!-- content -->
  <div id="left-bar">
    left-bar
  </div> <!-- left-bar -->
  <div id="right-bar">
    right-bar
  </div><!-- right-bar -->
</div> <!-- contentwrapper -->

```

Рисунок 2.11 – Код файла `index.html`

Таким образом мы подготовили 4 контейнера `<div>` для основного содержимого сайта. Единственное, что может Вас смутить – то, что `<div id="content">` (расположенный в центре на рисунке 2.10) – написан перед `<div id="left-bar">` (расположенного слева). Но это нормально, мы «раскидаем» контейнеры в разные стороны и последовательно опишем всё в свойствах стилей `css`.

Вначале, для того чтобы разобраться с версткой контента, просто дадим произвольные разные цвета нашим контейнерам. В файле стилей `style.css` опишите свойства контейнеров (рисунок 2.12). Обратите внимание, что в связи с тем, что вместо классов мы использовали для описания в файле `index.html` идентификаторы `id=...` для контейнеров `<div>`. В этом случае селекторы в файле `style.css` описываются с помощью символов решетки `#`.

```
#contentwrapper{
    float:left; width:100%; color:#fff;
}

#content{
    background:#066;
}

#left-bar{
    background:#03f;
}

#right-bar{
    background:#900;
}
```

Рисунок 2.12 – Код файла `style.css`

Мы задали выравнивание по левому краю, ширину 100% блока, в котором содержится элемент, и белый цвет шрифта для всего селектора `#contentwrapper` (который соответствует контейнеру `<div id="contentwrapper">`), а затем раскрасили в разные цвета фоны наших контейнеров.

Сохраните файл, посмотрите результат в браузере (рисунок 2.13).

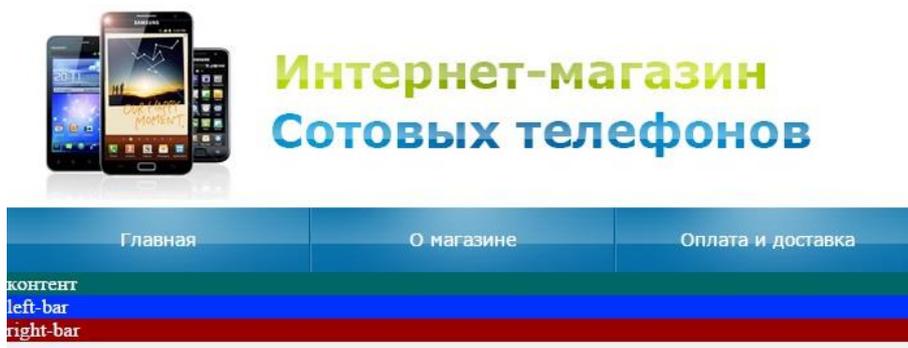


Рисунок 2.13 – Внешний вид главной страницы сайта с добавленными контейнерами <div>

Как видно из рисунка, контейнеры выстроились в порядке их описания в тексте программы один под другим и заняли всю длину страницы. Вначале зададим выравнивание и ширину для контейнера **left-bar** (рисунок 2.14).

```
#left-bar{
    background:#03f;
    float:left;
    width:240px;
}
```

Рисунок 2.14 - Код файла **style.css** (продолжение)

Посмотрим результат в браузере (рисунок 2.15).



Рисунок 2.15 – Внешний вид главной страницы сайта после описания свойств селектора **#left-bar**

Как видно из рисунка, контейнер **left-bar** ушел влево, а справа к нему «пристроился» контейнер **right-bar** и занял всё пространство строки до конца.

Мы хотим, чтобы контейнер **контент** «провалился» между контейнерами **left-bar** и **right-bar**. Для этого вначале дадим ему отступ слева (рисунок 2.16).

```
#content{  
    background:#066;  
    margin:0 0 0 240px;  
}
```



Рисунок 2.16 - Код файла **style.css** (продолжение)

Посмотрите результат в браузере (рисунок 2.17).



Рисунок 2.17 – Внешний вид главной страницы сайта после описания свойств селектора **#content**

Да, контейнер **Контент** отодвинулся влево, но пока еще не «провалился» между контейнерами **left-bar** и **right-bar**.

Для этого мы заставим левый блок, наоборот, подняться наверх. Мы возьмем контейнер `<div id="contentwrapper">` и «обернем» им только контейнер `<div id="content">`.

Для этого мы в файле **index.html** перенесем закрывающий тег `</div>` контейнера `<div id="contentwrapper">` наверх (рисунок 2.18)

```

<div id="contentwrapper">
  <div id="content">
    КОНТЕНТ
  </div> <!-- content -->
  <div id="left-bar">
    left-bar
  </div> <!-- left-bar -->
  <div id="right-bar">
    right-bar
  </div><!-- right-bar -->
</div> <!-- contentwrapper -->

```

Рисунок 2.18 - Код файла **index.html**

В результате код файла **index.html** должен выглядеть в соответствии с рисунком 2.19.

```

<div id="contentwrapper">
  <div id="content">
    КОНТЕНТ
  </div> <!-- content -->
</div> <!-- contentwrapper -->
  <div id="left-bar">
    left-bar
  </div> <!-- left-bar -->
  <div id="right-bar">
    right-bar
  </div><!-- right-bar -->

```

Рисунок 2.19 - Код файла **index.html** (исправленный)

Пара дивов (открывающий и закрывающий) контейнера **<div id="contentwrapper">** подсвечена в блокноте.

Если посмотреть на результат, то пока еще не все понятно (рисунок 2.20).



Рисунок 2.20 - Внешний вид главной страницы

Правый контейнер (красного цвета) стремится занять всё возможное пространство. Это свойство всех контейнеров `<div>`. Зададим размеры и отступы наших контейнеров. Вначале подвинем левый контейнер (код файла `style.css` на рисунке 2.21).

```
#left-bar{
  background:#03f;
  margin-left:-100%;
  float:left;
  width:240px;
}
```

Рисунок 2.21 - Код файла `style.css` (продолжение)

Результат изменений в браузере показан на рисунке 2.22.



Рисунок 2.22 – Вид главной страницы после изменения свойств контейнера `#left-bar`

Контейнер **#left-bar** поднялся вверх и занял положенное место. Допишите в файле стилей **style.css** следующий код (он выделен красным цветом на рисунке 2.23).

```
#contentwrapper{
    float:left; width:100%; color:#fff;
}
#content{
    background:#066;
    margin:0 245px 0 240px;
}
#left-bar{
    background:#03f;
    margin-left:-100%;
    float:left;
    width:240px;
}
#right-bar{
    background:#900;
    float:left;
    width:245px;
    margin-left:-245px;
}
```

Рисунок 2.23 - Код файла **style.css** (продолжение)

Мы задали ширину правого контейнера равной 245px, выровняли его по левому краю, дали отступы слева контейнеру **content** на 245px, а правому контейнеру на **минус** 245px.

Посмотрите результат в браузере (рисунок 2.24). Все контейнеры встали на свои положенные места. При этом «резиновость» верстки сохраняется. Можно изменять размеры окна браузера, при этом ширина контейнеров **left-bar** и **right-bar** будет оставаться неизменной, меняться будет только размер центрального контейнера **content**.



Рисунок 2.24 – Внешний вид главной страницы сайта

Таким образом, мы закончили разбираться с контейнерами <div> в оформлении контента сайта. Как Вы уже поняли, эти контейнеры очень «капризные» и сложные в разметке сайтов. Результат – именно тот, что и в макете нашей вёрстки.

Продолжаем программирование контента страницы и начнем с левого меню, а точнее, с основной его частью – с блоком навигации (он выделен на рисунке 2.25 красной рамкой). Почему мы назвали его блоком навигации – потому, что, выбирая любой из пунктов данного меню, пользователь получит определенный результат, различающийся в зависимости от того, какой именно пункт меню был выбран.

Первым делом мы создадим так называемый блок «Айстопперов» (от англ. Eyestopper – то, на чем глаза останавливаются). Обычно один или несколько айстопперов располагаются на главной странице сайта – считается, что покупатель может не тратить время на поиск товара, а сразу перейти к лидерам продаж, новинкам или распродажам. На рисунке 2.25 представлен макет главной страницы сайта с выделенным блоком навигации.



Рисунок 2.25 – Вид макета главной страницы сайта с выделенным блоком: навигации

Блок айстопперов состоит из заголовка «Каталог» и трех пунктов – «Новинки», «Лидеры продаж» и «Распродажа» (рисунок 2.26).



Рисунок 2.26 – Блок айстопперов (в макете главной страницы)

Мы оформим все содержимое блока в виде заголовков – «Каталог» оформим в тегах <h2>, а остальные пункты меню – в тегах <h3>.

Добавьте в код файла **index.html** вместо текста «**left-bar**» код, выделенный красной рамкой на рисунке 2.27.

```
<div id="left-bar">
  <div class="left-bar-cont">
    <h2><span>Каталог</span></h2>
    <h3 class="nav-new"><a href="#">Новинки</a></h3>
    <h3 class="nav-lider"><a href="#">Лидеры продаж</a></h3>
    <h3 class="nav-sale"><a href="#">Распродажа</a></h3>
  </div>
</div>
<div id="right-bar">
  right-bar
</div>
```

Рисунок 2.27 – Код файла **index.html**

Как видно из рисунка, блок айстопперов мы заключили в отдельный контейнер с названием **<div class="left-bar-cont">** (что означает сокращение от «контент (содержимое) левого меню»), а пункты - в виде заголовков (при этом заголовок «Каталог» заключен в строчный контейнер «**span**»). Применение контейнера «**span**» обусловлено тем, что возникает сложность при создании подложки (бэкграунда) названия «Каталог». Как видно из рисунка 2.26, подложка состоит из двух слоев: первый слой – это градиентное заполнение по горизонтали от светло-синего цвета к темно-синему; второй слой – это так называемый прозрачный «глянец», который был применен при оформлении верхнего горизонтального меню. Поэтому градиент мы применим к элементу **h2** класса **left-bar-cont**, а слой «глянец» - к контейнеру «**span**». В итоге глянец «наложится» на градиент и мы получим требуемый вид пункта «Каталог».

Посмотрите результат в браузере (рисунок 2.28).

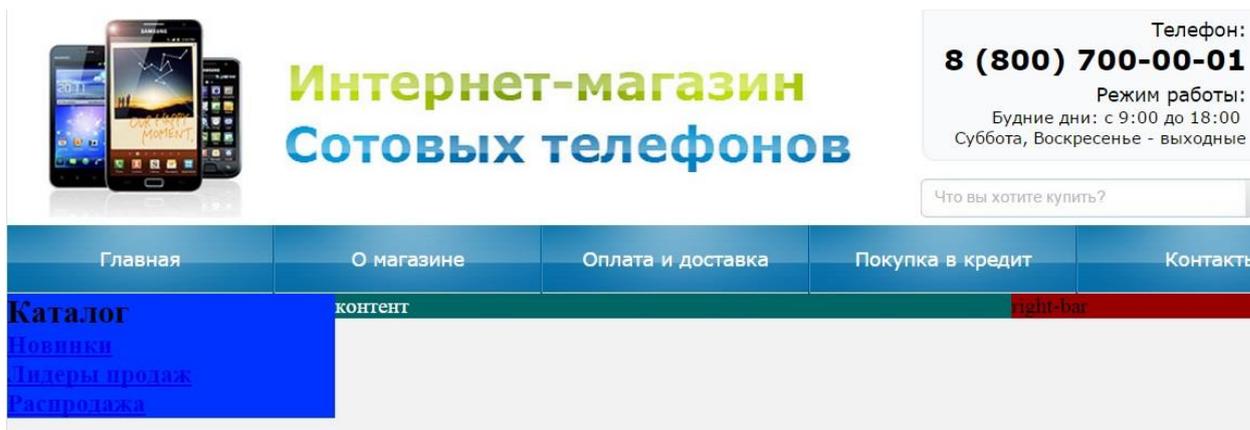


Рисунок 2.28 – Вид главной страницы с добавленными пунктами айстопперов

Пункты создаваемого меню созданы, но пока не оформлены. Оформлять, как всегда мы будем в файле стилей – **style.css**. Откройте данный файл.

Во-первых, уберите синий цвет (удалите строку) – удаленная строка показана на рисунке 2.29.

```
#left-bar{
  background:#03f;
  margin-left:-100%;
  float:left;
  width:240px;
}
```

Рисунок 2.29 – Удаленное цветовое оформление контейнера **#left-bar** (файл **style.css**)

Во-вторых, создайте новый класс **.left-bar-cont** и дайте свойства элементу **h2** этого класса (рисунок 2.30).

```

#left-bar{
    margin-left:-100%;
    float:left;
    width:240px;
}

.left-bar-cont{
    background:#f7f7f7; margin:6px 0 0 6px;
}
.left-bar-cont h2{
    color:#fff; font:normal 18px Verdana, Geneva, sans-serif;
}

#right-bar{

```

Рисунок 2.30 – Код файла **style.css**

Посмотрите результат в браузере (рисунок 2.31).

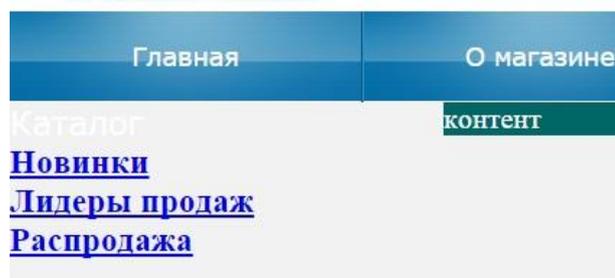


Рисунок 2.31 – Вид блока айстопперов

Как видно из рисунка 2.31, слово «Каталог» (белого цвета текста – это определяется тегом **color:#fff** элемента **h2** класса **.left-bar-cont h2**) «потерялось» на светло-сером фоне (**background:#f7f7f7;**) контейнера **div** с классом **left-bar-cont**. Чтобы убедиться в существовании текста «Каталог» можно либо приблизить элемент (и очень внимательно посмотреть на участок с текстом), либо выделить его мышью (как это показано на рисунке ниже 2.32).

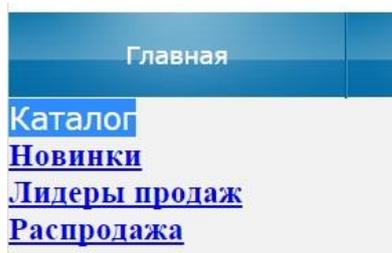


Рисунок 2.32 – Вид блока айстопперов с выделенным заголовком

Итак, заголовок «Каталог» существует, осталось заполнить его подложку градиентным переходом.

Градиент — вектор, своим направлением указывающий направление наибольшего возрастания некоторой величины. значение которой меняется от одной точки пространства к другой (скалярного поля), а по величине (модулю) равный скорости роста этой величины в этом направлении. Градиенты применяются в различных сферах, но нас интересует сфера веб-разработки, где градиенты часто применяются в качестве основного фона сайтов и различных контейнеров, линий, цитат, блоков и даже текста.

Для этого мы воспользуемся помощью Интернета. В Сети существует масса сайтов, посвященных web-разработкам, предлагающих готовые решения по оформлению сайтов — так называемые генераторы (меню, цветовых переходов и т.д.). Большинство таких сайтов предлагает воспользоваться их услугами совершенно бесплатно. Мы воспользуемся генератором градиентов **Ultimate CSS Gradient Generator** от фирмы **ColorZilla** (сайт <https://www.colorzilla.com/gradient-editor/>). Пройдите по ссылке, в браузере откроется страница генератора градиентов (рисунок 2.33), использующий механизм стилей (css).

Области, выделенные зелеными рамками, позволяют настраивать внешний вид градиента, а в области, обведенной красной рамкой, содержится код стилей для данного оформления, причем стили прописаны для различных браузеров, включая старые.

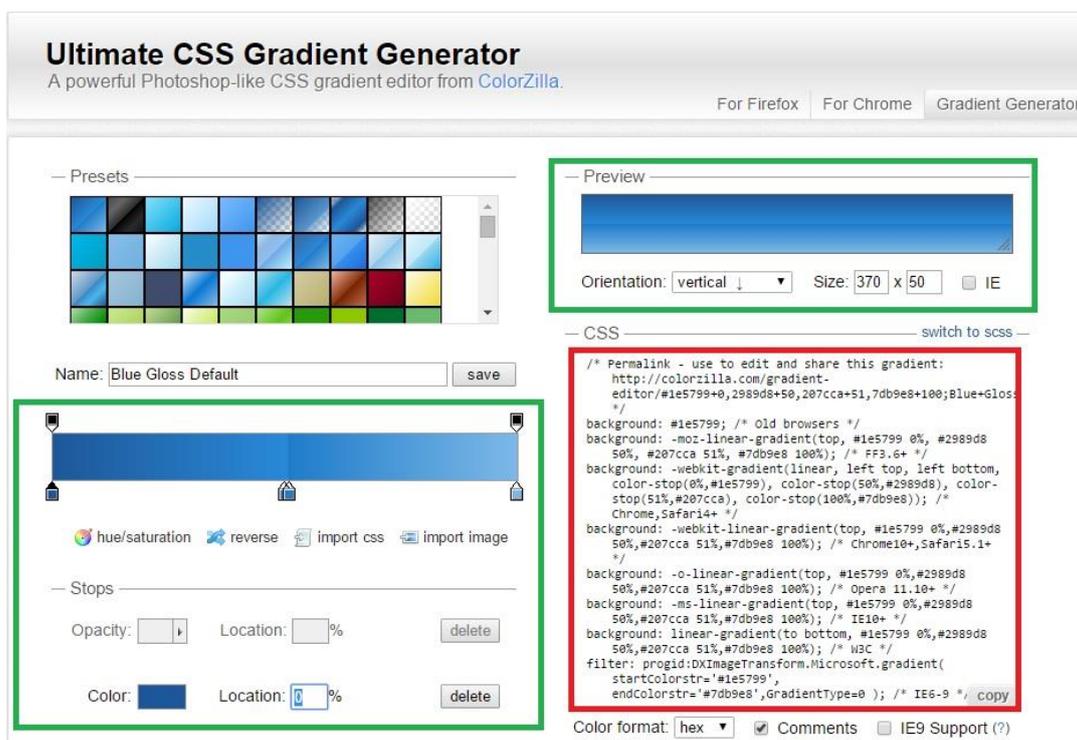


Рисунок 2.33 - Сайт градиентного оформления стилей элементов

В настоящей лабораторной работе Вам предлагается воспользоваться готовым кодом градиентной заливки. В папке «Для лабы 2» Вам предложен файл **градиент.txt**. Скопируйте приведенный в файле код и вставьте его в стиль оформления элемента **.left-bar-cont h2** (рисунок 2.34).

```
.left-bar-cont h2{
  color:#fff; font:normal 18px Verdana, Geneva, sans-serif;
  background: rgb(14,148,195); /* Old browsers */
  background: -moz-linear-gradient(left, rgba(14,148,195,1) 0%, rgba(4,88,144,1) 100%); /* FF3.6+ */
  background: -webkit-gradient(linear, left top, right top, color-stop(0%,rgba(14,148,195,1)), color-stop(100%,rgba(4,88,144,1))); /* Chrome10+,Safari5.1+ */
  background: -webkit-linear-gradient(left, rgba(14,148,195,1) 0%,rgba(4,88,144,1) 100%); /* Chrome10+,Safari5.1+ */
  background: -o-linear-gradient(left, rgba(14,148,195,1) 0%,rgba(4,88,144,1) 100%); /* Opera 11.10+ */
  background: -ms-linear-gradient(left, rgba(14,148,195,1) 0%,rgba(4,88,144,1) 100%); /* IE10+ */
  background: linear-gradient(to right, rgba(14,148,195,1) 0%,rgba(4,88,144,1) 100%); /* W3C */
  filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#0e94c3', endColorstr='#045890',GradientType=1 ); /*
border-radius:4px 4px 0 0;
}
```

Рисунок 2.34 - Код файла **style.css** (продолжение)

Сохраните файл стилей, обновите сайт в браузере (рисунок 2.35).



Рисунок 2.35 – Результат градиентного оформления пункта «Каталог»

Как видно из рисунка, градиент к фоновому изображению применился. Теперь необходимо добавить «глянец». Напоминаю, что «глянец» мы применяем к элементу **span**.

Добавьте в код файла стилей строки, заключенные в зеленую рамку на рисунке 2.36.

```
.left-bar-cont h2{
  color:#fff; font:normal 18px Verdana, Geneva, sans-serif;
  background: rgb(14,148,195); /* Old browsers */
  background: -moz-linear-gradient(left, rgba(14,148,195,1) 0%, rgba(4,88,144,1) 100%); /
  background: -webkit-gradient(linear, left top, right top, color-stop(0%,rgba(14,148,195,1) 0%, rgba(4,88,144,1) 100%);
  background: -webkit-linear-gradient(left, rgba(14,148,195,1) 0%,rgba(4,88,144,1) 100%);
  background: -o-linear-gradient(left, rgba(14,148,195,1) 0%,rgba(4,88,144,1) 100%); /* C
  background: -ms-linear-gradient(left, rgba(14,148,195,1) 0%,rgba(4,88,144,1) 100%); /*
  background: linear-gradient(to right, rgba(14,148,195,1) 0%,rgba(4,88,144,1) 100%); /*
  filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#0e94c3', endColorstr='#000000', gradientType=1);
  border-radius:4px 4px 0 0;
}

.left-bar-cont h2 span{ display:block; background:url(images/glyanec.png) top repeat-x;
  padding:5px 0 5px 10px;
}
```

Рисунок 2.36 - Код файла **style.css** (продолжение)

Сохраните изменения, посмотрите результат в браузере – на пункт «Каталог» должен «наложиться глянец». Таким образом, мы получили требуемый результат оформления пункта «Каталог» нашего блока айстопперов.

Переходим к оформлению пунктов данного меню. Добавьте в файл стилей код, заключенный в красную рамку на рисунке 2.37.

```
.left-bar-cont h2 span{ display:block; background:url(images/glyanec.png) top repeat-x;
padding:5px 0 5px 10px;
}
.nav-new, .nav-lider, .nav-sale{ padding:10px 0 10px 55px; }
.nav-new a, .nav-lider a, .nav-sale a{
color:#000; font:bold 14px Verdana, Geneva, sans-serif; text-decoration:none;
}
.nav-new a:hover, .nav-lider a:hover, .nav-sale a:hover{
text-decoration: underline;
}
.nav-new{ background:url(images/new-nav.jpg) left center no-repeat;
border-bottom:1px solid #e1e1e1;
}
.nav-lider{ background:url(images/lider-nav.jpg) left center no-repeat;
border-bottom:1px solid #e1e1e1; border-top:1px solid #fff;
}
.nav-sale{ background:url(images/sale-nav.jpg) left center no-repeat;
border-bottom:1px solid #a1a1a1; border-top:1px solid #fff;
}
#right-bar{
background:#900:
```

Рисунок 2.37 - Код файла style.css (продолжение)

Сохраните изменения, посмотрите результат в браузере (рисунок 2.38).

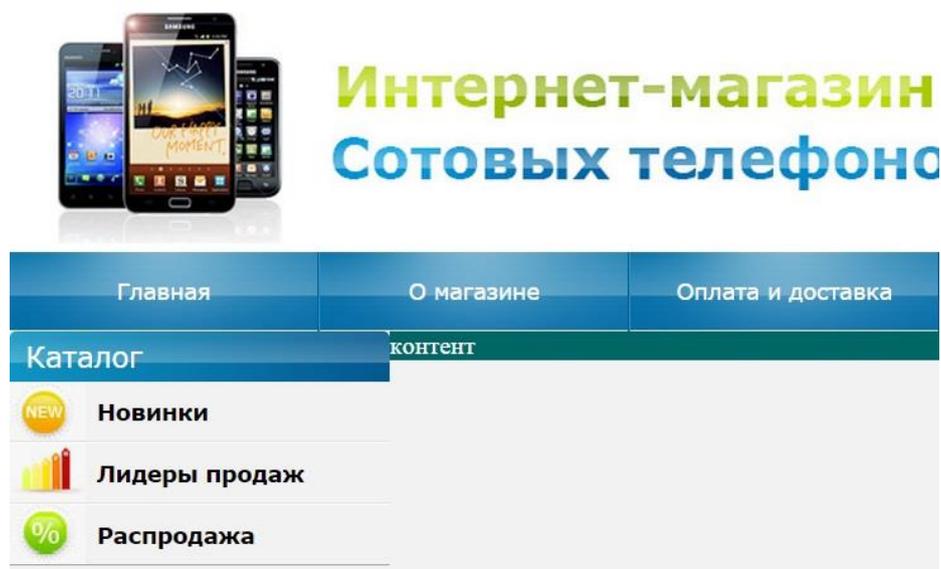


Рисунок 2.38 - Результат оформления блока айстопперов

Как видно из рисунка, мы получили именно то оформление айстопперов, которое было разработано web-дизайнером.

Самостоятельно.

1) Добавьте пару новых пунктов меню, посмотрите результат – пункты меню должны автоматически выравниваться, вписавшись в размеры страницы сайта. Убедившись в правильности вёрстки, удалите добавленные Вами пункты меню, предварительно показав их преподавателю.

2) Разобраться с приведенным кодом, ответить на любые вопросы преподавателя по оформлению блока хедера, разметке блока контента и оформлению блока айстопперов при защите лабораторной работы.

Лабораторная работа № 3

Создание блоков навигации, контактов, новостей и информеров

Прошрое занятие мы закончили написанием кода разметки для блока навигации, а точнее – блока айстопперов нашей страницы (раздел «Каталог»). Продолжаем оформление левого меню, а точнее, центральной его части – блока навигации (он выделен на рисунке 3.1 зеленой рамкой).

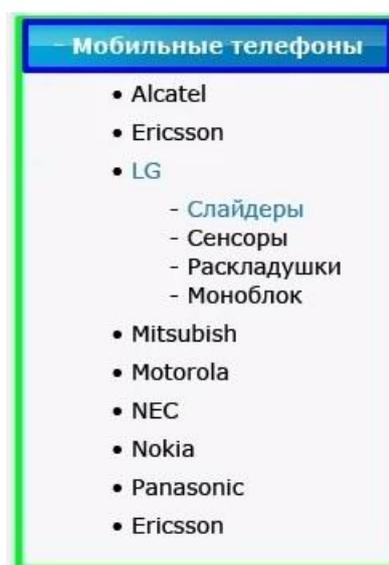


Рисунок 3.1 – Блок навигации

Блок навигации состоит из заголовка «Мобильные телефоны», девяти пунктов – названий фирм-производителей телефонов и четырех подпунктов. Подразумевается, что при программировании любой пункт будет «раскрываться» на произвольное количество подпунктов. Этот функционал будет реализован в следующем блоке лабораторных работ – «Программирование видимой части интернет магазина».

Мы не будем создавать отдельный контейнер `<div>` для блока навигации, а вставим требуемые пункты меню в созданный ранее контейнер `<div class="left-bar-cont">`.

Заголовок блока навигации «Мобильные телефоны» оформим в тегах **<h4>** (теги **<h2>** и **<h3>** использованы в этом контейнере ранее при оформлении блока айстопперов), а остальные пункты меню оформим в виде списков.

Добавьте в код файла **index.html** текст, выделенный красной рамкой на рисунке 3.2.

```
<div class="left-bar-cont">
  <h2><span>Каталог</span></h2>
  <h3 class="nav-new"><a href="#">Новинки</a></h3>
  <h3 class="nav-lider"><a href="#">Лидеры продаж</a></h3>
  <h3 class="nav-sale"><a href="#">Распродажа</a></h3>
  <h4><span>- Мобильные телефоны</span></h4>
  <ul class="nav-catalog">
    <li><a href="#">Alcatel</a></li>
    <li><a href="#">Ericsson</a></li>
    <li><a href="#">LG</a>
      <ul>
        <li><a href="#">Слайдеры</a></li>
        <li><a href="#">Сенсоры</a></li>
        <li><a href="#">Раскладушки</a></li>
        <li><a href="#">Моноблок</a></li>
      </ul>
    </li>
    <li><a href="#">Mitsubishi</a></li>
    <li><a href="#">Motorola</a></li>
    <li><a href="#">NEC</a></li>
    <li><a href="#">Nokia</a></li>
    <li><a href="#">Panasonic</a></li>
    <li><a href="#">Ericsson</a></li>
  </ul>
</div>
```

Рисунок 3.2 – Код файла **index.html**

Как видно из рисунка, заголовок «Мобильные телефоны» заключен в строчный контейнер «**span**». Применение контейнера «**span**» в заголовке **<h4>** аналогично тому, как этот строчный контейнер был использован в заголовке «Каталог» (см. лабораторную работу 4) - градиент мы применим к элементу **h4** класса **left-bar-cont**, а слой «глянец» - к контейнеру «**span**». В итоге глянец «наложится» на градиент и мы получим требуемый вид пункта «Мобильные телефоны».

Посмотрите результат в браузере (рисунок 3.3).

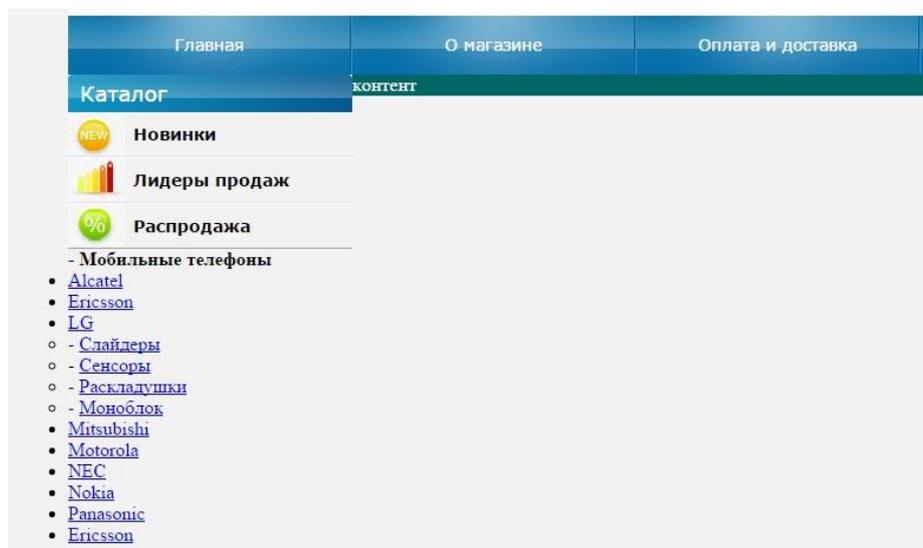


Рисунок 3.3 – Внешний вид блока навигации без стилового оформления

Пункты создаваемого меню навигации созданы, но пока не оформлены. Оформлять, как всегда мы будем в файле стилей – **style.css**. Откройте данный файл.

Начнем с заголовка «Мобильные телефоны». Стилизовое оформление возьмем из оформления заголовка **<h2>** (изменим только отступы, размеры шрифта, а также уберем закругления – `border-radius`). *Полностью скопируйте* стили оформления для элементов **.left-bar-cont h2** и **.left-bar-cont h2 span** и вставьте код ниже (после стиля элемента **.nav-sale**). *Исправьте* размер заголовка **h2** на **h4**. И уберите закругления заголовка (удалите строку), так как их (закруглений) нет в макете страницы. Код файла **style.css** с добавленным классом **.left-bar-cont h4** представлен на рисунке 3.4.

```

.nav-sale{ background:url(images/sale-nav.jpg) left center no-repeat;
           border-bottom:1px solid #a1a1a1; border-top:1px solid #fff;
           }
.left-bar-cont h4{
color:#fff; font:normal 18px Verdana, Geneva, sans-serif;
background: rgb(14,148,195); /* Old browsers */
background: -moz-linear-gradient(left, rgba(14,148,195,1) 0%, rgba(4,88,144,1) 100%); /* FF3.6+ */
background: -webkit-gradient(linear, left top, right top, color-stop(0%,rgba(14,148,195,1)), color-stop(100%,rgba(4,88,144,1)));
background: -webkit-linear-gradient(left, rgba(14,148,195,1) 0%,rgba(4,88,144,1) 100%); /* Chrome10+,Safari5.1+ */
background: -o-linear-gradient(left, rgba(14,148,195,1) 0%,rgba(4,88,144,1) 100%); /* Opera 11.10+ */
background: -ms-linear-gradient(left, rgba(14,148,195,1) 0%,rgba(4,88,144,1) 100%); /* IE10+ */
background: linear-gradient(to right, rgba(14,148,195,1) 0%,rgba(4,88,144,1) 100%); /* W3C */
filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#0e94c3', endColorstr='#045890',GradientType=1 ); /* IE6-9
border-radius:4px 4px 0 0;
           }
.left-bar-cont h4 span{ display:block; background:url(images/glyanec.png) top repeat-x;
                       padding:5px 0 5px 10px;
           }
#right-bar{
background:#900;
float:left;
width:245px;
margin-left:-245px;
}

```

Рисунок 3.4 – Код файла **style.css**

Посмотрите результат в браузере (рисунок 3.5).

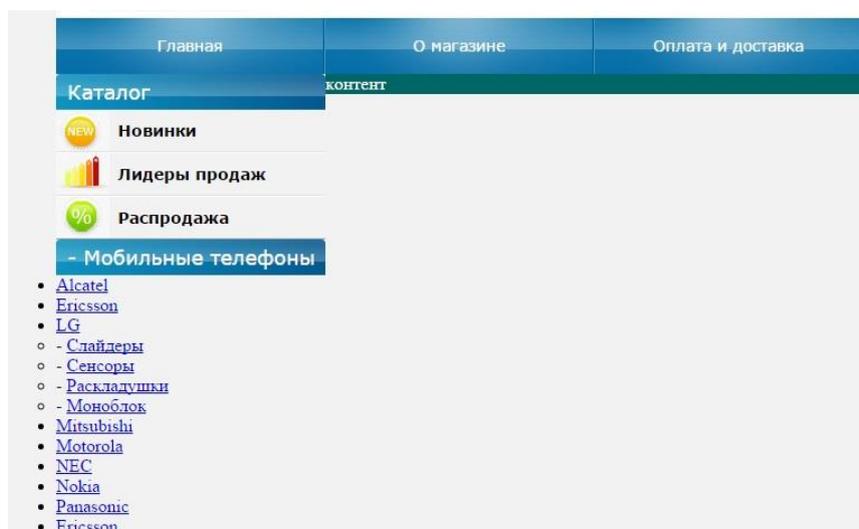


Рисунок 3.5 – Внешний вид блока навигации

Как видно из рисунка, заголовки «Каталог» и «Мобильные телефоны» практически идентичны по оформлению. Однако, если открыть файл макета нашей страницы (файлы **главная.psd** или **главная.jpg** в папке «Макет»), то можно увидеть, что заголовок «Мобильные телефоны» имеет немного меньшие размеры, чем заголовок «Каталог».

Исправим шрифты и отступы для заголовка «Мобильные телефоны». Изменения выделены красной рамкой на рисунке 3.6.

```
.left-bar-cont h4{
color:#fff; font: bold 14px Verdana, Geneva, sans-serif;
background: rgb(14,148,195); /* Old browsers */
background: -moz-linear-gradient(left, rgba(14,148,195,1) 0%, rgba(4,88,14,1) 100%);
background: -webkit-gradient(linear, left top, right top, color-stop(0%,rgba(14,148,195,1) 0%, rgba(4,88,14,1) 100%));
background: -o-linear-gradient(left, rgba(14,148,195,1) 0%, rgba(4,88,14,1) 100%);
background: -ms-linear-gradient(left, rgba(14,148,195,1) 0%, rgba(4,88,14,1) 100%);
background: linear-gradient(to right, rgba(14,148,195,1) 0%, rgba(4,88,14,1) 100%);
filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#0e94c3', endColorstr='#000000', GradientType=1);
}
.left-bar-cont h4 span{ display:block; background:url(images/glyanec.png) no-repeat;
padding:5px 0 5px 15px;
}
```

Рисунок 3.6 - Код файла **style.css** (продолжение)

Посмотрите результат в браузере (рисунок 3.7).

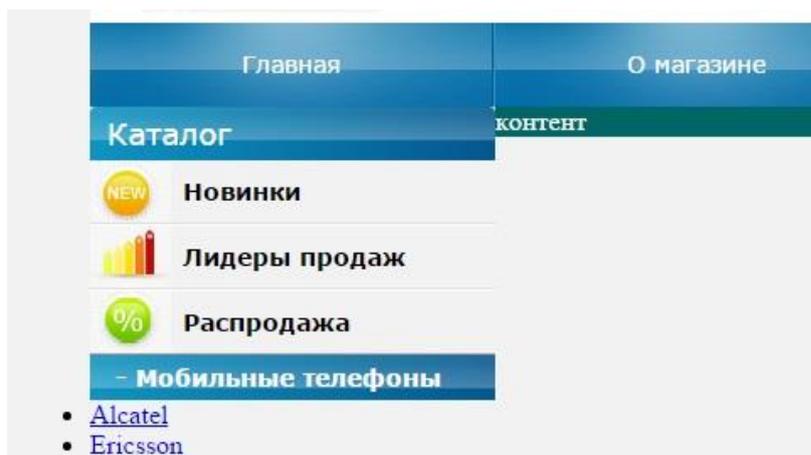


Рисунок 3.7 – Внешний вид пункта «Мобильные телефоны»

Как видно из рисунка, заголовок «Мобильные телефоны» практически достиг требуемого в макете оформления. За одним исключением – слой «глянец» съехал вниз. Это произошло потому, что размеры шрифта заголовка изменились с 18пт до 14 пт, и он, соответственно, уменьшился в высоте (так и должно быть, согласно макета страницы), а файл с глянцевым слоем **glyanec.png** остался прежнего размера. У Вас в папке **images** уже есть уменьшенная копия этого файла с именем **glyanec2.png**. Поэтому осталось только его прописать в файле стилей (рисунок 3.8).

```
.left-bar-cont h4 span{ display:block; background:url(images/glyanec2.png) top :  
padding:5px 0 5px 15px;  
}
```

Рисунок 3.8 – Изменения кода файла **style.css**

Посмотрите результат в браузере (рисунок 3.9).

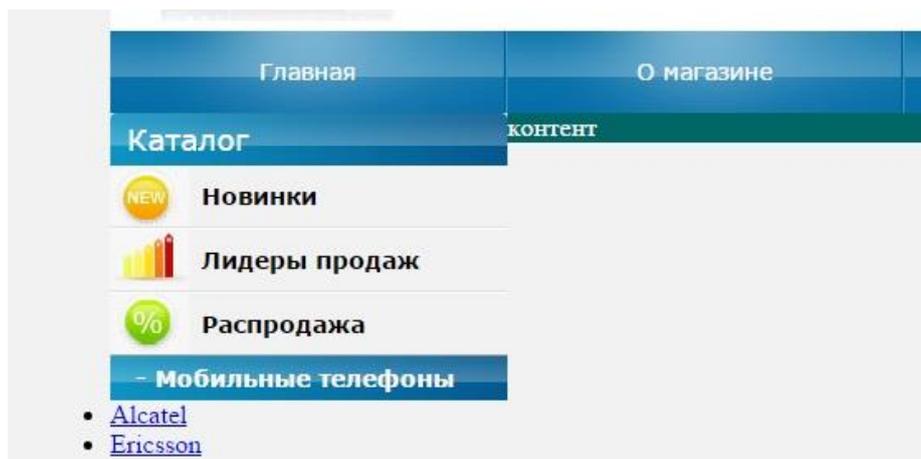


Рисунок 3.9 – Исправленный внешний вид пункта «Мобильные телефоны»

Сравните полученный результат с файлом макета и убедитесь в полном совпадении оформления.

Переходим к оформлению списков. Вначале небольшое замечание. Списки – очень «капризная вещь». Отображение маркеров у маркированных списков сильно зависит от браузера, в котором открывается файл со списком.

На рисунках 3.10 и 3.11 показаны вид списков в различных браузерах.

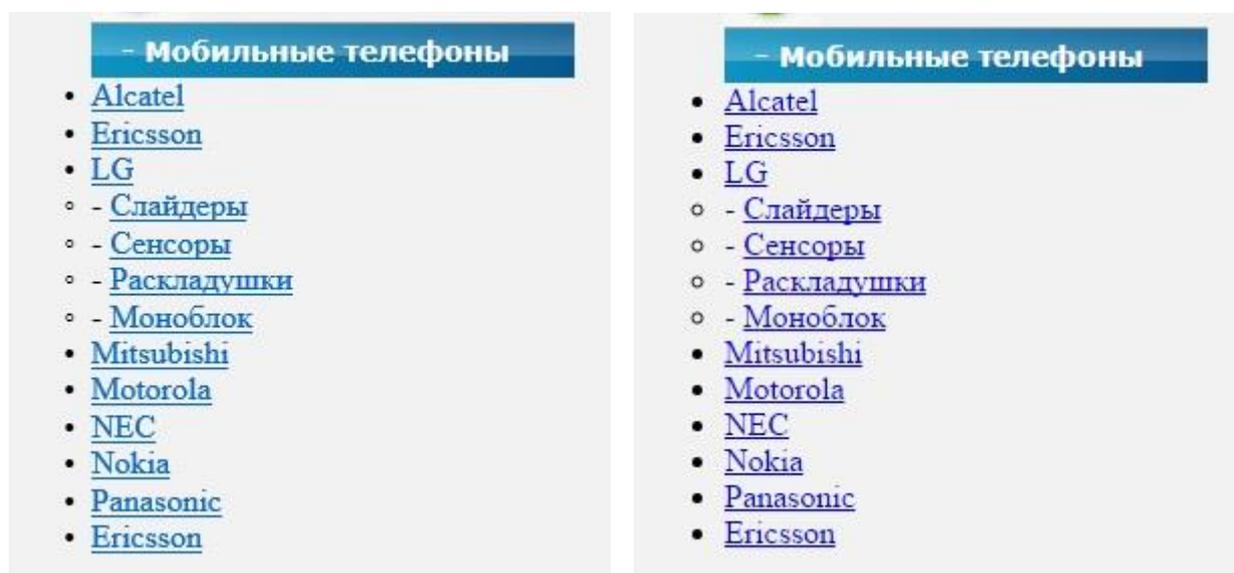


Рисунок 3.10 – Внешний вид списков в браузерах Chrome (слева) и IE (справа) – файл **index.html**

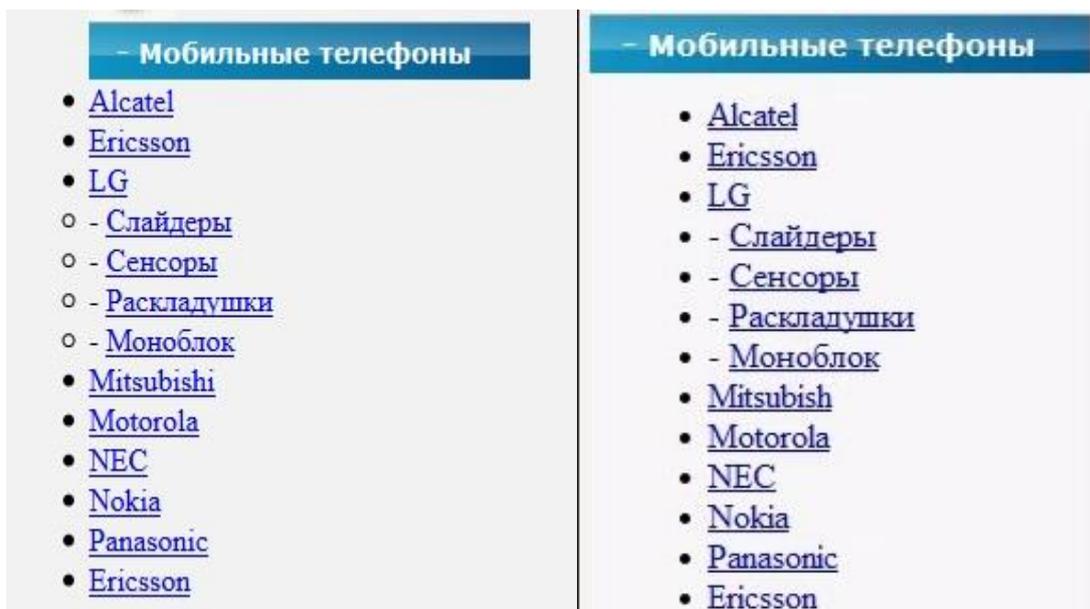


Рисунок 3.11 – Внешний вид списков в браузерах Firefox (слева) и Opera (справа) – файл **index.html**

Как видно из рисунков, в разных браузерах маркеры списков отображаются по-разному (по величине и по оформлению). Чтобы наши списки одинаково отображались во всех браузерах, в качестве маркеров списков используем картинку **disc.gif**. В этом случае оформление списков будет одинаково отображаться во всех браузерах.

Откройте файл стилей **style.css** и добавьте стилевое оформление для списков (рисунок 3.12).

```

.left-bar-cont h4 span{ display:block; background:url(images/glyanec2.png)
top repeat-x; padding:5px 0 5px 15px;
}
.nav-catalog{
padding:10px 10px 20px 55px; list-style:url(images/disc.gif);
border-bottom:1px solid #e1e1e1; border-top:1px solid #fff;
}
.nav-catalog ul{
padding:5px 0 0 20px; list-style:none;
}
.nav-catalog li{
margin:0 0 5px 0;
}
.nav-catalog li a{
font:bold 14px Verdana, Geneva, sans-serif; color:#000;
text-decoration:none;
}
.nav-catalog li a:hover{
font:bold 14px Verdana, Geneva, sans-serif; color:#1b7cb1;
text-decoration:none;
}
.nav-catalog ul li{
margin:0;
}
#right-bar{

```

Рисунок 3. 12 – Оформление элементов блока навигации в файле **style.css**

Сохраните файл стилей, обновите сайт в браузере (рисунок 3.13).

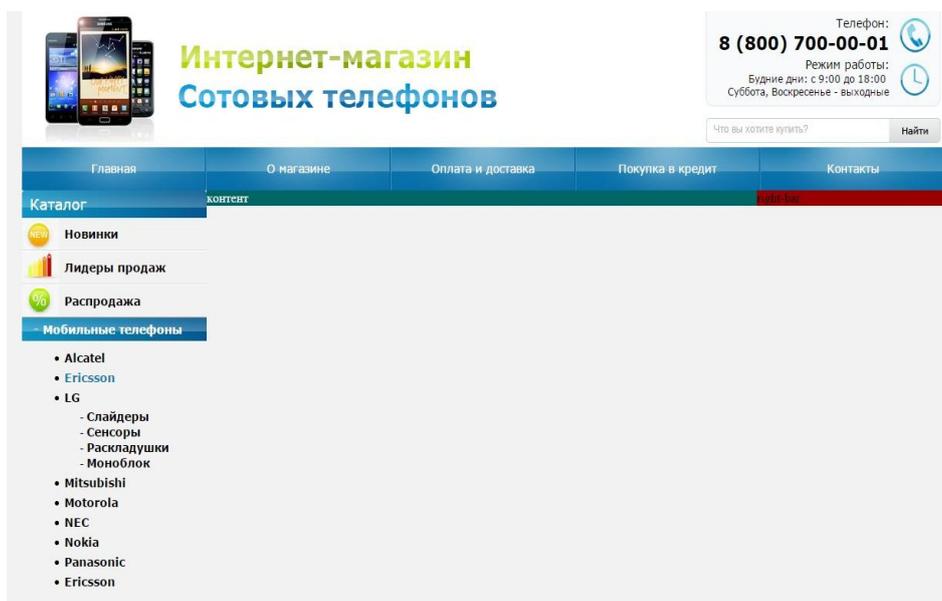


Рисунок 3.13 – Внешний вид блока навигации

Как видно из рисунка 5.14, мы получили именно то оформление меню навигации, которое было разработано web-дизайнером.

Откройте страницу в разных браузерах, убедитесь в правильном отображении в каждом из них. Таким образом, мы создали меню навигации нашего сайта.

Переходим к созданию блока контактов. Для блока контактов создадим свой класс и обернем его в контейнер `<div class="bar-contact">`.

Добавьте в файл `index.html` код, представленный на рисунке 3.14 (добавленный код заключен в красную рамку).

```

</ul>
<div class="bar-contact">
  <h3>Контакты:</h3>
  <p><strong>Телефон:</strong><br />
  <span>8 (800) 700-00-01</span></p>

  <p><strong>Режим работы:</strong><br />
  Будние дни: <br />
  с 9:00 до 18:00<br />
  Суббота, Воскресенье:<br />
  выходные</p>
</div>
</div>
<div id="right-bar">
right-bar
</div>

```

Рисунок 3.14 – Код блока контактов в файле **index.html**

Посмотрите результат в браузере (рисунок 3.15).

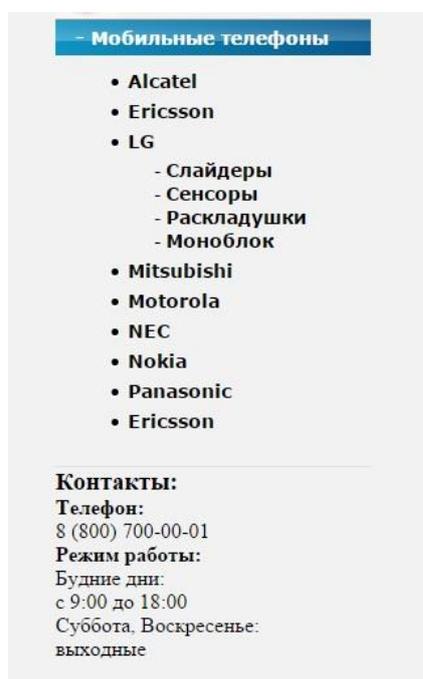


Рисунок 3.15 – Внешний вид блока «Контакты»

Пункты блока «Контакты» появились, но пока не оформлены. Откройте файл стилей и пропишите стилевое оформление для класса **.bar-contact** (рисунок 3.16).

```

        .nav-catalog ul li{
            margin:0;
        }
    }

    .bar-contact{
        border-bottom:1px solid #e1e1e1; border-top:1px solid #fff;
        padding:15px 0 20px 15px; color:#000;
        font-family:Verdana, Geneva, sans-serif;
    }

    .bar-contact h3{
        font-size:18px; font-weight:normal;
    }

    .bar-contact p{
        font-size:14px; margin:15px 0 0 0;
    }

    .bar-contact p span{
        font-size:18px; font-weight: bold;
    }
}

#right-bar{

```

Рисунок 3.16 – Код файла style.css (продолжение)

Сохраните файл, посмотрите результат в браузере (рисунок 3.17).

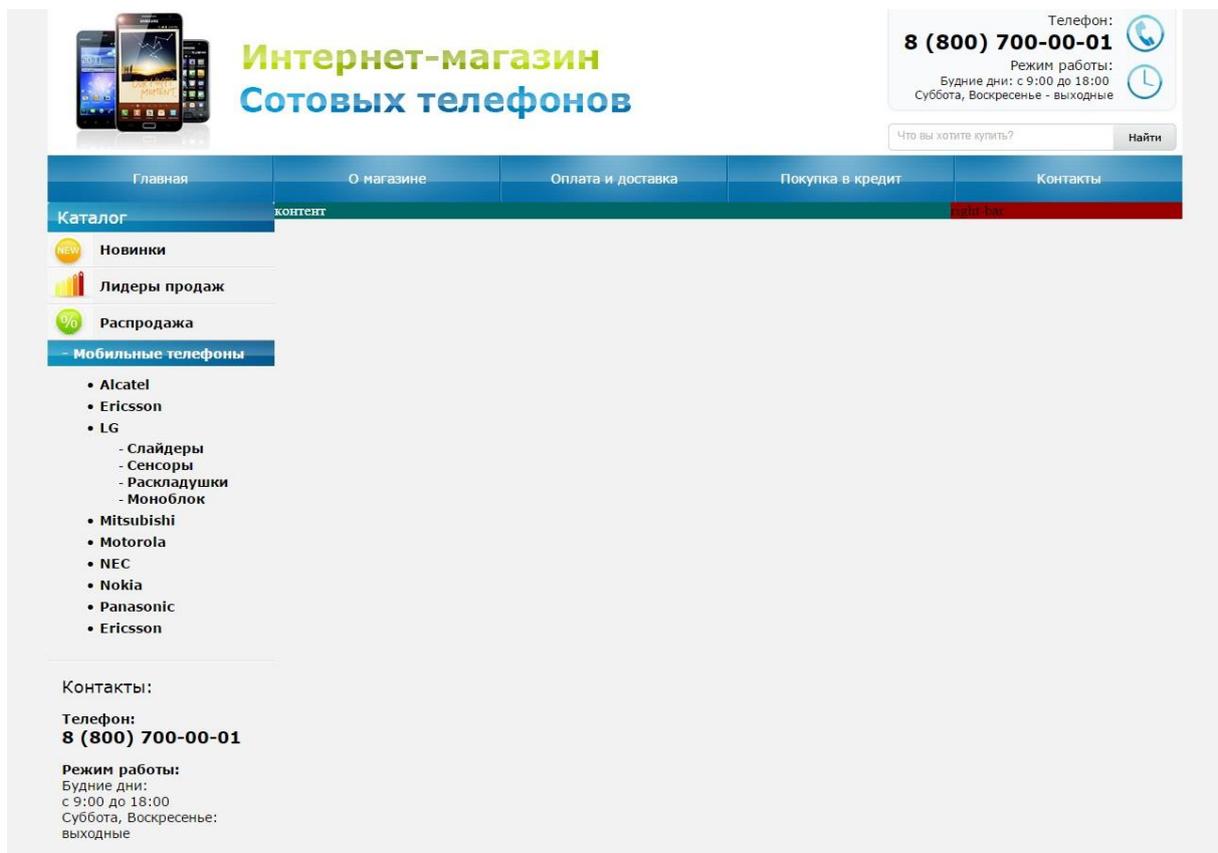


Рисунок 3.17 – Окончательный внешний вид блока «Контакты»

Сравните полученный результат с макетом, убедитесь в правильности верстки.

Продолжаем оформление левого меню, а точнее, центральной его части – блока новостей (он выделен на рисунке 3.18 зеленой рамкой).

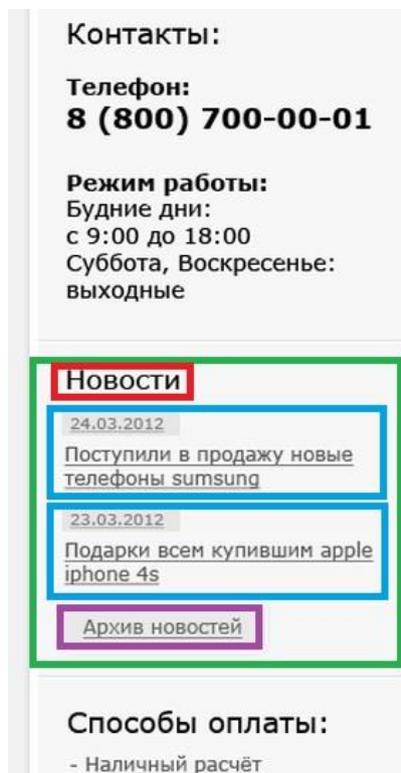


Рисунок 3.18 - Внешний вид блока «Новости» в макете

Блок новостей состоит из заголовка «Новости» (выделен красной рамкой), двух пунктов новостей (выделены голубыми рамками), в свою очередь состоящих из даты и ссылки на новость, а также ссылки на архив новостей (фиолетовая рамка). Подразумевается, что при программировании любой пункт будет «раскрываться» на произвольное количество подпунктов. Этот функционал будет реализован в следующем разделе лабораторных работ – «Программирование видимой части интернет-магазина».

Для блока новостей создадим свой класс и обернем его в контейнер `<div class="news">`.

Заголовок блока навигации «Новости» оформим в тегах `<h3>`, а остальные пункты меню оформим в виде абзацев. При этом дату оформим в рамках абзацев, используя контейнер ``.

Добавьте в код файла **index.html** текст, выделенный зеленой рамкой на рисунке 3.19.

Как видно из рисунка, заголовок «Новости» выполнен как заголовок `<h3>`, сами новости включены в абзац `<p>` и выполнены в виде гиперссылок, а дата новости дополнительно заключена в строчный контейнер «**span**». Пункт «Архив новостей» оформлен в виде гиперссылки с классом **class="news-arh"**.

```
<div class="bar-contact">
  <h3>Контакты:</h3>
  <p><strong>Телефон:</strong><br />
  <span>8 (800) 700-00-01</span></p>

  <p><strong>Режим работы:</strong><br />
  Будние дни: <br />
  с 9:00 до 18:00<br />
  Суббота, Воскресенье:<br />
  выходные</p>
</div>
<div class="news">
  <h3>Новости</h3>
  <p>
    <span>24.03.2012</span>
    <a href="#">Поступили в продажу новые
    телефоны samsung</a>
  </p>
  <p>
    <span>24.03.2012</span>
    <a href="#">Подарки всем купившим apple iphone 4s</a>
  </p>
  <a href="#" class="news-arh">Архив новостей</a>
</div>
</div>
</div>
```

Рисунок 3.19 - Код файла **index.html**

Посмотрите результат в браузере (рисунок 3.20).

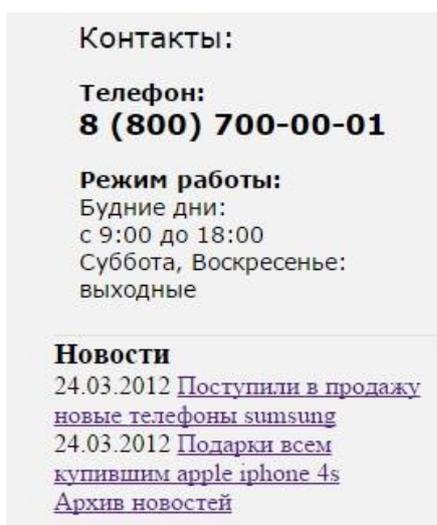


Рисунок 3.20 – Внешний вид блока «Новости»

Пункты создаваемого блока новостей созданы, но пока оформлены не в соответствие с макетом. Оформлять, как всегда мы будем в файле стилей – **style.css**. Откройте данный файл и добавьте в него код согласно рисунка 3.21.

Разберём написанные стили. Начнем с заголовка «Новости». Стилизовое оформление контейнера `<div class="news">` аналогично оформлению блока контактов (`<div class="bar-contact">`). Полностью *скопируйте стили* оформления для элементов `.bar-contact` и `.bar-contact h3` и вставьте код ниже (после стиля элемента `.bar-contact p span`), затем исправьте названия классов.

Обратите внимание, что из стиля оформления `.news` убран цвет текста `color:#000;`, а размер шрифта заголовка новости `<h3>` равен 17px. Все остальное оформление полностью совпадает.

```

.bar-contact p span{
    font-size:18px; font-weight: bold;
}

.news{
    border-bottom:1px solid #e1e1e1; border-top:1px solid #fff;
    padding:15px 0 20px 15px; font-family:Verdana, Geneva, sans-serif;
}
.news h3{
    font-size:17px; font-weight:normal;
}

#right-bar{
    background:#900;
    float:left;
    width:245px;
    margin-left:-245px;
}

```

Рисунок 3.21 - Код файла **style.css**

Посмотрите результат в браузере (рисунок 3.22).

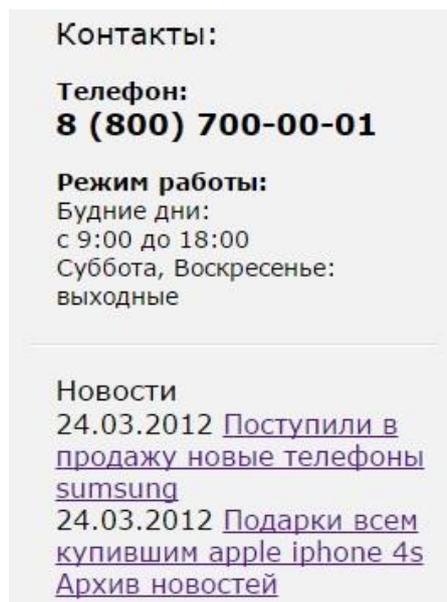


Рисунок 3.22 – Внешний вид блока новостей

Как видно из рисунка, заголовок «Новости» практически идентичен макету. Сравните полученный результат с файлом макета и убедитесь в полном совпадении оформления заголовка «Новости».

Переходим к оформлению абзацев. Откройте файл стилей **style.css** и добавьте стилевое оформление для абзацев класса **.news** (выделен зеленой рамкой на рисунке 3.23).

```
.news h3{
    font-size:17px; font-weight:normal;
}

.news p{
    margin:15px 0 0 0;
}
.news p span{
color:#7f7e7e; font-size:10px; background:#e7e7e7; padding:1px 7px;
}
.news a{
display:block; font-size:12px; color:#555; margin:5px 0 0 0;
}
.news a:hover{
text-decoration:none;
}

#right-bar{
background:#900;
float:left;
width:245px;
margin-left:-245px;
}
```

Рисунок 3.23 - Код файла стилей **style.css** в оформлении абзацев новостей

Сохраните файл стилей, обновите сайт в браузере (рисунок 3.24).



Рисунок 3.24 – Внешний вид блока «Новости»

Новости оформлены в соответствие с макетом. Осталось оформить пункт «Архив новостей». Добавьте в файл стилей код, выделенный зеленой рамкой на рисунке 3.25.

```

.news a:hover{
text-decoration:none;
}
.news-arh{
border: 1px solid #fff;
padding: 10px 15px;
text-align: center; width: 105px;
-webkit-border-radius: 5px;
-moz-border-radius: 5px;
border-radius: 5px;
-webkit-box-shadow: #eeeeee 0px 0px 7px;
-moz-box-shadow: #eeeeee 0px 0px 7px;
box-shadow: #eeeeee 0px 0px 7px;
background: #eaeaea;
behavior: url(/pie/PIE.htc);
margin:15px 0 0 0 !important;
}
#right-bar{
background:#900;
float:left;
width:245px;
margin-left:-245px;
}

```

Рисунок 3.25 - Код файла стилей **style.css** в оформлении архива новостей

Посмотрите результат в браузере (рисунок 3.26).



Рисунок 3.26 – Внешний вид блока «Новости» (окончательный)

Как видно из рисунка, мы получили именно то оформление абзацев блока новостей, которое было разработано web-дизайнером. Откройте страницу в

разных браузерах, убедитесь в правильном отображении в каждом из них. Таким образом, мы создали блок новостей нашего сайта.

Переходим к созданию блока информеров. Блок информеров состоит из трёх отдельных блоков: «Способы оплаты», «Доставка» и «Информация для Вас» (рисунок 3.27). При этом данные блоки оформлены абсолютно одинаково.

Поэтому мы создадим для всех трех пунктов блока информеров одинаковое оформление, создав класс `<div class="info">`.



Рисунок 3.27 – Внешний вид макета блока информеров

Откройте файл `index.html` и добавьте в него код, выделенный красной рамкой на рисунке 3.28.

```

        <a href="#" class="news-arh">Архив новостей</a>
    </div>

    <div class="info">
        <h3>Способы оплаты:</h3>
        <p>- <a href="#">Наличный расчёт</a></p>
        <p>- <a href="#">Пластиковой картой</a></p>
        <p>- <a href="#">Кредит</a></p>
        <p>- <a href="#">Рассчитать кредит</a></p>
        <p>- <a href="#">Безналичный расчёт</a></p>
    </div>
    <div class="info">
        <h3>Доставка:</h3>
        <p>- <a href="#">Почта России</a></p>
        <p>- <a href="#">Курьерская служба</a></p>
    </div>
    <div class="info">
        <h3>Информация для вас:</h3>
        <p>- <a href="#">Гарантия</a></p>
        <p>- <a href="#">Как выбрать телефон</a></p>
        <p>- <a href="#">Ремонт и обслуживание</a></p>
        <p>- <a href="#">Акции и новинки</a></p>
        <p>- <a href="#">Условия доставки</a></p>
    </div>
</div>

</div>
<div id="right-bar">

```

Рисунок 3.28 - Код блока информеров в файле **index.html**

Посмотрите результат в браузере (рисунок 3.29).

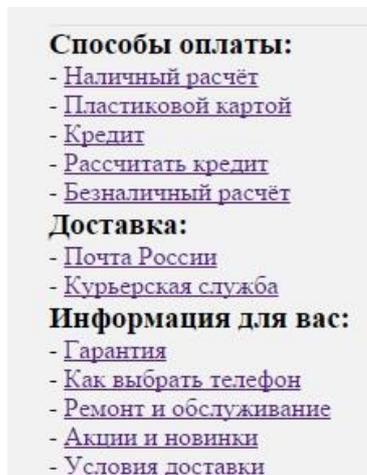


Рисунок 3.29 – Внешний вид блока информеров

Пункты блока информеров появились, но пока не оформлены. Откройте файл стилей и пропишите стилевое оформление для класса **.info** (выделен зеленой рамкой на рисунке 3.30).

```

behavior: url(/pie/PIE.htc);
margin:15px 0 0 0 !important;
}
.info{
border-bottom:1px solid #e1e1e1; border-top:1px solid #fff;
padding:15px 0 20px 15px; font-family:Verdana, Geneva, sans-serif; color:#555;
}
.info:last-child{
border-bottom:none;
}
.info h3{
font-size:18px; font-weight:normal; color:#000;
}
.info p{
margin:4px 0 0 0;
}
.info a{
font-size:12px; color:#555555;
}
.info a:hover{
text-decoration:none;
}
#right-bar{
background:#900;
}

```

Рисунок 3.30 – Оформление блока информеров в файле style.css

Сохраните файл, посмотрите результат в браузере (рисунок 3.31).

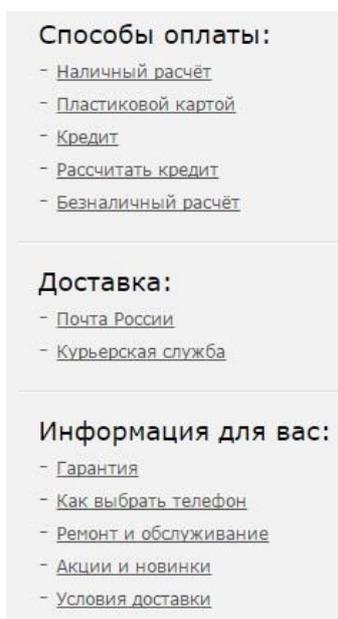


Рисунок 3.31 – Окончательный вид блока информеров

Сравните полученный результат с макетом, убедитесь в правильности верстки. Таким образом, мы полностью закончили оформление левого блока (меню) нашего сайта.

Самостоятельно. Разобраться с приведенным кодом, ответить на любые вопросы преподавателя по оформлению блока навигации.

Лабораторная работа № 4

Создание блоков основного контента, авторизации, корзины, выбора по параметрам и подвала (футера)

Переходим к оформлению центральной части главной страницы нашего сайта – блока основного контента (выделен красной рамкой на рисунке 4.1).

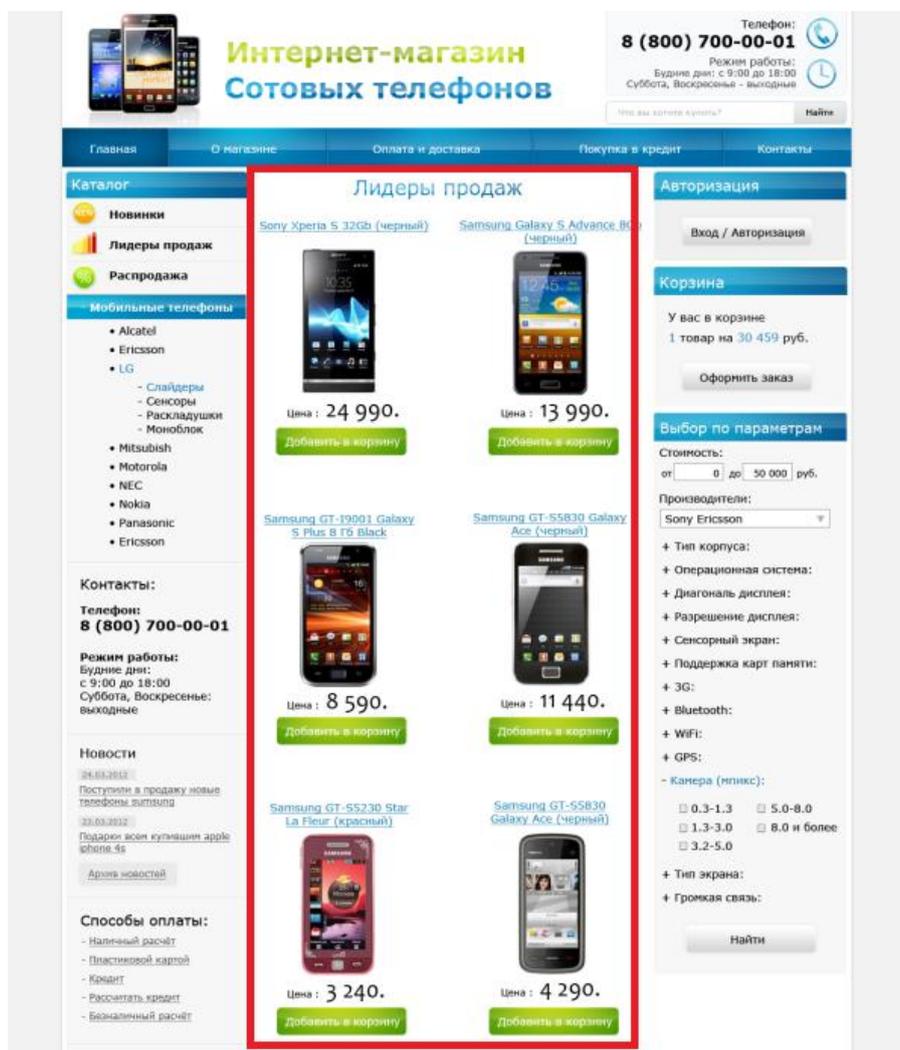


Рисунок 4.1 – Блок основного контента главной страницы сайта

Для блока основного контента создадим свой класс и обернем его в контейнер `<div class="catalog-index">`, включенный, в свою очередь в контейнер `<div class="content">`.

Заголовок «Лидеры продаж» будем оформлять картинкой – это связано со сложным градиентом, который применил к тексту web-дизайнер.

Оформление контента – телефонов разных моделей и фирм – абсолютно одинаковое, поэтому вначале мы создадим оформление только для одного телефона, а затем произведём «размножение».

Каждый телефон заключаем в контейнер `<div class="product-index">`. Внутри этого контейнера заголовком второго уровня `<h2>` (а также гиперссылкой) оформляется название телефона, ниже располагается картинка-гиперссылка с фотографией телефона, далее – в виде абзаца - слово «Цена» и заключенное в контейнер `` число – стоимость телефона. Последним элементом в контейнере `<div class="product-index">` располагается картинка – гиперссылка «Добавить в корзину».

Добавьте в код файла **index.html** текст, выделенный зеленой рамкой на рисунке 4.2.

```
<div id="contentwrapper">
  <div id="content">
    <div class="catalog-index">
      <h1></h1>
      <div class="product-index">
        <h2><a href="#">Sony Xperia S 32Gb (черный)</a></h2>
        <a href="#"></a>
        <p>Цена: <span>24 990.</span></p>
        <a href="#"></a>
      </div>
    </div>
  </div>
</div>
<div id="left-bar">
```

Рисунок 4.2 – Код блока контента в файле index.html

Посмотрите результат в браузере (рисунок 4.3).

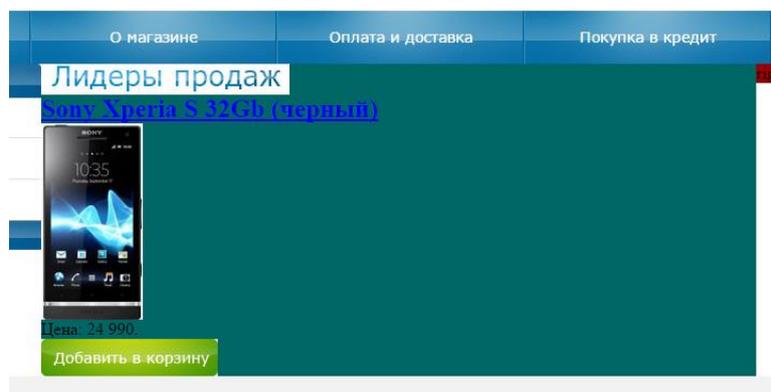


Рисунок 4.3 – Внешний вид блока контента (без стилей)

Оформлять блок контента, как всегда, мы будем в файле стилей – **style.css**. Откройте данный файл. Добавьте в код файла стилей строки, заключенные в зеленую рамку на рисунке 4.4.

```

.info a:hover{
    text-decoration:none;
}
.catalog-index{
    overflow:hidden;
}
.catalog-index h1{
    text-align:center; margin:15px 0 20px 0;
}
.product-index{
    text-align:center; width:50%; float:left; margin:0 0 60px 0;
}
.product-index h2{
    font:14px Verdana, Geneva, sans-serif; color:#2b8aba; margin:0 0 10px 0;
}
.product-index h2 a{
    color:#2b8aba;
}
.product-index h2 a:hover{
    text-decoration:none;
}
.product-index p{
    font:bold 12px Verdana, Geneva, sans-serif; margin:0 0 10px 0;
}
.product-index p span{
    font:30px Candara;
}
#right-bar{

```

Рисунок 4.4 - Код файла стилей **style.css** для блока контента

Посмотрите результат в браузере (рисунок 4.5).

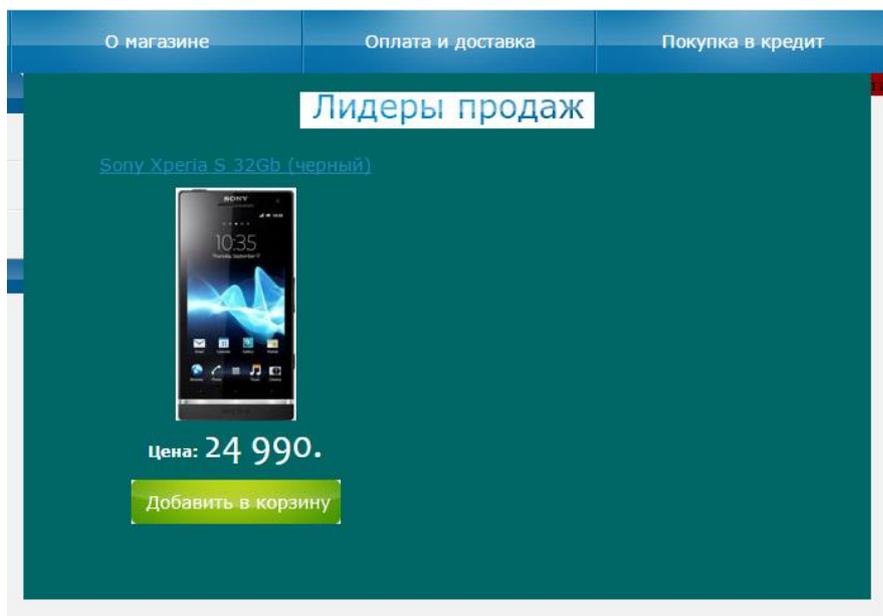


Рисунок 4.5 – Внешний вид блока контента

Цвета фона и цены не совпадают с макетом. Дело в том, что эти цвета мы выбирали (в лабораторной работе 2) для того, чтобы цветом выделить контейнеры **contentwrapper** и **content** при создании основной разметки сайта.

Откройте файл стилей, найдите в нем оформления этих контейнеров и измените код согласно рисунка: уберите белый цвет текста (**color:#fff;**), а также **background:#066;** в оформлении элемента **#content**. Добавьте белый цвет в оформлении бэкграунда (подложки) контейнера **#contentwrapper** (рисунок 4.6).

```
#contentwrapper{
    float:left; width:100%; color:#fff; background:#fff;
}

#content{
    background:#066;

    margin:0 245px 0 240px;
}

#left-bar{
```

Рисунок 4.6 – Код стилового оформления контейнера **#contentwrapper** в файле **style.css**

Сохраните файл стилей, обновите сайт в браузере (рисунок 4.7).



Рисунок 4.7 – Внешний вид блока контента

Итак, оформление одного телефона закончено. Однако их в макете главной страницы – 8 штук. Для увеличения количества телефонов откройте файл **index.html**, скопируйте контейнер `<div class="product-index">` со всем содержимым (до закрывающего тега `</div>` включительно) и вставьте ниже 7 раз.

Сохраните файл, посмотрите результат в браузере (рисунок 4.8). Как видно из рисунка, мы получили именно то оформление блока контента, которое было разработано web-дизайнером. Однако у нас все телефоны одинаковые. Ничего страшного, мы не будем изменять нашу верстку, для верстальщика представить страницу в таком виде – нормально. Открою маленькую тайну – программист при написании кода вырежет всё, кроме первого телефона, а телефоны будет выводить в цикле путем запроса к базе данных. Кроме того, описание отдельных телефонов будет выполняться в третьем разделе наших лабораторных работ – в «Администраторском разделе».

Откройте страницу в разных браузерах, убедитесь в правильном отображении в каждом из них. Таким образом, мы создали блок контента нашего сайта.

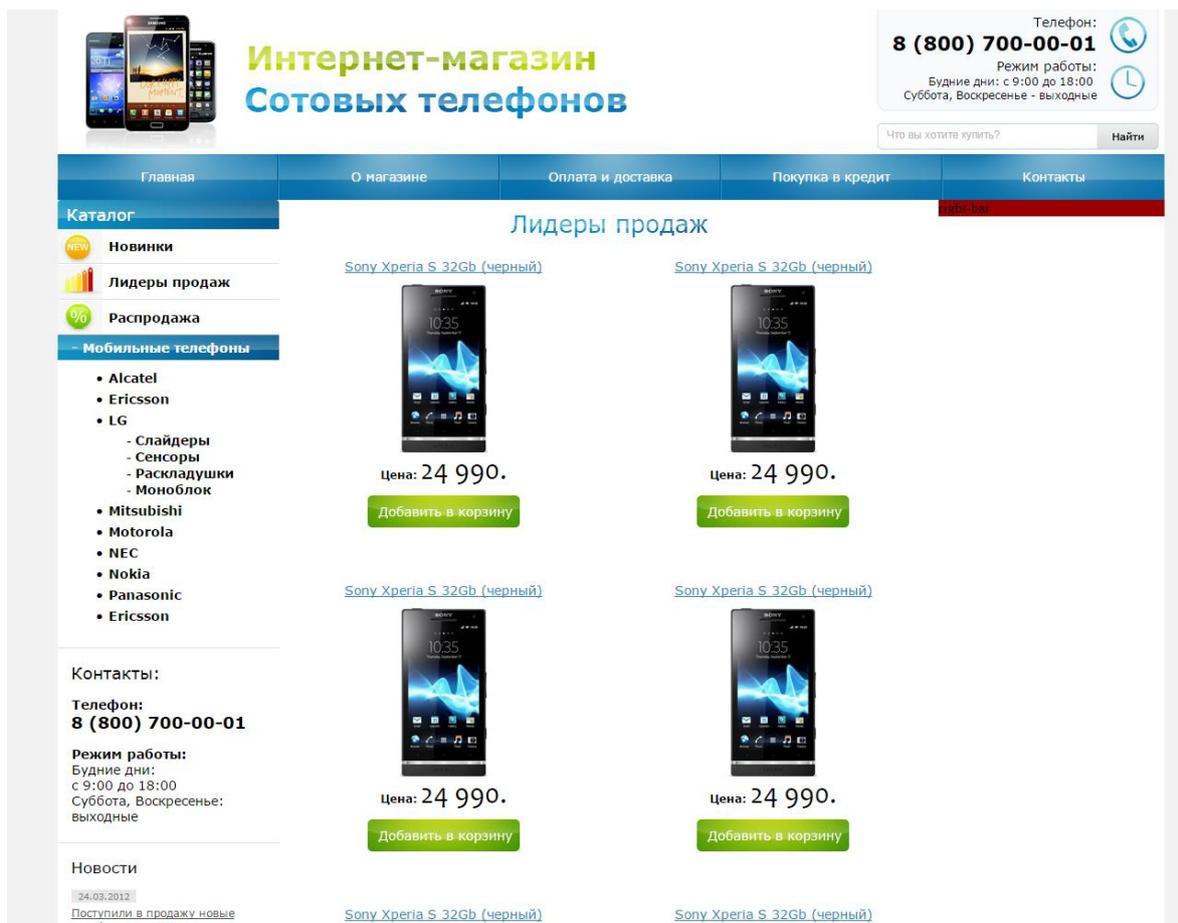


Рисунок 4.8 – Окончательный вид блока контента

Переходим к созданию правой колонки (**right-bar**) главной страницы (рисунок 4.9). Блок **right-bar** состоит из трёх отдельных блоков: «Авторизация», «Корзина» и «Выбор по параметрам». При этом данные блоки оформлены почти одинаково, по крайней мере, заголовки блоков – абсолютно одинаковы по стиливому оформлению.

Для создания правой колонки мы создадим контейнер `<div class="right-bar-cont">` (выделен желтым цветом), который вставлен в контейнер `<div id="right-bar">`, но имеет от него определенные внутренние отступы (**margin**).

Заголовки блоков «Авторизация», «Корзина» и «Выбор по параметрам» будут выполнены в виде заголовков `<h2>`, при этом заголовки обёрнуты в контейнеры `«span»`. Для блоков авторизации и корзины мы создаем свои контейнеры.



Рисунок 4.9 – Внешний вид макета правой колонки сайта

Откройте файл **index.html** и добавьте следующий код, выделенный зеленой рамкой на рисунке 4.10.

```

<div id="right-bar">
  <div class="right-bar-cont">
    <div class="enter">
      <h2><span>Авторизация</span></h2>
      <div>
        <a href="#"></a>
      </div>
    </div>
    <div class="basket">
      <h2><span>Корзина</span></h2>
      <div>
        <p>
          У вас в корзине<br />
          <span>1</span> товар на <span>30 459</span> руб
        </p>
        <a href="#"></a>
      </div>
    </div>
  </div>
</div>

```

Рисунок 4.10 – Код правой колонки (авторизация и корзина) в файле **index.html**

Сохраните файл, посмотрите результат в браузере (рисунок 4.11).

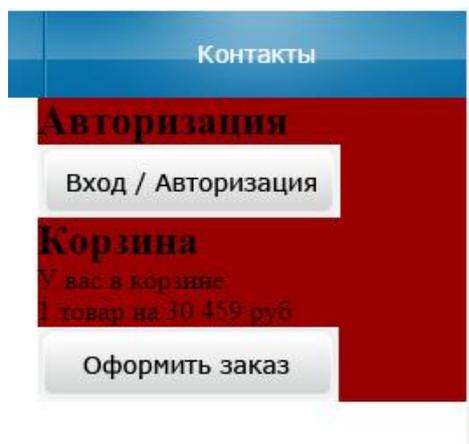


Рисунок 4.11 – Внешний вид блоков «Авторизация» и «Корзина»

Оформление не соответствует макету, для этого откройте файл стилей **style.css** и добавьте в него следующий код, заключенный в зеленую рамку на рисунке 4.12 (не забудьте убрать строчку с красным бэкграундом в оформлении контейнера `<div id="right-bar">` - в файле стилей он представлен селектором `#right-bar`).

Маленькое замечание: не надо писать код вручную. Дело в том, что оформление заголовков левой колонки («Каталог») и правой колонки («Авторизация», «Корзина» и «Выбор по параметрам») абсолютно идентичны. Поэтому *найдите* в файле стилей код оформления **.left-bar-cont h2**, **.left-bar-cont h2 span** и скопируйте данный код, вставьте его в нужное место и исправьте названия **.left-bar-cont h2** и **.left-bar-cont h2 span** соответственно на **.right-bar-cont h2** и **.right-bar-cont h2 span**.

```

#right-bar{
  background:#900;
  float:left;
  width:245px;
  margin-left:-245px;
}
.right-bar-cont{
  margin:6px 6px 0 0;
}
.right-bar-cont h2{
  color:#fff; font:normal 18px Verdana, Geneva, sans-serif;
  background: rgb(14,148,195); /* Old browsers */
  background: -moz-linear-gradient(left, rgba(14,148,195,1) 0%,
  rgba(4,88,144,1) 100%); /* FF3.6+ */
  background: -webkit-gradient(linear, left top, right top,
  color-stop(0%,rgba(14,148,195,1)), color-stop(100%,
  rgba(4,88,144,1))); /* Chrome,Safari4+ */
  background: -webkit-linear-gradient(left, rgba(14,148,195,1) 0%,
  rgba(4,88,144,1) 100%); /* Chrome10+,Safari5.1+ */
  background: -o-linear-gradient(left, rgba(14,148,195,1) 0%,
  rgba(4,88,144,1) 100%); /* Opera 11.10+ */
  background: -ms-linear-gradient(left, rgba(14,148,195,1) 0%,
  rgba(4,88,144,1) 100%); /* IE10+ */
  background: linear-gradient(to right, rgba(14,148,195,1) 0%,
  rgba(4,88,144,1) 100%); /* W3C */
  filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#0e94c3',
  endColorstr='#045890',GradientType=1 ); /* IE6-9 */
  border-radius:4px 4px 0 0;
}
.right-bar-cont h2 span{
  display:block; background:url(images/glyanec.png) top repeat-x;
  padding:5px 0 5px 10px;
}

```

Рисунок 4.12 – Код файла стилей **style.css**

Сохраните файл стилей, посмотрите результат в браузере (рисунок 4.13).

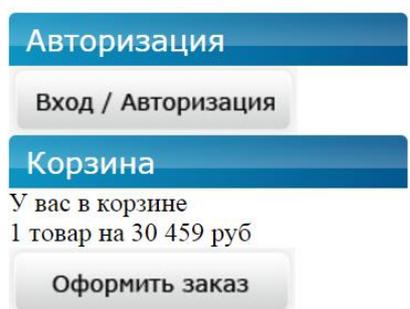


Рисунок 4.13 – Внешний вид блоков «Авторизация» и «Корзина» после применения стилей

Заголовки разделов («Авторизация» и «Корзина») приняли требуемый вид. Переходим к оформлению содержимого данных разделов.

Откройте файл стилей **style.css**, добавьте в него следующий код (рисунок 4.14).

```
.right-bar-cont h2 span{
    display:block; background:url(images/glyanec.png) top repeat-x;
    padding:5px 0 5px 10px;
}
.enter, .basket, {
margin: 0 0 10px 0;
}
.enter div, .basket div {
    background:#f0f0f0; padding:15px 0;
}
.enter div img, .basket div img {
    display:block; margin:0 auto;
}
.basket div p{
    padding:0 0 15px 20px; font:15px/1.8 Verdana,Geneva,sans-serif;
}
.basket div p span{
    color:#2c88b9;
}
```

Рисунок 4.14 – Код файла стилей для блоков «Авторизация» и «Корзина»

Сохраните файл стилей, посмотрите результат в браузере (рисунок 4.15).

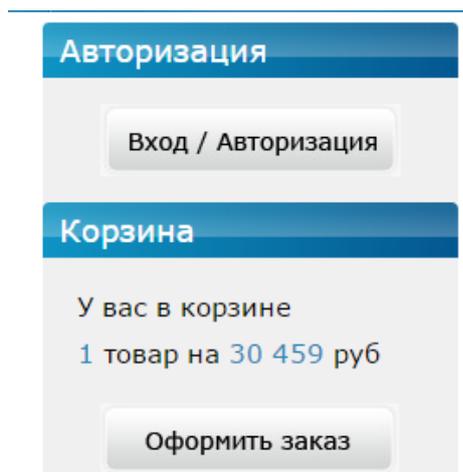


Рисунок 4.15 – Внешний вид блоков «Авторизация» и «Корзина»

Посмотрим на сайт целиком (рисунок 4.16).

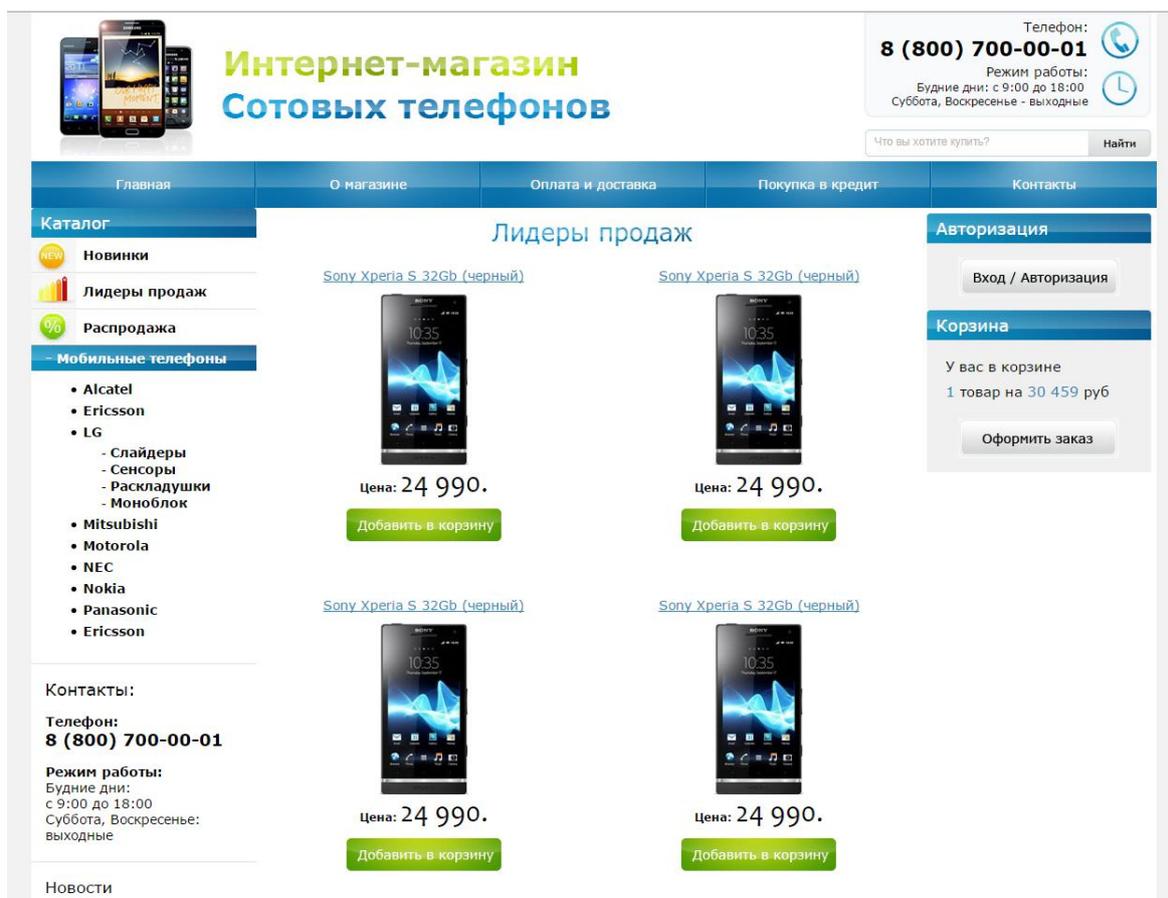


Рисунок 4.16 – Внешний вид главной страницы с блоками «Контент», «Авторизация» и «Корзина»

Сравните полученный результат с макетом, убедитесь в правильности верстки. Таким образом, мы закончили оформление блоков основного контента, авторизации и корзины нашего сайта.

Переходим к оформлению блока «Выбор по параметрам» главной страницы нашего сайта (выделен красной рамкой на рисунке 4.17).

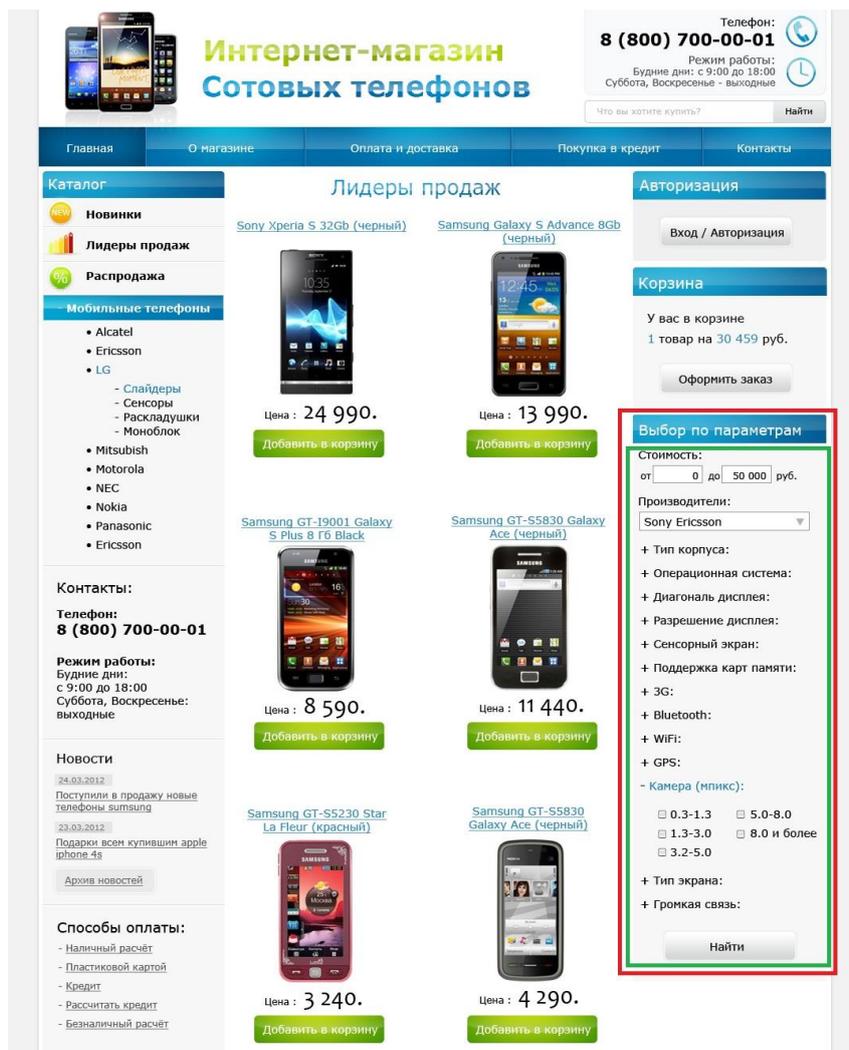


Рисунок 4.17 – Внешний вид блока «Выбор по параметрам» макета сайта

Заголовок блока «Выбор по параметрам» будет выполнен в виде заголовка **<h2>**, при этом заголовок обернут в контейнеры ****. Именно эти элементы были использованы ранее при оформлении идентичных по внешнему виду пунктов правой колонки «Авторизация» и «Корзина».

Для всего блока «Выбор по параметрам» (выделен на рисунке выше красной рамкой) создадим свой класс и обернем его в контейнер **<div class="share-search">**, включенный, в свою очередь, в контейнер **<div class="right-bar-cont">**. Само содержание пункта меню «Выбор по параметрам» (показано на рисунке 4.17 зеленой рамкой) мы оформим в виде формы, которая, в свою очередь, обернута в контейнер **<div>**.

Откройте файл **index.html** и добавьте следующий код (выделенный красной рамкой на рисунке 4.18).

```
    У вас в корзине<br />
    <span>1</span> товар на <span>30 459</span> руб
  </p>
  <a href="#"></a>
</div>
</div> <!-- Закрывающий тег дива basket -->

<div class="share-search">
<h2><span>Выбор по параметрам</span></h2>
  <div>
    <form method="post" action="">
      <p>Стоимость:</p>
      от <input class="podbor-price" type="text" name="start-price" />
      до <input class="podbor-price" type="text" name="end-price" />
      руб.
      <br /><br />
      <p>Производители:</p>
      <select>
        <option>Ericsson</option>
        <option>Alcatel</option>
        <option>Mitsubish</option>
        <option>Motorola</option>
        <option>NEC</option>
        <option>Nokia</option>
      </select>
      <input class="podbor" type="image" src="images/podbor.jpg" />
    </form>
  </div>
</div>

</div> <!-- Закрывающий тег дива right-bar-cont -->
</div> <!-- Закрывающий тег дива right-bar -->
```

Рисунок 4.18 – Код файла **index.html** для блока «Выбор по параметрам»

Обратите внимание, в код добавлены комментарии – для удобства ориентации в месте, куда необходимо вставить код (в блокноте Notepad++ комментарии подсвечены зеленым цветом). Сохраните файл, посмотрите результат в браузере (рисунок 4.19).



Рисунок 4.19 – Блок «Выбор по параметрам» без стилового оформления

Оформление заголовка «Выбор по параметрам» соответствует макету (стиль оформления унаследован от ранее использованного стиля оформления заголовка `.right-bar-cont h2 span`, а контейнер `<div class="share-search">` входит в описанный ранее в стилях контейнер `<div class="right-bar-cont">`). Таким образом, для заголовка нашего блока мы не будем создавать свой стиль.

Остальные элементы блока требуют стилового оформления. Для этого откройте файл стилей `style.css` и добавьте в него следующий код. Во-первых, отступы блоков «Авторизация», «Корзина» и «Выбор по параметрам» друг от друга должны быть одинаковыми.

Мы не будем прописывать отступы `margin` отдельно для класса `share-search`, а добавим этот класс к списку с отступами (зеленая рамка на рисунке 4.20).

```

.right-bar-cont h2 span{
    display:block; background:url(images/glyanec.png) top repeat-x;
    padding:5px 0 5px 10px;
}
.enter, .basket, .share-search {
margin:0 0 10px 0;
}
.enter div, .basket div {
    background:#f0f0f0; padding:15px 0;
}

```

Рисунок 4.20 – Код файла `style.css`

Далее нам необходимо выполнить стилевое оформление класса **share-search**. Опускаемся в самый низ файла **style.css** и добавляем следующий код (выделен на рисунке 4.21 зеленой рамкой):

```
.basket div p span{
    color:#2c88b9;
}

.share-search div{
    font-size:12px; font-family:Verdana, Geneva, sans-serif;
    padding:10px 10px 25px 10px; background:#f0f0f0;
}

.share-search div p{
    font-size:14px; margin:0 0 8px 0;
}

.share-search div select{
    width:200px; font-size:14px; padding:5px;
    font-family:Verdana, Geneva, sans-serif;
}

.podbor-price{
    width:60px; text-align:right; font:12px Verdana, Geneva, sans-serif;
    padding:3px; border:1px solid #959b9e;
}

.podbor{
    display:block; width:170px; margin:20px auto 0 auto;
}
```

Рисунок 4.21 – Код файла **style.css** (продолжение)

Сохраните файл стилей, посмотрите результат в браузере (рисунок 4.22).

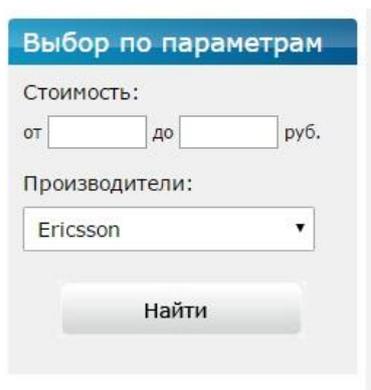


Рисунок 4.22 - Внешний вид блока «Выбор по параметрам»

Таким образом, разделы правой колонки сайта приняли требуемый вид.

Переходим к оформлению футера (подвала) сайта. Внешний вид подвала сайта (футера) приведен на рисунке 4.23.

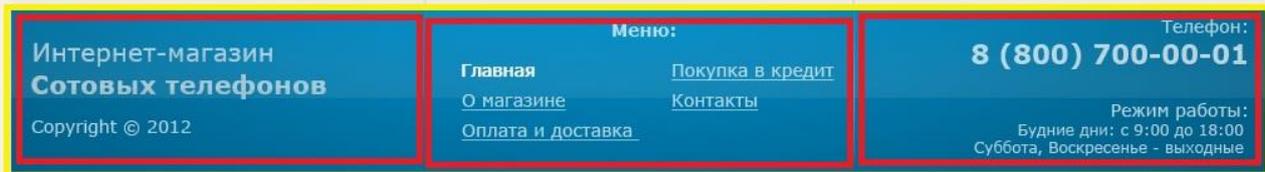


Рисунок 4.23 - Внешний вид футера сайта в макете

Футер сайта входит в общее пространство сайта (прописывается до закрывающего тега дива main) и представляет собой пространство (завернутое в контейнер `<div>`), которое, в свою очередь, разбивается на три подпространства (для каждого создается свой контейнер `<div>`).

Откройте файл **index.html** и добавьте в него код (почти в конец файла) код в соответствии с рисунком 4.24.

```

        </div> <!-- Закрывающий тег дива right-bar-cont -->
    </div> <!-- Закрывающий тег дива right-bar -->

    <div class="clr"></div>
    <div class="footer">
        <div class="flogo">
            <a href="/"></a>
            <p>Copyright © 2012</p>
        </div>
        <div class="fphone">
            <h2>Телефон:</h2>
            <h1>8 (800) 700-00-01</h1>

            <h2>Режим работы:</h2>
            <p>Будние дни: с 9:00 до 18:00<br />
            Суббота, Воскресенье - выходные </p>
        </div>
        <div class="fmenu">
            <p>Меню:</p>
            <ul>
                <li><a href="#">Главная</a></li>
                <li><a href="#">О магазине</a></li>
                <li><a href="#">Оплата и доставка</a></li>
            </ul>
            <ul>
                <li><a href="#">Покупка в кредит</a></li>
                <li><a href="#">Контакты</a></li>
            </ul>
        </div>
    </div>
</div> <!-- Закрывающий тег дива main -->
</body>
</html>

```

Рисунок 4.24 – Код футера в файле **index.html**

Первый див, который мы прописали – это контейнер `<div class="clr">`, при этом имя класса `clr` происходит от английского слова “clear” – очистка. То есть при помощи данного контейнера (в настоящее время пустого) мы очищаем все настройки, связанные с контейнерами - в первую очередь нам нужно сбросить выравнивание контейнеров, примененное для центральной части сайта (left-bar, content, right-bar). Если этого не сделать, могут возникнуть проблемы с отображением контейнеров (когда новый стиль «наложится» на старый). Лучше очистку прописать в начале файла стилей. Откройте файл `style.css` и в начале файла добавьте в него следующий код (выделен зеленой рамкой на рисунке 4.25).

```
/*{
    margin:0; padding:0;
}
img{
    border:none;
}
body{
    background:#f2f2f2;
}
.clr{
    clear:both;
}
.main{
    background:#fff; min-width:980px; max-width:1200px; margin:0 auto;
}
```

Рисунок 4.25 – Код блока очистки стилей в файле `style.css`

Если этого не сделать, то в некоторых браузерах футер может не отображаться.

Теперь посмотрите результат в браузере (рисунок 4.26).



Рисунок 4.26 – Внешний вид футера без стилового оформления

Необходимо прописать стиловое оформление сайта. Вначале оформим стили для классов **footer**, **flogo** и **fphone**.

Откройте файл **style.css** и в конце файла (в самом низу) добавьте в него следующий код (выделен зеленой рамкой на рисунке 4.27).

```

        .podbor{
            display:block; width:170px; margin:20px auto 0 auto;
        }
    .footer{
        background:#006b98 url(images/footer-bg.jpg) top repeat-x;
        overflow:hidden; color:rgba(255, 255, 255, 0.7); padding:0 0 10px 0;
    }
    .flogo{
        float:left; margin:35px 0 0 20px;
    }
    .flogo p{
        margin:10px 0 0 0; font:bold 14px Verdana, Geneva, sans-serif;
    }
    .fphone{
        float:right; text-align:right; font:12px Verdana, Geneva, sans-serif;
        margin:15px 20px 0 0;
    }
    .fphone h2{
        font-size:14px;
    }
    .fphone h1{
        font-size:21px; margin-bottom:25px;
    }

```

Рисунок 4.27 – Код блока «**footer**» стилей в файле **style.css**

Далее создадим стиль для центральной части футера – для контейнера с классом **fmenu** (рисунок 4.28).

```
.fphone h1{
    font-size:21px; margin-bottom:25px;
}

.fmenu{
    font:14px Verdana, Geneva, sans-serif; overflow:hidden;
    margin:15px auto 0 auto; width:360px;
}

.fmenu p{
    text-align:center; font-weight:bold;
}

.fmenu ul{
    float:left; list-style:none; margin:15px;
}

.fmenu ul li{
    margin:0 0 5px 0;
}

.fmenu ul a{
    color:rgba(255, 255, 255, 0.7); font-weight:bold;
}

.fmenu ul a:hover{
    color:rgba(255, 255, 255, 1); font-weight:bold;
    text-decoration:none;
}
```

Рисунок 4.28 – Код класса **fmenu** в файле **style.css**

Сохраните файл стилей, посмотрите результат в браузере (рисунок 4.29).



Рисунок 4.29 – Внешний вид подвала сайта (футера)

Посмотрите на сайт целиком. Сравните полученный результат с макетом, убедитесь в правильности верстки.

Таким образом, в данной лабораторной работе мы закончили оформление блоков основного контента, авторизации, корзины и футера нашего сайта.

В итоге главная страница нашего сайта полностью свёрстана.

Самостоятельно. Разобраться с приведенным кодом, ответить на любые вопросы преподавателя по оформлению блока основного контента.

Лабораторная работа № 5

Вёрстка страниц линейного и табличного просмотра каталогов товаров ([index-line.html](#) и [index-table.html](#)), а также текстовой страницы сайта

В прошлой лабораторной работе мы закончили верстку главной страницы нашего сайта ([index.html](#)). Теперь можно приступить к созданию других страниц сайта. Рассмотрим подробно структуру файлов линейного и табличного просмотра товаров. Откройте в просмотрщике графических файлов (в любом т.н. «вьюере») файлы «каталог_линейный вид.jpg» и «каталог_табличный вид.jpg» (они содержатся в папке «**maket**»). Лучше же будет просмотреть данные файлы в формате фотошоп-файлов (файлов с теми же именами, но с расширениями **.psd**). Внешний вид данных файлов во вьюере представлен на рисунках 5.1 и 5.2.

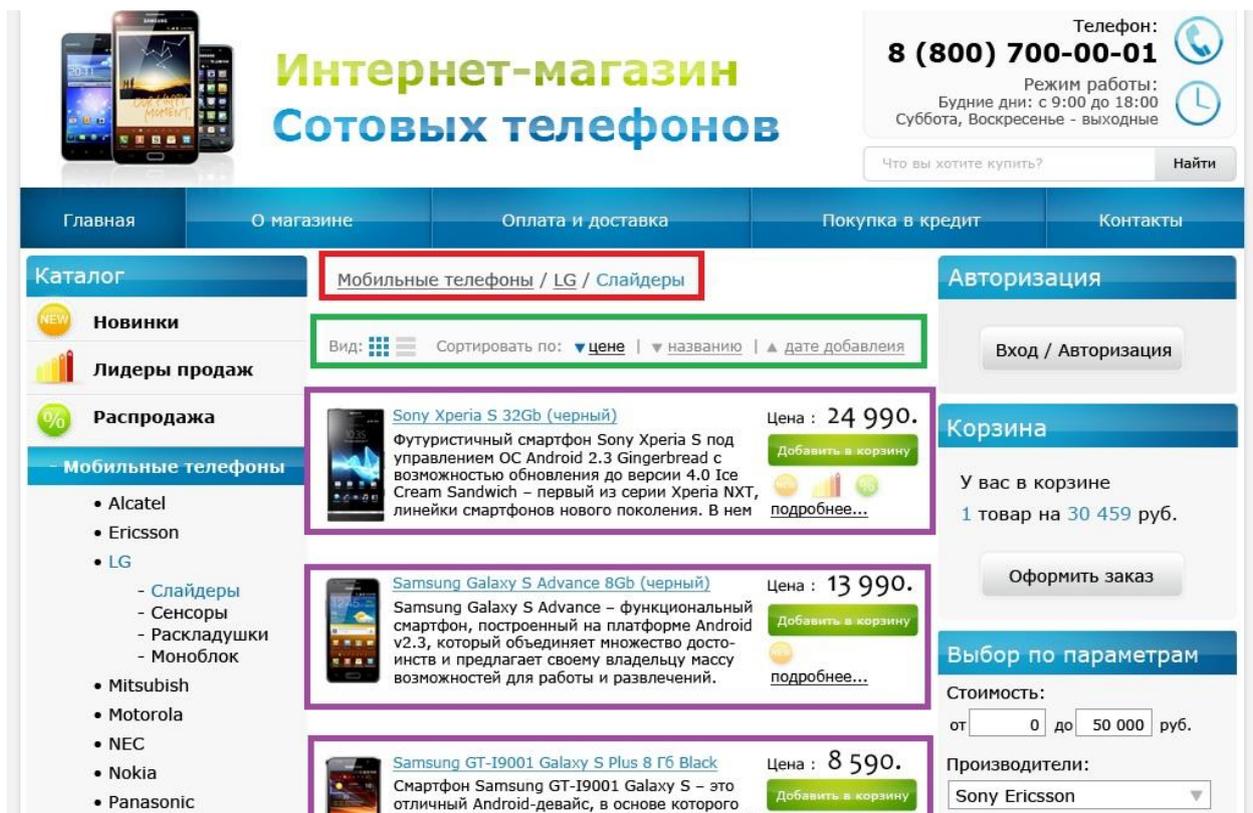


Рисунок 5.1 – Линейный вид страницы просмотра товаров

Анализ данных файлов показывает, что они имеют на 80% одинаковый внешний вид. Во-первых, у них абсолютно одинаковые хедер, футер, верхнее меню, а также левая и правая колонки. Отличия идут только в разделе «контент». Во-вторых, два элемента у этих файлов в блоке основного контента также полностью совпадают:

- 1) Так называемая «цепочка навигации» (выделена на обоих рисунках красной рамкой), показывающая пользователю путь к разделу, в котором пользователь сейчас находится. Другое распространенное название цепочки навигации – «хлебные крошки».
- 2) Блок выбора вида представления товара (линейный, табличный) и блоки сортировок (по цене, названию, дате). Данные блоки заключены на рисунках в зеленую рамку.

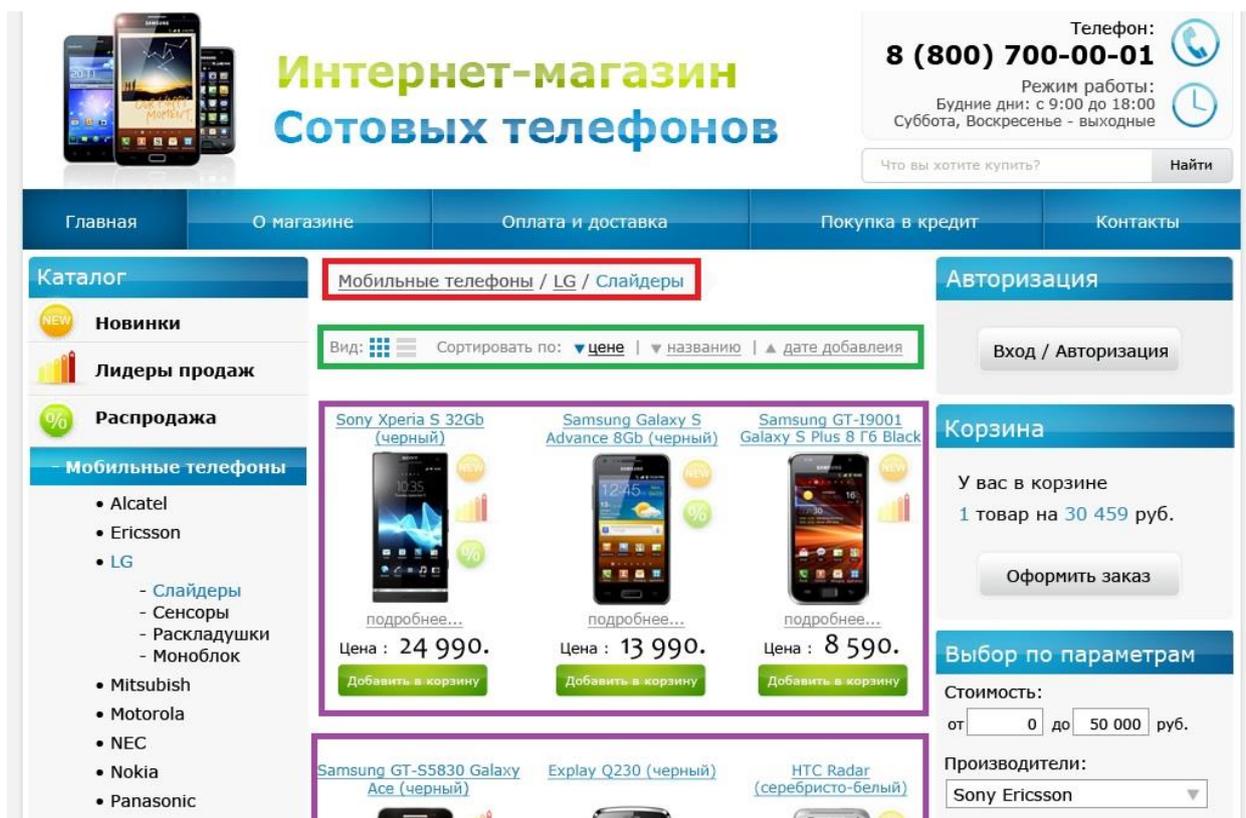


Рисунок 5.2 – Табличный вид страницы просмотра товаров

Отличия между файлами показаны в блоках, заключенных в фиолетовые рамки.

В качестве основы наших будущих файлов возьмем файл главной страницы сайта. Откройте в блокноте файл **index.html**. Сохраните его (выбрав в блокноте пункт меню Файл/Сохранить как..) с именем **index-line.html**.

Далее нам необходимо очистить поле контента страницы. Удалите всё содержимое в коде файла **index-line.html**, расположенного между открывающим и закрывающим тегами контейнера **<div class="catalog-index">**, как это показано на рисунке 5.3 (красной рамкой выделена пустая строка). Сохраните изменения, посмотрите результат в браузере.

Неожиданно, не правда ли? Это лишний раз подтверждает мои слова о «капризности» контейнеров **<div>**. Впрочем, если вместо пустой строки написать любой текст (несколько любых символов), то внешний вид страницы вернется к привычному нам виду. Ну «не любят» контейнеры **<div>** пустоту!

```
<div id="contentwrapper">
  <div id="content">
    <div class="catalog-index">
      
    </div>
  </div>
</div>
<div id="left-bar">
  <div class="left-bar-cont">
```

Рисунок 5.3 – Очищенное от контента содержимое страницы

Теперь необходимо заполнить содержимое страницы (рисунок 5.4 – в красной рамке).

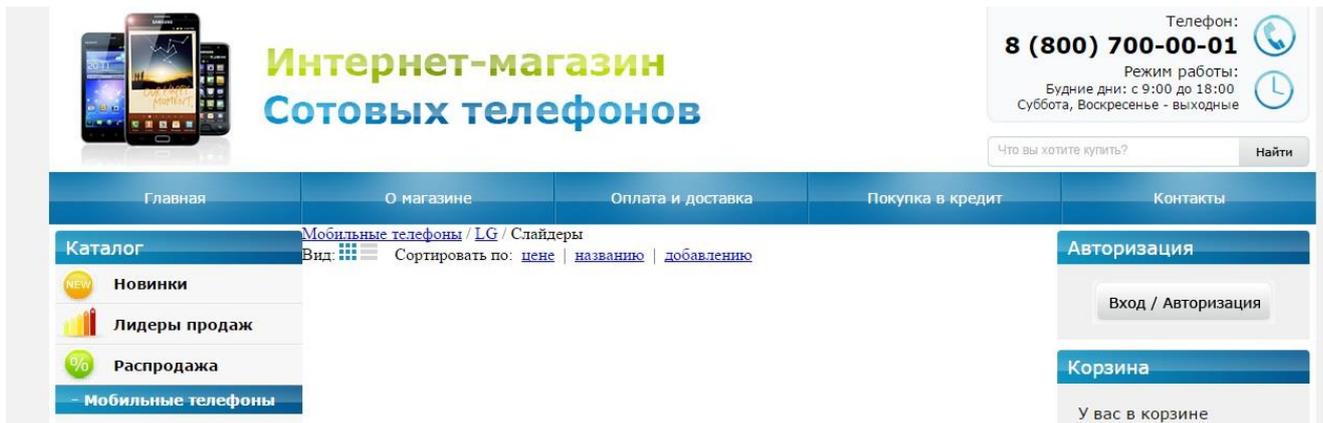


Рисунок 5.5 – Внешний вид первых двух пунктов центрального контента

Оформить страницу `index-line.html` в требуемом согласно макету виде нам помогут стили. Откройте файл `style.css` и добавьте в самый низ страницы файла следующий код (рисунок 5.6, выделено зеленой рамкой).

```

353     .fmenu ul a:hover{
354         color:rgba(255, 255, 255, 1); font-weight:bold;
355         text-decoration:none;
356     }
357
358     /*      Каталог-line      */
359     .kroshka{
360         font:14px Verdana, Geneva, sans-serif; color:#555555; margin:15px 0 30px 15px;
361     }
362     .kroshka span{
363         color:#2b8aba;
364     }
365     .kroshka a{
366         color:#555;
367     }
368     .kroshka a:hover{
369         text-decoration:none;
370     }
371

```

Рисунок 5.6 – Стиль для «хлебных крошек»

Добавьте ниже код для стилизового оформления вида сортировок (рисунок 5.7).

Сохраните изменения, посмотрите результат в браузере (рисунок 5.8). Как видно из рисунка, первые два пункта центрального контента линейного вида просмотра товаров полностью соответствуют макету.

```

.kroshka a:hover{
    text-decoration:none;
}

.vid-sort{
    font:12px Verdana, Geneva, sans-serif; color:#888; background:#fafbf3;
    border-bottom:1px solid #dee7ec; margin:0 15px 30px 15px; padding:10px 0 10px 2px;
}

.vid-sort img{
    margin:0 2px; position:relative; top:3px;
}

.vid-sort a{
    color:#888;
}

.vid-sort a:hover{
    text-decoration:none;
}

a.sort-top{
    background:url(images/sort-top.gif) left 6px no-repeat; padding:0 0 0 12px;
}

a.sort-bot{
    background:url(images/sort-bot.gif) left 6px no-repeat; padding:0 0 0 12px;
}

a.sort-top-act{
    background:url(images/sort-top-act.gif) left 6px no-repeat; padding:0 0 0 12px; color:#000;
}

a.sort-bot-act{
    background:url(images/sort-bot-act.gif) left 6px no-repeat; padding:0 0 0 12px; color:#000;
}

```

Рисунок 5.7 – Стиль для «вида сортировки»



Рисунок 5.8 – Внешний вид первых двух пунктов страницы


```

3 a.sort-bot-act{
4     background:url(images/sort-bot-act.gif) left 6px no-repeat; padding:0 0 0 12px; color:#000;
5 }
6
7 product-line{
8     overflow:hidden; margin:0 15px 35px 15px;
9 }
10
11 .product-line-img{
12     float:left;
13 }
14
15 .product-line-price{
16     float:right;
17 }
18
19 .product-line-price p{
20     font:12px Verdana, Geneva, sans-serif; margin:0 0 5px 0;
21 }
22
23 .product-line-price p span{
24     font:24px Candara; line-height:0;
25 }
26
27 .product-line-price p a{
28     color:#000;
29 }
30
31 .product-line-price p a:hover{
32     text-decoration:none;
33 }
34
35 .product-line-price div{
36     margin:5px 0 0 0; height:26px;
37 }
38
39 .product-line-price div img{
40     margin:0 3px;
41 }

```

Рисунок 5.10 – Стиль оформления линейного вида телефона

```

1 .product-line-price div img{
2     margin:0 3px;
3 }
4
5 .product-line-opis{
6     font:12px Verdana, Geneva, sans-serif; padding:0 130px 0 60px;
7 }
8
9 .product-line-opis h2{
10     font:12px Verdana, Geneva, sans-serif; color:#2b8aba;
11     margin:0 0 7px 0; line-height:0.8;
12 }
13
14 .product-line-opis h2 a{
15     color:#2b8aba;
16 }
17
18 .product-line-opis h2 a:hover{
19     text-decoration:none;
20 }

```

Рисунок 5.11 – Стиль оформления линейного вида телефона

(продолжение)

Сохраните изменения, посмотрите результат в браузере (рисунок 5.12).

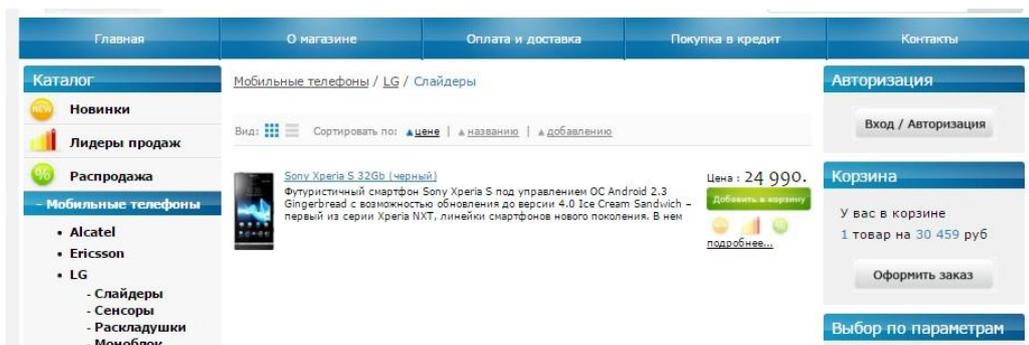


Рисунок 5.12 – Линейный вид просмотра отдельного телефона

Как видно из рисунка, оформление отдельного телефона полностью соответствует линейному виду, представленному в макете. Однако в макете телефонов представлено существенно больше. Для того, чтобы увеличить число телефонов на странице, нам необходимо в файле **index-line.html** найти контейнер **<div class="product-line">** и полностью скопировать содержимое данного контейнера в буфер обмена, после чего 7-10 раз подряд вставить скопированный фрагмент ниже контейнера-оригинала.

Сохраните изменения, посмотрите результат в браузере (рисунок 5.13).

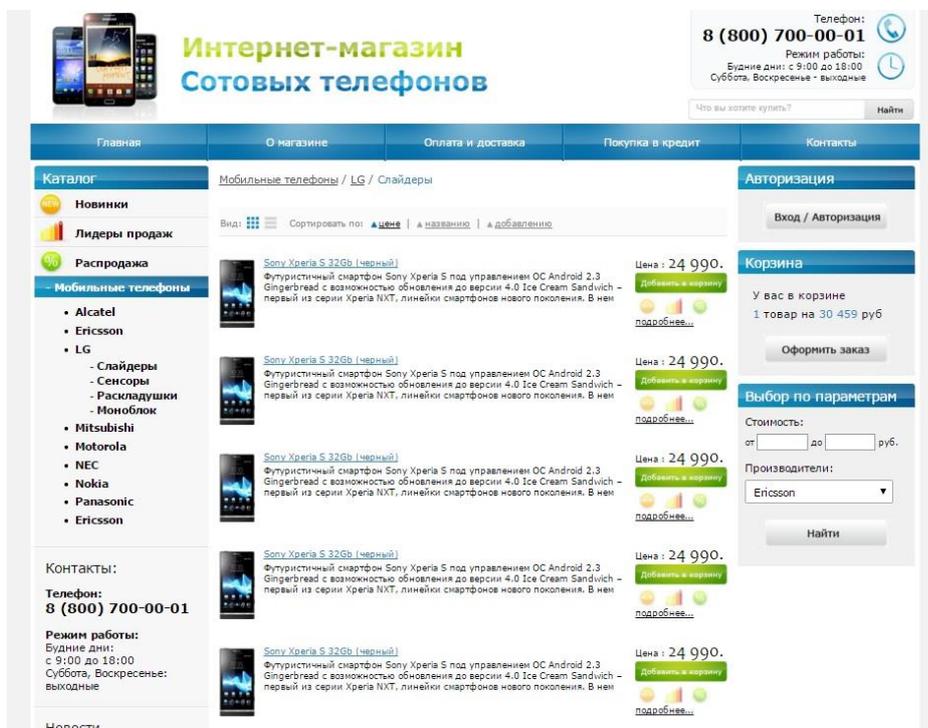


Рисунок 5.13 – Линейный вид просмотра телефонов

Как видно из рисунка, оформление линейного вида страницы просмотра телефонов почти полностью совпадает с макетом. За единственным исключением – в макете есть блок пагинации (постраничной навигации) для просмотра телефонов, когда их слишком много и они не входят всем комплектом на одну страницу (рисунок 5.14).



Рисунок 5.14 – Постраничная навигация телефонов в макете сайта

Добавим в код файла **index-line.html** следующие строки (выделены красной рамкой на рисунке 5.15).

```
<p>футуристичный смартфон Sony Xperia S под управлением ОС Android 2.3
</div>
<div class="pager">
  <a href="#"></a>
  1
  <a href="#">2</a>
  <a href="#">3</a>
  <a href="#">4</a>
  <a href="#">5</a>
  <a href="#">...</a>
  <a href="#">27</a>
  <a href="#"></a>
</div>
</div>
</div>
</div>
<div id="left-bar">
```

Рисунок 5.15 – Код страничного просмотра телефонов

Для стилевого оформления кода страничного просмотра откройте файл **style.css** и добавьте в низ файла код на рисунке 5.16.

```

.pager{
background:#fafbfb; border-top:1px solid #eb1f4; font:15px Verdana, Geneva, sans-serif;
color:#2b8aba; padding:10px; margin:15px;
}

.pager a{
color:#888888; font-size:14px; padding:0 3px;
}
.pager a:hover{
text-decoration:none; color:#2b8aba;
}
.pager img{
position:relative; top:5px;
}

```

Рисунок 5.16 – Стиль для оформления постраничного просмотра телефонов

Сохраните изменения, посмотрите результат в браузере (рисунок 5.17). Как видно из рисунка, линейный вид просмотра телефонов полностью совпадает с представленным в макете.

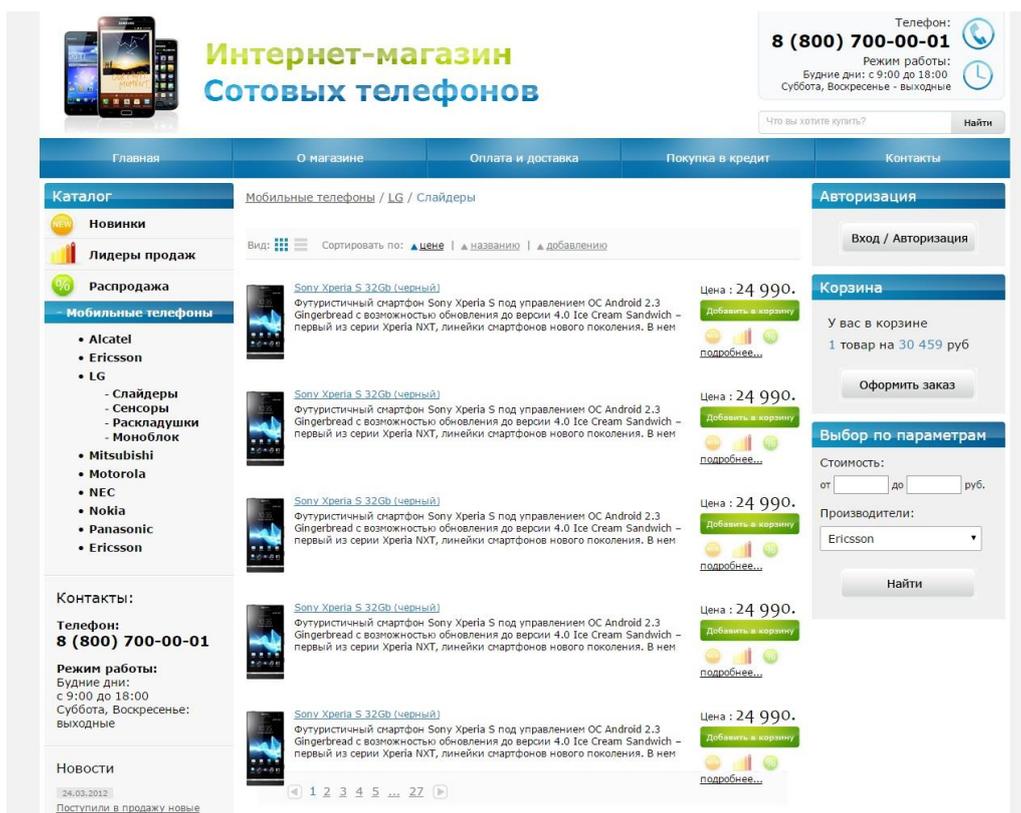


Рисунок 5.17 – Страница линейного просмотра телефонов


```

/*      Katalog-table      */
.product-table{
  float:left; width:33%; text-align:center; margin:0 0 60px 0;
}
  .product-table h2{
    font:12px Verdana, Geneva, sans-serif; color:#2b8aba; margin:0 0 10px 0;
  }
    .product-table h2 a{
      color:#2b8aba;
    }
    .product-table h2 a:hover{
      text-decoration:none;
    }
  .product-table p{
    font:12px Verdana, Geneva, sans-serif; margin:0 0 10px;
  }
    .product-table p span{
      font:24px Candara;
    }
  .cat-table-more{
    margin:5px 0 0 0 !important;
  }
    .cat-table-more a{
      color:#888;
    }
    .cat-table-more a:hover{
      text-decoration:none;
    }

```

Рисунок 5.19 – Стиль для табличного просмотра товаров

```

    .cat-table-more a:hover{
      text-decoration:none;
    }
  .product-table-img-main{
    position:relative;
  }
  .product-table-img{
    display:table-cell; height:140px; position:relative; vertical-align:middle; width:1%;
  }
  .product-table-img div{
    position:absolute; top:0; right:0; margin:0 20px 0 0;
  }
  .product-table-img div img{
    display:block;
  }

```

Рисунок 5.20 – Стиль для табличного просмотра товаров (продолжение)

Сохраните изменения, посмотрите результат в браузере (рисунок 5.21).



Рисунок 5.21 – Табличный просмотр одного телефона

Теперь нам необходимо еще 8-11 раз добавить код описания одного телефона на страницу. Для того, чтобы увеличить число телефонов на странице, нам необходимо открыть файл **index-table.html**, найти контейнер **<div class="product-table">** и полностью скопировать содержимое данного контейнера в буфер обмена, после чего 8-11 раз подряд вставить скопированный фрагмент ниже контейнера-оригинала.

Сохраните файл, посмотрите результат в браузере.

Посмотрим на сайт целиком (рисунок 5.22).

Сравните полученный результат с макетом, убедитесь в правильности верстки.

Приступаем к созданию текстовых страниц сайта. В качестве текстовой страницы могут выступать страницы новостей или информеров, страницы «О магазине», «Оплата и доставка» и т.п. Откройте в просмотрщике графических файлов (в любом т.н. «вьюере») файл «Текстовая страница.jpg» (она содержится в папке «maket»). Внешний вид данного файла во вьюере представлена на рисунке 5.23.



Рисунок 5.23 – Вид текстовой страницы (макет)

Если посмотреть на страницу (рисунок 5.23), то можно увидеть, что внятного текста в ней нет. Наша задача – создать для такой страницы некий «шаблон» со стилями, для которой разное содержимое будет оформляться одинаково.

В качестве основы для текстового файла можно взять любой из файлов, где в зоне основного контента есть «хлебные крошки», например, файл `index-table.html`. Откройте в блокноте файл **`index-table.html`** и сохраните как файл с именем **`index-txt.html`**.

Теперь нужно удалить лишнее содержимое, а именно полностью контейнеры `<div class="vid-sort">`, `<div class="product-table">` (их много, от 6 до 12, кто как накопировал) и `<div class="pager">`.

В результате должен остаться код, представленный на рисунке 5.24. Удалите также две строки из кода на рисунке 5.24 – иначе стили каталогов будут наследоваться в нашей странице.

```
<div id="contentwrapper">
  <div id="content">
    <div class="catalog-index">
      <div class="kroshka">
        <a href="#">Мобильные телефоны</a> / <a href="#">LG</a> / <span>Слайдеры</span>
      </div>
    </div>
  </div>
</div>

<div id="left-bar">
  <div class="left-bar-cont">
```

Рисунок 5.24 – Код текстовой страницы

Можно проверить правильность удаления лишнего кода (рисунок 5.25) - в центральной части страницы должны остаться только «хлебные крошки».

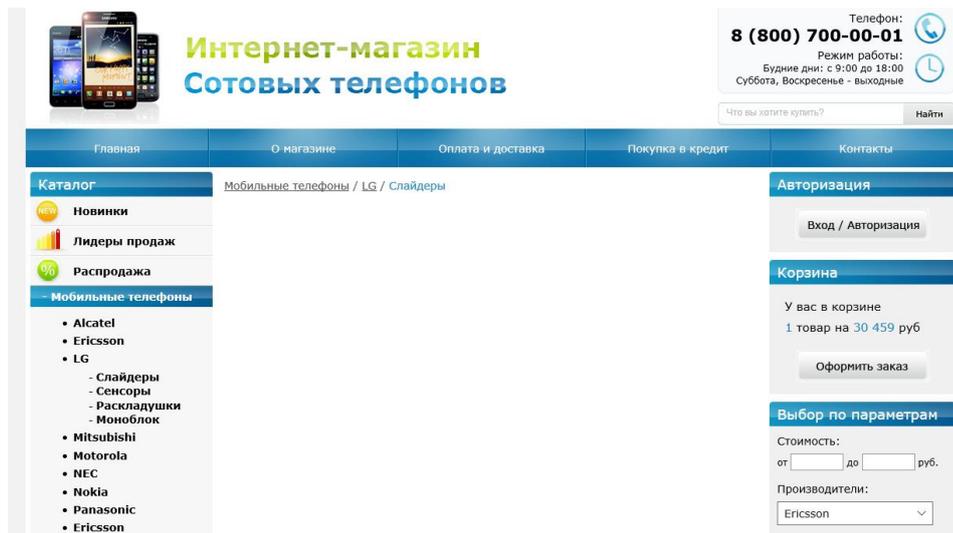


Рисунок 5.25 – Вид текстовой страницы (заготовка)

В качестве текстовой основы для страницы мы будем использовать произвольный текст, главная задача которого – отразить все представленные в макете элементы страницы с требуемыми стилями.

К работе прилагается текстовый файл с именем «Текст для л.р. 5.txt». Откройте в блокноте данный файл, скопируйте содержимое и вставьте его в файл **index-txt.html**. Оформите разметку текста страницы в соответствии с рисунками 5.26 - 5.28.

```

<div class="kroshka">
  <a href="#">Мобильные телефоны</a> / <a href="#">LG</a> / <span>Слайдеры</span>
</div>

<div class="content-txt">
<h1>Баланс в веб-дизайне.</h1>
<p>В веб-дизайне очень важно, чтобы пользователь не чувствовал себя не комфортно – это как смотреть на раковину, полную грязной посуды. Если ваш дизайн визуальне неуравновешен, то сайт будет выглядеть неорганизованным, а это может заставить пользователей чувствовать себя неудобно. Напротив, ваш <a href="#">дизайн должен быть</a> привлекательным и позволять посетителям расслабиться, оглядеться вокруг, и самый лучший способ достичь такого эффекта – визуальный баланс. Визуальный баланс – это, в основном, баланс элементов дизайна, их расположение – словно жонглирование элементами вашего дизайна.</p>
<p>Когда мы представляем себе баланс, то проще всего его сравнить с понятием веса. Физический вес и вес элементов дизайна на самом деле очень похожи – физический объект может быть маленьким, но очень тяжелым, независимо от массы, так же, как визуальный элемент дизайна может быть очень небольшим, но очень привлекающим внимание. Тяжесть физического объекта можно сравнить с визуальной яркостью элементов дизайна, и каждый элемент взаимодействует с другими элементами дизайна так же, как и физические объекты.</p>
<h2>Почему сбалансированные сайты выглядят хорошо</h2>
<p>Мы подсознательно любим баланс. Любовь к устойчивым конструкциям заложена в нас на подсознательном уровне, дисбаланс, напротив, вызывает чувство отторжения. Таким образом, визуальный баланс – залог успешного дизайна. Это та дополнительная «кислора», которую одни сайты имеют, а другие – нет. В качестве дополнительного преимущества, сбалансированный дизайн хорошо сочетается с функциональностью.</p>

```

Рисунок 5.26 – Оформление текста страницы (начало)

Обратите внимание – мы добавили контейнер с новым классом `<div class="content-txt">`. Далее мы оформим стиливое оформление текстовой страницы нашего сайта.

```
<h2>Почему сбалансированные сайты выглядят хорошо</h2>
<p>Мы подсознательно любим баланс. Любовь к устойчивым конструкциям заложена в нас на подсознательном уровне, дисбаланс, напротив, вызывает чувство отторжения. Таким образом, визуальный баланс – залог успешного дизайна. Это та дополнительная «искра», которую одни сайты имеют, а другие – нет. В качестве дополнительного преимущества, сбалансированный дизайн хорошо сочетается с функциональностью.</p>
<p>Есть несколько различных факторов, которые влияют на работу с элементами дизайна. Как вы балансируете двумя или более элементами по отношению друг к другу, и как вы представляете визуальные свойства элементов – это тип баланса и свойства баланса.</p>
<h3>Типы баланса</h3>
<p>Есть несколько различных методов создания баланса. Вы можете использовать эти различные методы, чтобы организовать и упорядочить элементы, чтобы создать хорошо сбалансированное впечатление, что позволит пользователям чувствовать себя комфортно. Рассмотрим пять основных способов достижения баланса</p>
<ul>
  <li>Горизонтальный баланс</li>
  <li>Вертикальный Баланс</li>
  <li><a href="#">Радиальный Баланс</a></li>
  <li>Симметричный баланс</li>
  <li>Асимметричный баланс</li>
</ul>
<h4>Горизонтальный и вертикальный баланс</h4>
<p>Отличный способ визуально упорядочить элементы – сбалансировать их по осям – горизонтальной или вертикальной. Горизонтальный баланс – вы располагаете элементы слева и справа от друга – как на детских качелях. Вертикальный баланс – вы развешиваете элементы сверху вниз по центральной оси.</p>
<h5>Радиальный баланс</h5>
<p>Вместо того, чтобы сбалансировать композиционные элементы по центральным осям, вы можете также сбалансировать элементы вокруг центра – радиальный баланс. В принципе, вы можете разместить элементы вокруг центральной оси, или же наподобие солнечных лучей.</p>
```

Рисунок 5.27 – Оформление текста страницы (продолжение)

```

<h5>Радиальный баланс</h5>
<p>Вместо того, чтобы сбалансировать композиционные элементы по центральным осям, вы можете также сбалансировать элементы вокруг центра – радиальный баланс. В принципе, вы можете разместить элементы вокруг центральной оси, или же наподобие солнечных лучей.</p>
<h6>Симметричный и асимметричный баланс</h6>
<p>Еще один вид баланса, который, как правило, сочетается с горизонтальным, вертикальным и радиальным балансом – симметричный и асимметричный баланс. Симметричный баланс – когда обе стороны композиции являются зеркальным отображением друг друга – как будто складывается лист бумаги пополам. Симметрия – это очень формальный, строгий баланс, он комфортный для глаз, и широко используется в дизайне зданий или макетах. Он также становится все более популярным, потому что его легче воспроизвести в любом размере.</p>
<div class="clr"></div>
</div> <!-- Закрывающий тег дива content-txt -->
</div> <!-- Закрывающий тег дива content -->
</div> <!-- Закрывающий тег дива contentwrapper -->
<div id="left-bar">
<div class="left-bar-cont">

```

Рисунок 5.28 – Оформление текста страницы (окончание)

Сохраните изменения в блокноте, посмотрите результат в браузере.



Рисунок 5.29 – Вид текстовой страницы (без стилей)

Страница оформлена в соответствии со своими (браузера) стилями. Приступаем к стилевому оформлению страницы. Откройте файл **style.css** и в конце существующего добавьте в него следующий код (рисунок 5.30).

```
/*          CONTENT-TXT          */
.content-txt{
  overflow:hidden; font:12px Verdana, Geneva, sans-serif; color:#000; margin:0 15px;
}
.content-txt h1{
  font-size:24px; color:#2b8aba; font-weight:normal; margin:0 0 15px 0;
}
.content-txt h2{
  font-size:20px; color:#000; font-weight:normal; margin:20px 0 15px 0;
}
.content-txt h3{
  font-size:18px; color:#000; font-weight:normal; margin:20px 0 15px 0;
}
.content-txt h4{
  font-size:14px; color:#000; font-weight:bold; margin:20px 0 15px 0;
}
.content-txt h5{
  font-size:12px; color:#000; font-weight:bold; margin:20px 0 15px 0;
}
.content-txt h6{
  font-size:10px; color:#000; font-weight:bold; margin:20px 0 15px 0;
}
.content-txt p{
  margin:0 0 10px 0;
}
.content-txt a{
  color:#2b8aba;
}
.content-txt a:hover{
  text-decoration:none;
}
.content-txt ul{
  list-style:disc; margin:20px 0 15px 30px;
}
.content-txt ul li{
  line-height:1.7;
}
```

Рисунок 5.30 – Стили для текстовой страницы

Сохраните изменения, посмотрите результат в браузере (рисунок 5.31).

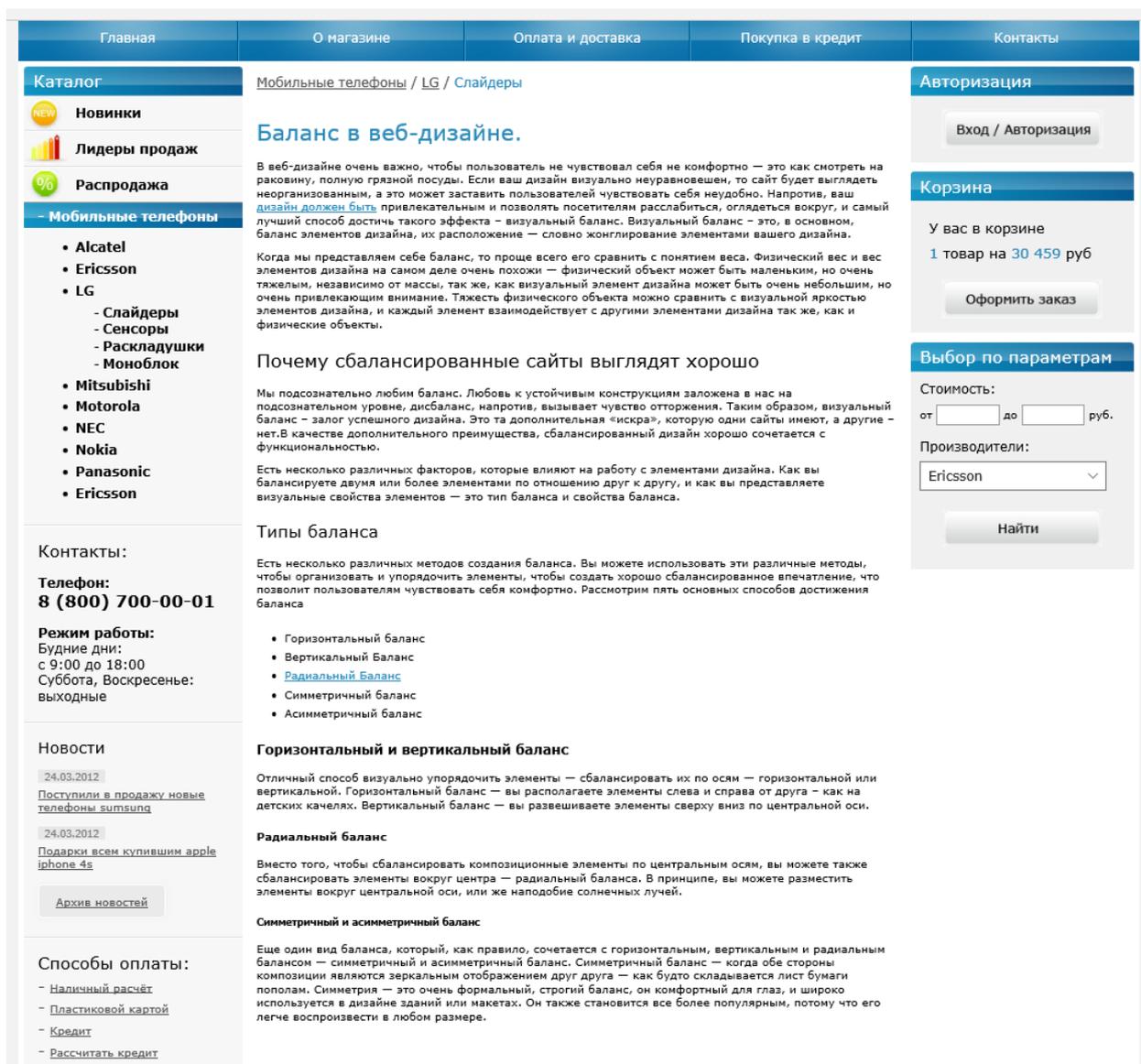


Рисунок 5.31 – Внешний вид текстовой страницы

Как видно из рисунка, оформление текстовой страницы полностью совпадает с представленным в макете (рисунок 5.23).

Самостоятельно. Разобраться с приведенным кодом, ответить на любые вопросы преподавателя по оформлению блока основного контента.

Лабораторная работа 6

Карточка товара

Переходим к созданию страницы детального просмотра товара, также называемой «карточкой товара». На рисунке 6.1 приведён внешний вид карточки товара. Вначале идёт название телефона, затем слайдер с большими и малыми картинками телефонов, далее краткое описание телефона, цена, кнопка «Добавить в корзину» и раздел с техническими характеристиками телефонов.

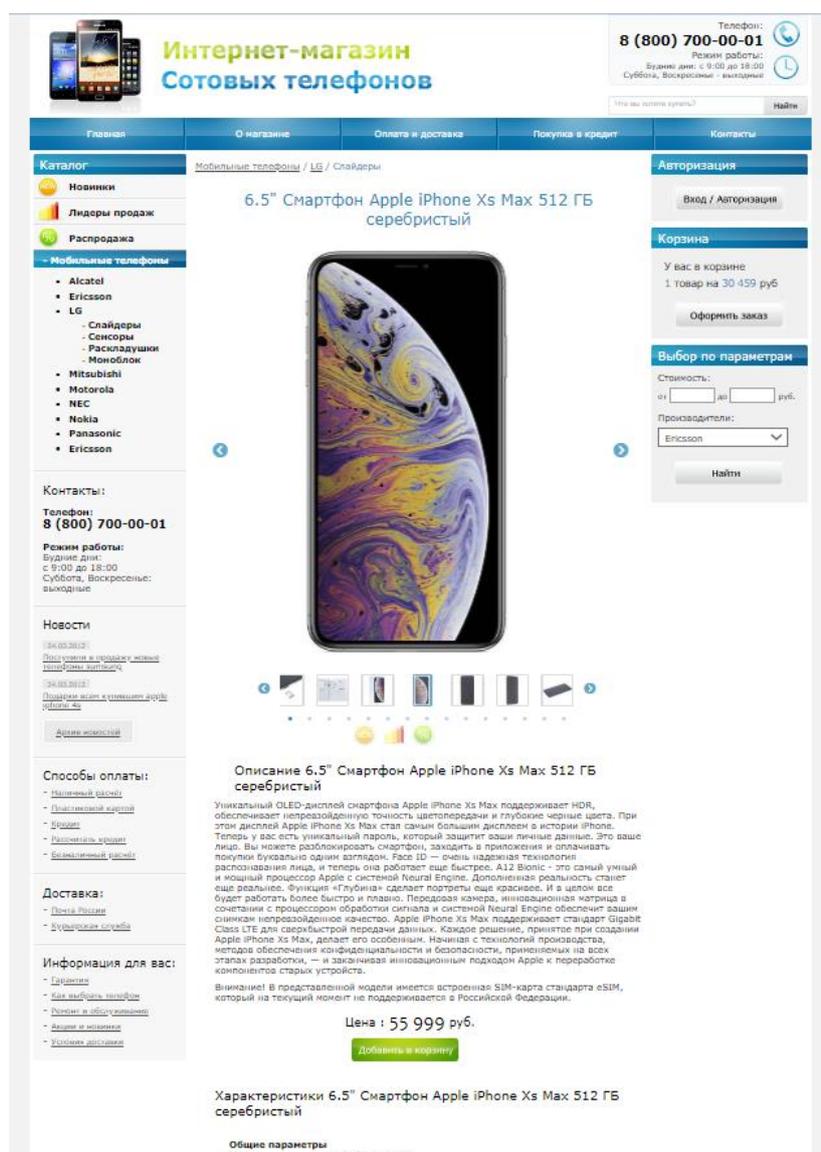


Рисунок 6.1 – Макет страницы детального просмотра товара

Приступаем к написанию кода страницы карточки товара. В качестве заготовки используем файл текстовой страницы из предыдущей лабораторной работы. Откройте в блокноте файл **index-txt.html** и сохраните его под именем **index-detail.html**.

Удалите содержимое контейнера `<div class="content-txt">` (за исключением очищающего контейнера `<div class="clr"></div>`).

В результате код страницы детального просмотра товара должен быть следующим (рисунок 6.2). Обратите внимание, что контейнер `<div class="content-txt">` был переименован в контейнер `<div class="catalog-detail">`.

```
<div id="contentwrapper">
  <div id="content">
    <div class="kroshka">
      <a href="#">Мобильные телефоны</a> / <a href="#">
    </div>
    <div class="catalog-detail">
    </div>
    <div class="clr"></div>
  </div>
</div>

<div id="left-bar">
  <div class="left-bar-cont">
```

Рисунок 6.2 – Код файла **index-detail.html** перед наполнением содержимым

Для работы Вам представлена папка «*Для лабы 6*». Внутри этой папки есть 5 файлов («6.5 Смартфон Apple iPhone Xs Max 512 ГБ серебристый.jpg», «6.5 Смартфон Apple iPhone Xs Max 512 ГБ серебристый.html», «код скрипта слайдера.txt», «стиль для слайдера в css.txt» и «цена.txt») и 3 папки («В корень сайта», «В папку js» и «Слайдер»).

Работать со слайдером мы будем в конце лабораторной работы, поэтому на время оставим папки и начнём работать с файлами. Откройте в блокноте файл 6.5 Смартфон Apple iPhone Xs Max 512 ГБ серебристый.html. Код этого файла скачан со страницы подробного просмотра товара (карточки товара) с

сайта **dns-shop.ru** для данного телефона. Скопируйте название модели телефона из данного файла и вставьте в файл **index-detail.html**. Почему не взять название телефона из имени файла? Дело в том, что в имени файла нельзя писать символы дюймов (""), а вот в тексте внутри файла – можно. Будем копировать название файла с дюймами. Откройте файл **цена.txt**, скопируйте цену телефона и вставьте в код файла **index-detail.html** согласно рисунка 6.3. Также добавьте код

```

<div class="catalog-detail">
  <h1>6.5" Смартфон Apple iPhone Xs Max 512 ГБ серебристый</h1>
  <h2>Здесь будет слайдер!!!</h2>
  <div class="icon-detail">
    
    
    
  </div> <!-- Закрывающий тег icon-detail -->
  <div class="short-opais">
    <h3>Здесь будет краткое описание:</h3>
    <p> </p>
    <p> </p>
    <p class="price-detail">Цена : <span>55 999 </span>руб.</p>
    <a href="#"></a>
  </div> <!-- Закрывающий тег short-opais -->
  <h3> </h3>
  <div class="clr"></div>
  <div class="tech-harakt">
    <h2>Здесь будут технические характеристики</h2>
  </div> <!-- Закрывающий тег tech-harakt -->
  <div class="clr"></div>
</div> <!-- Закрывающий тег catalog-detail -->

```

Рисунок 6.3 – Код файла **index-detail.html**

Внешний вид карточки товара до применения стилей и наполнения содержимым представлен на рисунке 6.4.



Рисунок 6.4– Внешний вид карточки товара

Добавьте в файл **style.css** код согласно рисунка 6.5. Код стилей пишем в самом низу файла стилей.

```
/*      CATALOG DETAIL      */
.catalog-detail{
    overflow:hidden; font:12px Verdana, Geneva, sans-serif; color:#000; margin:0 15px;
}
.catalog-detail h1{
    font:24px verdana; color:#2b8aba; margin:0 0 20px 0; text-align: center;
}
.icon-detail{
    margin:0 0 0 240px;
}
.price-detail {
    font:18px Verdana; margin:0 0 10px 230px;
}
.price-detail span{
    font-size:30px!important; font-family:Candara;
}
```

Рисунок 6.5 – Код файла **style.css**

Сохраните изменения, посмотрите результат в браузере (рисунок 6.6).



Рисунок 6.6 – Страница детального просмотра телефона

Стили оформлены почти верно. Теперь нужно добавить тексты короткого описания телефона и далее его технических характеристик. Скопируйте из файла **6.5 Смартфон Apple iPhone Xs Max 512 ГБ серебристый.html** див (**div**) с классом `<div class="price_item_description" itemprop="description">` целиком, далее откройте файл **index-detail.html**, очистите контейнер `<div class="short-opais">` от 3-х строчек начального текста (удалите строки сразу после объявления контейнера `<div class="short-opais">`) и вставьте в данное

место скопированный в буфер текст. На рисунке 6.7 показан результат вставки кода короткого описания телефона в файл **index-detail.html**.

```
<div class="short-opais">
  <div class="price_item_description" itemprop="description">
    <h2>Описание 6.5" Смартфон Apple iPhone Xs Max 512 GB серебристый</h2>
    <p> Уникальный OLED-дисплей смартфона Apple iPhone Xs Max поддерживает HDR, об
непревзойденную точность цветопередачи и глубокие черные цвета. При этом дисплей Apple
iPhone Xs Max стал самым большим дисплеем в истории iPhone. Теперь у вас есть уникальный
пароль, который защитит ваши личные данные. Это ваше лицо. Вы можете разблокировать смартфон,
заходить в приложения и оплачивать покупки буквально одним взглядом. Face ID – очень надежная
технология распознавания лица, и теперь она работает еще быстрее.
A12 Bionic – это самый умный и мощный процессор Apple с системой Neural Engine. Дополненная
реальность станет еще реальнее. Функция «Глубина» сделает портреты еще красивее. И в целом
все будет работать более быстро и плавно. Передовая камера, инновационная матрица в сочетании
с процессором обработки сигнала и системой Neural Engine обеспечит вашим снимкам непревзойденное к
Apple iPhone Xs Max поддерживает стандарт Gigabit Class LTE для сверхбыстрой передачи данных.
Каждое решение, принятое при создании Apple iPhone Xs Max, делает его особенным. Начиная с
технологий производства, методов обеспечения конфиденциальности и безопасности, применяемых
на всех этапах разработки, – и заканчивая инновационным подходом Apple к переработке
компонентов старых устройств.</p>
    <p>Внимание! В представленной модели имеется встроенная SIM-карта стандарта eS
момент не поддерживается в Российской Федерации.</p>
  </div>
<p class="price-detail">Цена : <span>55 999 </span>руб.</p>
```

Рисунок 6.7 – Код короткого описания телефона в файле **index-detail.html**

Откуда взялись дивы `<div class="price_item_description" itemprop="description">`? Они были скопированы со страницы карточки товара сайта **dns-shop.ru**. Так как было накопировано достаточно много описаний различных товаров (телефонов), то было принято решение не менять название этих дивов – так меньше работы!

Вообще менеджеры в магазине получают партию товаров вместе с файлом (или несколькими файлами – по категориям товаров), содержащими описание товаров. Обычно это файлы в формате **.xml** (eXtensible Markup Language - расширяемый язык разметки, предназначенный для хранения и передачи данных) или даже **.xls** (таблицы Excel). Кроме того, для интеграции сайтов и баз данных применяется формат файлов **csv**. **CSV** (от англ. Comma-Separated Values - значения, разделённые запятыми) - текстовый формат, предназначенный для представления табличных данных. Строка таблицы соответствует строке текста, которая содержит одно или несколько полей, разделенных запятыми. Именно в формате **csv** осуществляется обмен данными между 1С и сайтом на 1С-Битрикс: обмен товарами и обмен справочниками.

Формат CSV стандартизирован не полностью. Идея использовать запятые для разделения полей очевидна, но при таком подходе возникают проблемы, если исходные табличные данные содержат запятые или переводы строк. Возможным решением проблемы запятых и переносов строк является заключение данных в кавычки, однако исходные данные также могут содержать кавычки. Помимо этого, термином «CSV» могут обозначаться похожие форматы, в которых разделителем является символ табуляции (TSV) или точка с запятой. Многие приложения, которые работают с форматом CSV, позволяют выбирать символ разделителя и символ кавычек.

Для работы с файлами, содержащих описание товаров, существует большое разнообразие различных скриптов, написанных на языках javascript или php и оптимизированных под конкретный сайт (точнее, под базу данных сайта).

Итак, мы написали код блока краткого описания телефона. Посмотрите результат в браузере (рисунок 6.8).



Рисунок 6.8 – Внешний вид карточки товара с блоком описания телефона

Как видно из рисунка, все основные элементы блока описания телефона присутствуют, однако стилевое оформление не соответствует модели. Правда, стили некоторых элементов оформлены почти правильно – но это только потому, что они «приняли» стили ранее описанных элементов. Это неправильно. Нужно описать стили именно для данных элементов блока.

Приступаем к оформлению стилей для данного блока короткого описания товара в файле **style.css** (рисунок 6.9).

```
558 .price-detail span{
559     font-size:30px!important; font-family:Candara;
560 }
561 .short-opais{
562     margin:0 0 40px 10px; font-family:Verdana;
563 }
564 .short-opais .price_item_description h2{
565     font-size:20px; font-weight: normal; text-align:center;
566 }
567 .short-opais .price_item_description p{
568     font-size:14px; margin:10px 20px 10px 10px; text-align:justify; text-indent: 20px;
569 }
570 .short-opais .price-detail{ float: center;
571     font-size:18px!important;
572     margin:0 0 10px 220px;
573 }
574 .short-opais .price-detail span{ float: center;
575     font:26px Candara!important;
576 }
577 .short-opais a{
578     margin:0 0 40px 230px;
579 }
```

Рисунок 6.9 – Стилль для блока короткого описания товара в файле **style.css**

Первая строка стилей назначает внешние отступы для блока и назначает основной шрифт. Далее назначаем стили для заголовка и абзацев блока короткого описания телефона. Обратите внимание на некоторые улучшения стилевого оформления абзацев даже по сравнению с макетом. Первое – мы выровняли текст по ширине страницы (параметр **justify** у атрибута **text-align**). Второе – сделали абзацный отступ размером 20px у каждого абзаца (атрибутом **text-indent: 20px;**). Кроме того, мы использовали механизм повышения приоритета для некоторых элементов, имеющих сложную структуру вложения селекторов (с помощью команды **!important**). Прочитать про команду

!important можно в статье по следующей ссылке:
<https://habr.com/ru/post/484556/>.

Однако использование этого механизма не является рекомендуемым в стандартах **css**. Поэтому после применения стилей к блоку описания мы очищаем свойства стилей с помощью слоя `<div class="clr"></div>`.

Сохраните изменения, посмотрите результат в браузере (рисунок 6.10).

The screenshot shows a mobile phone catalog page. On the left is a sidebar with a 'Каталог' (Catalog) menu, including 'Новинки' (New), 'Лидеры продаж' (Bestsellers), 'Распродажа' (Sale), and 'Мобильные телефоны' (Mobile phones) with sub-categories like Alcatel, Ericsson, LG, etc. The main content area displays a product listing for a '6.5" Smartphone Apple iPhone Xs Max 512 GB silver'. It includes a 'Здесь будет слайдер!!!' (Slider will be here!!!) placeholder, a detailed description of the phone's features like the OLED display and Face ID, and a price of 55,999 rubles. A 'Добавить в корзину' (Add to cart) button is visible. On the right, there are sections for 'Авторизация' (Authorization) with a login button, 'Корзина' (Cart) showing 1 item for 30,459 rubles, and 'Выбор по параметрам' (Filter by parameters) with a search button.

Рисунок 6.10 – Внешний вид блока короткого описания товара со стиливым оформлением

Как видно из рисунка, блок описания телефона оформлен даже лучше, чем это было в макете страницы.

Приступаем к оформлению блока технических характеристик телефона. Вначале в файле **index-denail.html** удалите строку с текстом «Здесь будут технические характеристики» вместе с тегами. Далее скопируйте из файла **6.5 Смартфон Apple iPhone Xs Max 512 ГБ серебристый.html** код, начиная с дива (**div**) с классом `<div class="price_item_description" itemprop="description">` до самого конца файла, и вставьте в файл **index-denail.html** в очищенное ранее место скопированный в буфер текст (рисунок 6.11).

```

        </div> <!-- Закрывающий тег short-opais -->
        <div class="clr"></div>

<div class="tech-harakt">
  <div class="price_item_description" itemprop="description">
    <h2>Характеристики 6.5" Смартфон Apple iPhone Xs Max 512 ГБ серебристый</h2>
  </div>
  <div class="options-group">
    <table class="table-params table-no-bordered">
      <tr><td colspan="2" class="table-part">Общие параметры</td></tr>
      <tr><td><div class="dots"><span>Модель</span></div></td>
      <td>Apple iPhone Xs MAX</td></tr>
      <tr>
      <td><div class="dots"><span>Год релиза</span></div></td>
      <td>2018</td>
      </tr>
      <tr>
      <td colspan="2" class="table-part">Внешний вид</td>
      </tr>
      <tr>
      <td><div class="dots"><span>Цвет задней панели</span></div></td>
      <td>серебристый</td>
      </tr>
      <tr>
      <td><div class="dots"><span>Цвет передней панели</span></div></td>
      <td>черный</td>
    </table>
  </div>
</div>

```

Рисунок 6.11 – Код блока «Технические характеристики» в файле **index-detail.html**

Далее добавим код в файл стилей **style.css**, описывающий теги технических характеристик товара (рисунок 6.12).

```

577 .short-opais a{
578     margin:0 0 40px 230px;
579 }
580 .tech-harakt{
581     margin:30px 0 20px 20px; font-family:Verdana;
582 }
583 .tech-harakt .price_item_description h2{
584     font-size:20px; font-weight: normal; text-align:center;
585 }
586 .tech-harakt .table-part{
587     font-size:14px; font-weight:bold; padding:10px 0 0 20px;
588 }
589 .tech-harakt .dots span{
590     font-size:12px; font-weight:normal; padding:0;
591 }
592 .tech-harakt td{
593     font-size:12px; font-weight:normal; padding:0 20px;
594 }

```

Рисунок 6.12 – Стиль для блока «Технические характеристики» товара в файле **style.css**

В принципе код файла стилей стандартен, он описывает размеры шрифтов, внешние (**margin**) и внутренние (**padding**) отступы между элементами кода, а также толщину (**normal** или **bold**) текста.

Сохраните изменения в коде, посмотрите результат (рисунок 6.13).

Характеристики 6.5" Смартфон Apple iPhone Xs Max 512 ГБ серебристый

Общие параметры

Модель Apple iPhone Xs MAX
Год релиза 2018

Внешний вид

Цвет задней панели серебристый
Цвет передней панели черный
Цвет граней серебристый
Цвет, заявленный производителем серебристый

Мобильная связь

Поддержка сетей 2G GSM 850, GSM 1800, GSM 1900, GSM 900
Поддержка сетей 3G UMTS 1700/2100, UMTS 1900, UMTS 850, UMTS 900, UMTS 2100
Поддержка сетей 4G (LTE) есть
диапазонов частот LTE LTE 1900 (B25), LTE 1500 (B32), LTE 1700 (B4), LTE 1700/2100 (B66), LTE 700 (B29), LTE 800 (B26), LTE 800 (B18), LTE 800 (B19), LTE 700 (B13), LTE 2300 (B30), LTE 1800 (B3), LTE 850 (B5), LTE 2100 (B1), LTE 800 (B20), LTE 700 (B12), LTE 2600 (B7), LTE 900 (B8), LTE 700 (B17), LTE 1900 (B2)
Формат SIM-карт Nano-SIM (12.3x8.8x0.67 мм)
Количество SIM-карт 1 SIM
Количество SIM-карт 1 SIM

Рисунок 6.13 – Внешний блока «Характеристики товара»

Таким образом, оформление текстовой части страницы описания товара выполнено в полном соответствии с заданием. Осталось только оформить слайдер.

Мы используем современный слайдер – **slick**. Данный слайдер имеет множество настроек, которые подробно описано на сайте слайдера **slick** (адрес слайдера: <http://kenwheeler.github.io/slick/>). Язык сайта – только английский. Однако есть большое множество мануалов по использованию данного слайдера, включая видео на русском языке.

При написании лабораторной работы были использованы три видеоурока по слайдеру **slick** одного автора (Master-CSS):

1. <https://www.youtube.com/watch?reload=9&v=MIWkcBqWISk>
2. <https://www.youtube.com/watch?v=-AFVIf8Uzxo>
3. <https://www.youtube.com/watch?v=WNyRZytNx14>

Первое видео описывает собственно слайдер **slick**, его возможности, работу с сайтом слайдера, первичную установку слайдера.

Второе видео – настройки java-script-а слайдера и немного по настройке стилей.

Третье видео – настройка внешнего вида слайдера при помощи **css**.

Обращаю Ваше внимание на то, что автор видеороликов использует в своей работе свободный текстовый редактор для веб-разработчиков **Brackets**. Brackets ориентирован на работу с HTML, CSS и JavaScript. Эти же технологии лежат в основе самого редактора, что обеспечивает его кроссплатформенность т.е. совместимость с операционными системами Mac, Windows и Linux. Brackets создан и развивается Adobe Systems под лицензией MIT License и поддерживается на GitHub.

Автор использует возможности текстового редактора Brackets по автоматизации создания HTML-кода страницы (когда нужно создать несколько почти повторяющихся строк кода). Кстати, автор (Master-CSS) создал курсы по работе с Brackets. Кто заинтересовался, вот ссылка на ютубе: <https://www.youtube.com/playlist?list=PLPpaecEYRC8bGw3oj3ecnPjSe09r4bPyo>

Кроме того, Вы можете посмотреть видео другого автора про более расширенное использование слайдера **Slick** с эффектом увеличительного стекла: <https://www.youtube.com/watch?reload=9&v=jipN0zK13oQ> - Slick slider + zoom эффект.

Мы не можем пользоваться слайдером с зум-эффектом, потому что у нас слишком много картинок (от 13 до 16). Обычно слайдеры с зум-эффектом (часто с зум-эффектом в отдельном окне) имеют 3-5 миниатюр с вертикальным расположением (как, например, на сайте ali-express). Нам же эти миниатюры просто физически невозможно расположить вертикально, да и горизонтально они займут всё свободное пространство. Поэтому мы будем использовать самый простой слайдер «с точками» (дотами от слова **dot** – точка). Подобные простые слайдеры есть почти на всех сайтах.

Таким образом, мы будем использовать «чистый» слайдер **slick**, без дополнительных слайдеров (позволяющих создать эффекты просмотра выбранной картинки в ограниченном окне, в новом полноэкранным окне или с эффектом «лупы»).

На сайте <http://kenwheeler.github.io/slick/> мы выбираем внешний вид слайдера с первичным кодом java-скрипта (рисунок 6.14). Кстати, слайдер интерактивен, можно «пощелкать» по его элементам.

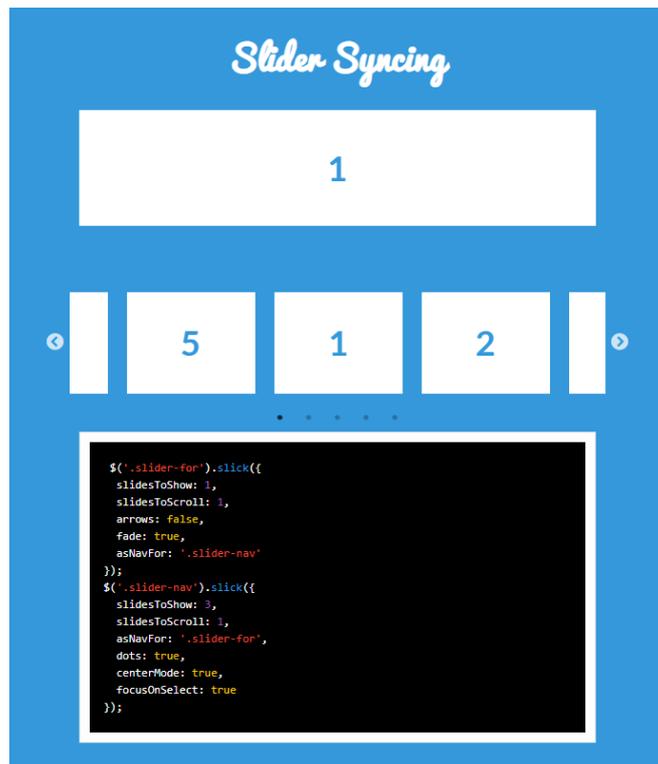


Рисунок 6.14 – Внешний вид и Java-script выбранного слайдера

Краткий анализ java-скрипта слайдера показывает, что у нас есть две галереи товаров: первая – с классом **.slider-for**, второй – с классом **.slider-nav**. Во втором классе приведены «точки» (**dots:true**), присутствующие на большинстве стандартных слайдеров.

Для работы слайдера нам необходимо прописать ссылки на файлы стилей данного слайдера. Сами файлы стилей предложены Вам в папке «Для лабы 6 \ Вставить на сайт\». Вставьте папку **slick** из допматериалов лабораторной работы в корень сайта **ishop** (рядом с файлами **style.css** и **index-detail.html**). Кроме того, вставьте скрипты из предложенной к уроку папке **js** в папку **js** на сайте **ishop**.

Далее откройте файл **index-detail.html** и добавьте в шапку сайта код подключения файлов стилей в красной рамке, представленный на рисунке 6.15.

```

<link rel="stylesheet" type="text/css" href="style.css" />
<link rel="stylesheet" type="text/css" href="slick/slick.css" />
<link rel="stylesheet" type="text/css" href="slick/slick-theme.css" />
<title>Карточка товара</title>
</head>
<body>

```

Рисунок 6.15 – Применение стилей слайдера **slick** в файле **index-detail.html**

Далее нам необходимо скопировать картинки на наш сайт. Откройте предоставленную Вам к занятию папку «Для лабы 6», далее папку «Слайдер». Внутри этой папки расположены 2 папки: **photos** с 15 большими картинками для слайдера и папку **thumbnails** с 15 миниатюрами. Скопируйте содержимое этих папок в аналогичные папки на сайте. Названия больших картинок и миниатюр совпадают, кроме того, они имеют одинаковый префикс в имени: **41_** Этот префикс назначается на стороне администратора автоматически при занесении информации о новом телефоне в базу данных, номер префикса соответствует номеру телефона в базе данных.

Итак, нам теперь есть, что выводить в слайдере. Откройте файл **index-detail.html** и добавьте в него код согласно рисунку 6.16.

```
<div class="catalog-detail">
  <h1>6.5" Смартфон Apple iPhone Xs Max 512 ГБ серебристый</h1>
  <div class="slider-for">
    <div class="image-center"><a href="#"></a></div>
    <div class="image-center"><a href="#"></a></div>
  </div>
  <div class="slider-nav">
    <div><a href="#"></a></div>
    <div><a href="#"></a></div>
  </div>
  <div class="icon-detail">
```

Рисунок 6.16 – Код слайдера в файле **index-detail.html**

Как видно из рисунка, мы добавляем для контейнера <div>:

- <div class="slider-for"> - для больших картинок;
- <div class="slider-nav"> - для картинок – миниатюр.

Большие картинки и миниатюры, как видно, берутся из разных папок.

Стилевое оформление галереи фотографий можно скопировать из текстового файла «стиль для слайдера в css.txt», прилагаемого к лабораторной работе. Скопируйте код из текстового файла и вставьте скопированный блок в файл **style.css** после стиля элемента **.catalog-detail h1{** (рисунок 6.17).

```
529      /*          CATALOG DETAIL          */
530  .catalog-detail{
531      overflow:hidden; font:12px Verdana, Geneva, sans-serif; color:#000; margin:0 15px;
532  }
533  .catalog-detail h1{
534      font:24px verdana; color:#2b8aba; margin:0 0 20px 0; text-align: center;
535  }
536
537  .slider-for .image-center{
538      display:block; height:650px;
539      margin:0px auto; padding:0px auto;
540  }
541  .slider-for .slick-prev:before, .slider-for .slick-next:before{
542      font-family: 'slick'; font-size: 25px; line-height: 1;
543  }
544  .slider-nav{
545      width: 450px; height:65px; padding-top: 20px;
546      margin-left: 130px;
547  }
548  .slider-nav img{
549      padding:3px; height:100%; background:#ddd;
550      border-radius:3px; margin: 0px auto;
551  }
552  .slider-nav img:hover{
553      padding:3px; background:#2b8aba;
554      opacity: .75; margin: 0px auto;
555  }
556  .slider-nav .slick-current img{
557      padding:3px;
558      background:#2b8aba;
559      opacity: .7;
560  }
561  .slider-nav .slick-next{
562      right: -35px; top: 70%; z-index: 1;
563  }
564  .slider-nav .slick-prev{
565      left: -35px; top: 70%; z-index: 1;
566  }
567
568  .short-opais{
```

Рисунок 6.17 – Код стилей слайдера в файле **style.css**

Итак, картинки для слайдера скопированы, стили прописаны, осталось только запустить непосредственно слайдер. В материалах к занятиям найдите

папку «**slick**» и вставьте её в корень сайта. Кроме того, в папке «**Вставить на сайт**» содержится папка «**В папку js**». Скопируйте её содержимое в папку **js** Вашего сайта.

Из предоставленных к занятию материалов найдите файл «**код скрипта слайдера.txt**», откройте его в блокноте и скопируйте код целиком. Далее вставьте скопированный код в файл **index-detail.html** после закрывающего тега дива **main** (рисунок 6.18).

```
644     </div>      <!-- Закрывающий тег дива main -->
645
646     <script type="text/javascript" src="js/jquery-3.5.1.min.js"></script>
647     <script type="text/javascript" src="slick/slick.min.js"></script>
648     <script type="text/javascript">
649     $(document).ready(function(){
650     $('.slider-for').slick({
651         slidesToShow: 1,
652         slidesToScroll: 1,
653         arrows: true,
654         asNavFor: '.slider-nav'
655     });
656     $('.slider-nav').slick({
657         slidesToShow: 5,
658         slidesToScroll: 1,
659         centerMode: true,
660         arrows: true,
661         asNavFor: '.slider-for',
662         dots: true,
663         centerMode: true,
664         focusOnSelect: true
665     });
666     });
667 </script>
668
669 </body>
670 </html>
```

Рисунок 6.18 – Код скрипта слайдера в файле **index-detail.html**

Проанализируем код скрипта. Скрипт запускается стандартной функцией **\$(document).ready(function){...}**. Далее описываются свойства для класса, отображающего большие картинки - **.slider-for**. Мы указываем скрипту показывать одновременно по одной большой картинке (**slidesToShow: 1**), при щелчке по управляющей стрелке перемещаться по одной картинке вправо или влево на 1 позицию (**slidesToScroll: 1**), отображать управляющие (влево-вправо) стрелки (**arrows: true**) и даётся ссылка на второй, связанный с текущим, массив картинок-миниатюр командой **asNavFor: '.slider-nav'**. Для второй части слайдера – аналогично, за исключением того, что миниатюр одновременно отображается 5 картинок (**slidesToScroll: 1**) и ниже, под

картинками показываються точки (**dots: true**), выровненные по центру и с показом фокуса (точки активной картинки).

Сразу же опишем стили для слайдера. Откройте файл **стиль для слайдера в css.txt**, полностью скопируйте его содержимое и вставьте в самый низ файла стилей (рисунок 6.19).

```
592 .tech-harakt td{
593     font-size:12px; font-weight:normal; padding:0 20px;
594 }
595
596 .slider-for{width:715px; padding:0; height:650px;}
597 .slider-for .image-center{ display:block; height:650px;
598     margin:0px auto; padding:0px auto;}
599 .slider-for .slick-prev:before, .slider-for .slick-next:before{
600     font-family: 'slick'; font-size: 25px; line-height: 1;}
601
602 .slider-nav{ width: 450px; height:65px; padding-top: 20px;
603     margin-left: 130px; }
604 .slider-nav img{ padding:3px; height:100%; background:#ddd;
605     border-radius:3px; margin: 0px auto; }
606 .slider-nav img:hover{ padding:3px; background:#2b8aba;
607     opacity: .75; margin: 0px auto; }
608 .slider-nav .slick-current img{ padding:3px;
609     background:#2b8aba; opacity: .7;}
610 .slider-nav .slick-next{ right: -35px; top: 70%; z-index: 1;}
611 .slider-nav .slick-prev{ left: -35px; top: 70%; z-index: 1;}
612
```

Рисунок 6.19 – Стиль для слайдера в файле **style.css**

Может возникнуть вопрос – откуда взялись новые селекторы классов (**.slider-for** и др.), ведь на странице файла **index-detail.html**, с которой мы работаем, элементов с такими классами мы не писали? Ответ прост – слайдер сам создал новые элементы при запуске карусели рисунков. В оформлении стилей данных элементов использовались видеоуроки, ссылки на которые давались ранее в данной лабораторной работе. В первую очередь были заданы размеры картинок (ширина и высота больших картинок и миниатюр), а также внешние и внутренние отступы (маргины и паддинги) элементов слайдера. Интересным представляется способ задания рамок вокруг миниатюр. Для этого был задан цвет подложки картинок миниатюр с классом **.slider-nav img** светло-серого цвета (**#ddd**) и изменяющего свой цвет при наведении курсора (**.slider-nav img:hover**) на светло-голубой цвет (**background:#2b8aba**). Кроме того, задан светло-голубой цвет подложки для текущего элемента (**.slider-nav .slick-current img**). Текущий элемент – это тот элемент (большая картинка), который находится в середине слайдера (остальные элементы скрыты от показа за границами окна слайдера). Для того, чтобы картинки миниатюр имели рамку,

заданы внутренние отступы шириной 3px (**padding:3px;**) с закруглением по краям (**border-radius:3px;**).

Сохраняем результат, обновляем страницу карточки товара. Если всё выполнено верно, то результат обновления должен соответствовать показанному на рисунке 6.20.

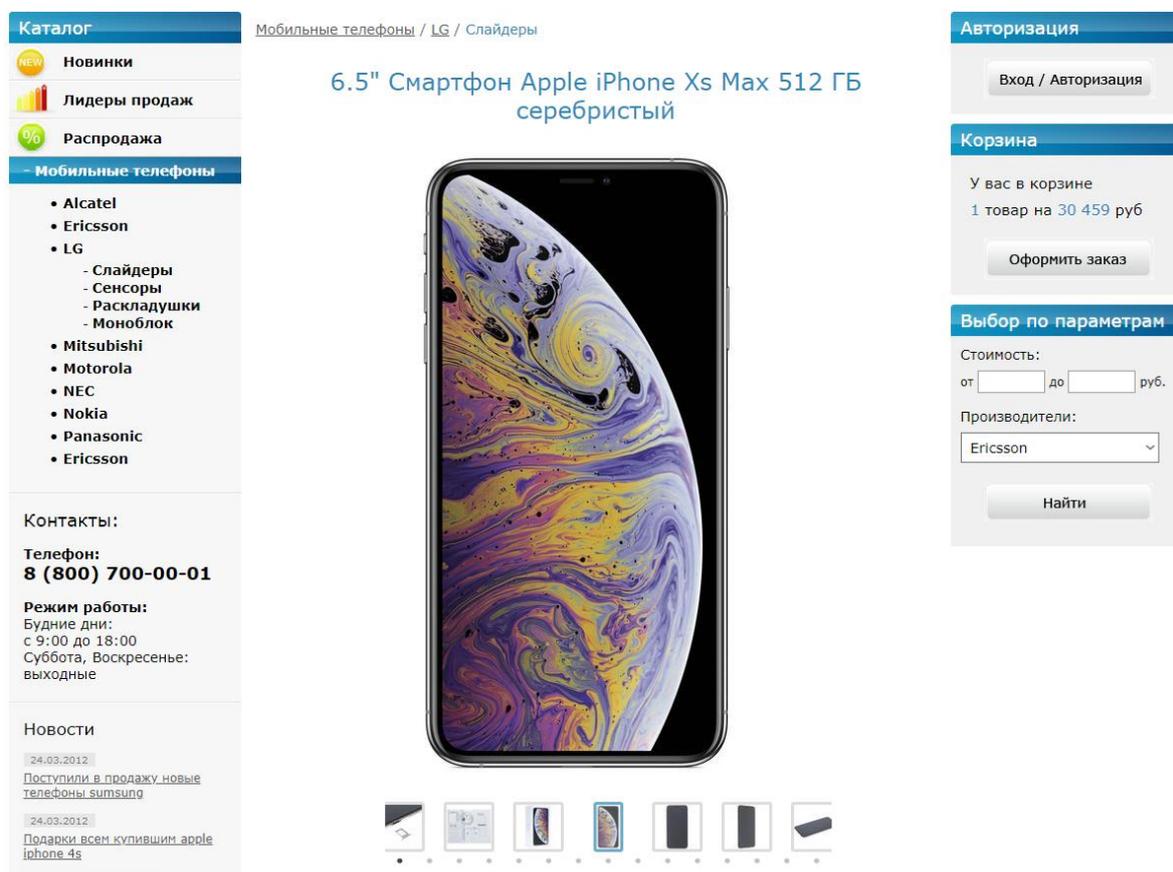


Рисунок 6.20 – Внешний вид слайдера

В целом вроде нормально, но возникает вопрос – где управляющие стрелки? Их нет. Однако, если подвести курсор к краям картинок, то можно щелканьем мыши их определить (и они работают!). Таким образом, стрелки есть, но они нам не видимы (они белые на белом фоне страницы!). Всё дело в стилях. Откройте файл **slick-theme.css**, расположенный в папке **slick** в корне Вашего сайта и исправьте белый цвет стрелок (**white**) на голубой, соответствующий палитре нашего сайта (рисунок 6.21). Обратите внимание, что в данном файле мы описываем стили, общие для больших картинок, малых картинок (миниатюр) и для точек слайдера. Кроме того, прозрачность неактивных стрелок равна **75%** (**opacity: .75;**).

```

66 .slick-prev:before,
67 .slick-next:before
68 {
69     font-family: 'slick';
70     font-size: 20px;
71     line-height: 1;
72
73     opacity: .75;
74     color: #2b8aba;
75 }

```

Рисунок 6.21 – Стили (цвет) стрелок (файл **slick-theme.css**)

Настроим отступы стрелок от границ блока слайдера. Начнём со стрелок «назад» (**.slick-prev**), как это показано на рисунке 6.22. Мы убираем знак «-» (минус) у стиля атрибута **left: -25px**. Знак минус говорит о том, что мы выходим за границы блока. Мы же сдвинем стрелки внутрь блока слайдера. Кроме того, передвинем изображение стрелок на «передний план» - атрибутом **z-index: 1**. Теперь изображения картинок никогда не будут закрывать стрелки из виду пользователя.

```

80 .slick-prev
81 {
82     left: 5px;
83     z-index: 1;
84 }
85 [dir='rtl'] .slick-prev
86 {
87     right: 25px;
88     left: auto;
89 }

```

Рисунок 6.22 – Стили стрелок «назад» - **.slick-prev** (файл **slick-theme.css**)

Аналогичные атрибуты пропишем для стилей стрелок «вперёд» (рисунок 6.23).

```

99 .slick-next
100 {
101     right: 35px;
102     z-index: 1;
103 }
104 [dir='rtl'] .slick-next
105 {
106     right: auto;
107     left: 25px;
108 }

```

Рисунок 6.23 – Стили стрелок «вперёд» - **.slick-next** (файл **slick-theme.css**)

Переходим к стилевому оформлению точек (**.slick-dots**). Меняем отступ сверху от строки миниатюр с «минус 25px» на «минус 35px» (рисунок 6.23).

```

125 .slick-dots
126 {
127     position: absolute;
128     bottom: -35px;
129
130     display: block;

```

Рисунок 6.23 – Отступ «точек» сверху - **.slick-dots**
(файл **slick-theme.css**)

Назначаем цвета и прозрачность (рисунок 6.24) для точек активных (**focus**) и при наведении курсора (**hover**).

```

170 .slick-dots li button:hover,
171 .slick-dots li button:focus
172 {
173     outline: none;
174     opacity: 1;
175     color: #2b8aba;
176 }
177 .slick-dots li button:hover:before,
178 .slick-dots li button:focus:before
179 {
180     opacity: 1;
181     color: #2b8aba;
182 }

```

Рисунок 6.24 – Стили «точек» при наведении курсора - **.slick-dots**
(файл **slick-theme.css**)

И последнее – назначаем (рисунок 6.25) цвет и прозрачность неактивных точек (серый цвет) и точек активных (голубой цвет).

```

199     opacity: 1;
200     color: #ddd;
201
202     -webkit-font-smoothing: antialiased;
203     -moz-osx-font-smoothing: grayscale;
204 }
205 .slick-dots li.slick-active button:before
206 {
207     opacity: 1;
208     color: #2b8aba;
209 }

```

Рисунок 6.25 – Стили «неактивных и активных точек»
(файл **slick-theme.css**)

Обновите страницу карточки товара – стрелки должны появиться. Пощелкайте по ним – слайдер должен работать. Неактивные элементы должны быть серого цвета, активные – голубого, при подведении курсора к неактивным

элементам их цвет меняется на голубой, а у стрелок должна меняться прозрачность (они должны слегка «потемнеть»).



Рисунок 6.26 – Оформленный слайдер

Таким образом, цель достигнута – карточка товара создана и полностью функциональна.

Самостоятельно. Разберите код слайдера (нажать правую кнопку мыши, выбрать пункт «Исследовать элемент»), найдите длину всей «линейки» слайдера в пикселях, подсчитайте, сколько картинок шириной 750px можно на ней разместить. Сообщите результат преподавателю. Также ответьте на контрольные вопросы преподавателя по процессу вёрстки карточки товара.

Лабораторная работа № 7

Верстка страницы заказа товара (index-zakaz.html)

Внешний вид корзины пользователя показан на рисунке 7.1 и представлен в папке «market» в файле «страница заказа.jpg». Правда, на рисунке 7.1 обрезан подвал страницы (для экономии места).

Интернет-магазин
Сотовых телефонов

Телефон: 8 (800) 700-00-01
Режим работы: Будние дни: с 9:00 до 18:00
Суббота, Воскресенье - выходные

Что вы хотите купить?

Главная О магазине Оплата и доставка Покупка в кредит Контакты

Оформление заказа

наименование	количество	стоимость
 Samsung Galaxy S Advance 8Gb (черный)	<input type="text" value="1"/>	12390 <input type="button" value="✖"/>
 Samsung Galaxy S Advance 8Gb (черный)	<input type="text" value="1"/>	12390 <input type="button" value="✖"/>
 Samsung Galaxy S Advance 8Gb (черный)	<input type="text" value="1"/>	12390 <input type="button" value="✖"/>
Итого:	3 шт	34569 руб.

Способы доставки:

- Курьером, 200 руб
- Самовывоз, бесплатно
- В магазин, бесплатно

Информация для доставки:

ФИО Пример: Иванов Сергей Александрович

Е-маил Пример: test@mail.ru

телефон Пример: 8 937 999 99 99

Адрес доставки Пример: г. Москва, пр. Мира, ул. Петра Великого д.19, кв 51.

Примечание Пример: Позвоните пожалуйста после 10 вечера, до этого времени я на работе

Рисунок 7.1 – Корзина пользователя

Как видно из рисунка, на странице «Оформление заказа» отсутствуют левый и правый бары (left-bar и right-bar), содержащие некоторые пункты меню.

Для создания файла данной страницы возьмем любой ранее созданный файл, например, **index-detail.html** и сохраним его под именем **index-zakaz.html**.

Теперь необходимо очистить файл от ненужного содержимого. Удалите всё содержимое (вместе с самими контейнерами) контейнеров `<div id="content">`, `<div id="left-bar">` и `<div id="right-bar">`. Кроме того, для нашей страницы мы обернем всё центральное содержимое страницы «Оформление заказа» в контейнер `<div id="content-zakaz">`. В результате код страницы **index-zakaz.html** должен соответствовать приведённому на рисунке 7.2.

```
<ul class="menu">
  <li><a href="#">Главная</a></li>
  <li><a href="#">О магазине</a></li>
  <li><a href="#">Оплата и доставка</a></li>
  <li><a href="#">Покупка в кредит</a></li>
  <li><a href="#">Контакты</a></li>
</ul>
<div id="contentwrapper">
  <div id="content-zakaz">
    </div> <!-- Закрывающий тег дива content-zakaz -->
</div> <!-- Закрывающий тег дива contentwrapper -->

<div class="clr"></div>
<div class="footer">
  <div class="flogo">
```

Рисунок 7.2 – Начальный код страницы **index-zakaz.html**

Сохраните изменения, посмотрите результат в браузере (рисунок 7.3).



Рисунок 7.3 – Внешний вид заготовки страницы **index-zakaz.html**

Начинаем добавлять код на страницу `index-zakaz.html`. Следует отметить, что практически всё содержимое контента страницы представляет собой **форму** (тэг `<form method="post" action=" ">`). И это естественно – ведь именно через форму можно передать команду на выполнение определенных действий (например, оформить заказ, уменьшить в базе данных количество товаров на величину купленных в данном заказе и т.д.). Кроме того, на форме будут располагаться две таблицы, два заголовка, блок радиокнопок и кнопка «Заказать». На рисунках 7.4 – 7.9 представлено начало кода основного контента страницы `index-zakaz.html`.


```

    </tr> <!-- Закрытие пятой строки таблицы -->
</table>

<div class="sposob-dostavki">
    <h4>Способы доставки:</h4>
    <p><input type="radio" name="1" value="1" /> Курьером, 200 руб</p>
    <p><input type="radio" name="1" value="2" /> Самовывоз, бесплатно</p>
    <p><input type="radio" name="1" value="3" /> В магазин, бесплатно</p>
</div>
<h3>Информация для доставки:</h3>
</div> <!-- Закрывающий тег дива content-zakaz -->
</div> <!-- Закрывающий тег дива contentwrapper -->

```

Рисунок 7.7 – Код файла **index-zakaz.html** (продолжение)

```

<h3>Информация для доставки:</h3>
<table class="zakaz-data" border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td class="zakaz-txt">ФИО</td>
    <td class="zakaz-inpt"><input type="text" name="address" /></td>
    <td class="zakaz-prim">Пример: Иванов Сергей Александрович</td>
  </tr>
  <tr>
    <td class="zakaz-txt">Е-маил</td>
    <td class="zakaz-inpt"><input type="text" name="address" /></td>
    <td class="zakaz-prim">Пример: test@mail.ru</td>
  </tr>
  <tr>
    <td class="zakaz-txt">Телефон</td>
    <td class="zakaz-inpt"><input type="text" name="address" /></td>
    <td class="zakaz-prim">Пример: 8 937 999 99 99</td>
  </tr>
  <tr>
    <td class="zakaz-txt">Адрес доставки</td>
    <td class="zakaz-inpt"><input type="text" name="address" /></td>
    <td class="zakaz-prim">Пример: г. Москва, пр. Мира, ул.
      Петра Великого д.19, кв 51.</td>
  </tr>
  <tr>
    <td class="zakaz-txt" style="vertical-align:top;">Примечание </td>
    <td class="zakaz-txtarea"><textarea></textarea></td>
    <td class="zakaz-prim" style="vertical-align:top;">
      Пример: Позвоните пожалуйста после 10 вечера,
      до этого времени я на работе </td>
  </tr>
</table>
</div> <!-- Закрывающий тег дива content-zakaz -->

```

Рисунок 7.8 – Код файла **index-zakaz.html** (продолжение)

```

<td class="zakaz-txt" style="vertical-align:top;">Примечание </td>
<td class="zakaz-txtarea"><textarea></textarea></td>
<td class="zakaz-prim" style="vertical-align:top;">
Пример: Позвоните пожалуйста после 10 вечера,
до этого времени я на работе </td>
</tr>
</table>

<input type="image" src="images/zakazat.jpg" />
<br /><br /><br /><br />
</form>

</div> <!-- Закрывающий тег дива content-zakaz -->
</div> <!-- Закрывающий тег дива contentwrapper -->

```

Рисунок 7.9 – Код файла index-zakaz.html (окончание)

The screenshot shows a web page for 'Интернет-магазин Сотовых телефонов'. The main content area is titled 'Оформление заказа' and contains a table with the following items:

наименование	количество	стоимость
Samsung Galaxy S Advance 8Gb (черный)	1	12390
Samsung Galaxy S Advance 8Gb (черный)	1	12390
Samsung Galaxy S Advance 8Gb (черный)	1	12390
Итого:	3 шт	34569 руб.

Below the table, there are radio buttons for shipping options: Курьером, 200 руб; Самовывоз, бесплатно; В магазин, бесплатно. There is also a section for 'Информация для доставки' with input fields for ФИО, E-mail, Телефон, Адрес доставки, and Примечание, each with a corresponding example value.

The page footer contains a navigation menu with links: Главная, О магазине, Оплата и доставка, Покупка в кредит, Контакты. The contact information includes the phone number 8 (800) 700-00-01 and the operating hours: Будние дни: с 9:00 до 18:00; Суббота, Воскресенье - выходные.

Рисунок 7.10 – Внешний вид страницы без стилового оформления

Как видно из рисунка, содержание файла соответствует макету, однако стили пока не оформлены. Откройте файл **style.css** и добавьте (в конце файла) стиловое оформление (в зеленой рамке) для элементов содержимого заказа (рисунок 7.11).

```
.tech-harakt ul li{
    font-size:12px; line-height:1.5;
}
```

```
/* ZAKAZ */
#content-zakaz{
    width:98%; margin:0 auto;
}
#content-zakaz h2{
    text-align:center; font:normal 24px Verdana; margin:20px 0 10px 0;
}
.zakaz-main-table{
    width:100%; margin:0 0 30px 0;
}
.zakaz-main-table td{
    padding:10px;
}
.z_top{
    background:#fafbf; border-bottom:1px solid #dee7ec; font:bold 12px Verdana; color:#555;
}
.z_bot{
    background:#fafbf; border-top:1px solid #dee7ec; font:bold 12px Verdana; color:#555;
}
```

Рисунок 7.11 – Код файла **style.css** (начало)

```
.z_bot{
    background:#fafbf; border-top:1px solid #dee7ec; font:bold 12px Verdana; color:#555;
}
```

```
.z_name{
}
.z_name img{
    float:left; margin:0 10px 0 0;
}
.z_name a{
    font:12px Verdana; color:#2b8aba;
}
.z_name a:hover{
    text-decoration:none;
}
.z_kol, .z_price, .z_del{
    text-align:center;
}
.z_kol{
    width:75px;
}
.z_price{
    width:80px; font:24px Candara;
}
.z_del{
    width:65px;
}
.kolvo{
    width:48px; height:22px; font:16px Verdana; color:#000; background:#fafafa;
    border:1px solid #b5b5b5; text-align:center;
}
```

Рисунок 7.12 – Код файла **style.css** (продолжение)

Сохраните изменения, посмотрите результат в браузере (рисунок 7.13).

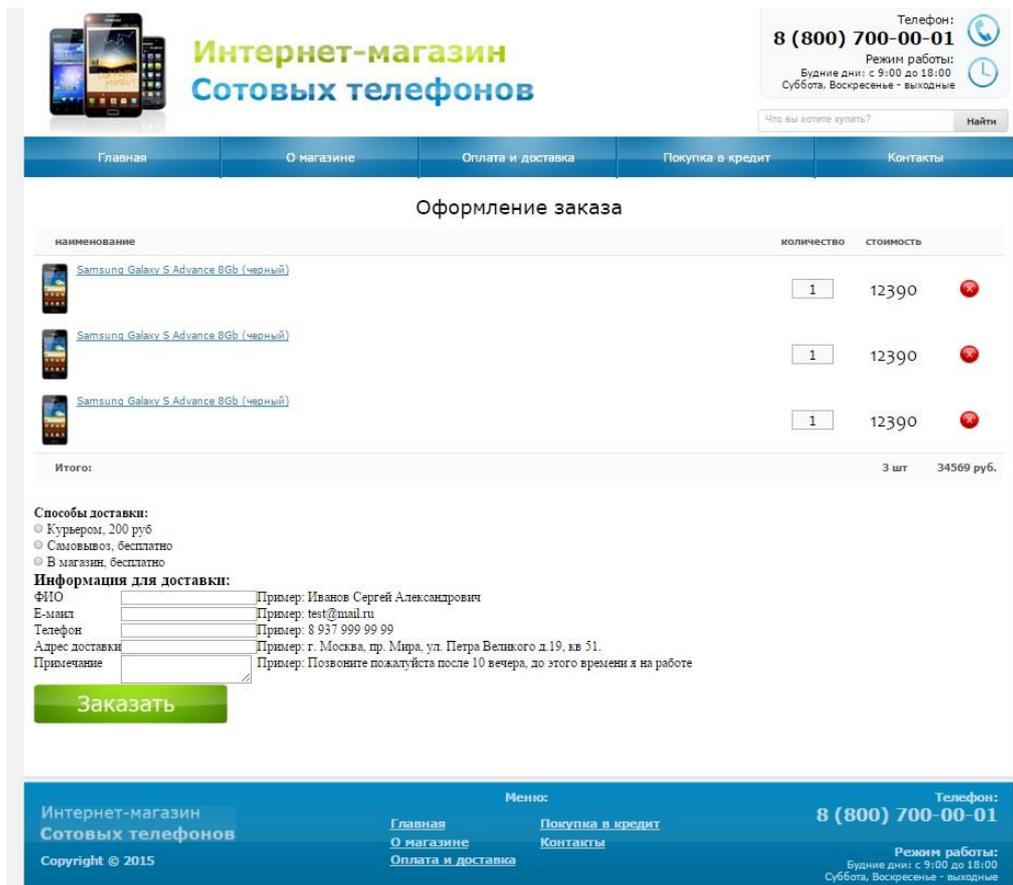


Рисунок 7.13 – Код файла index-detail.html

Как видно из рисунка, первый блок (таблица заказов) оформлен в соответствии с макетом. Продолжаем стилевое оформление страницы заказа. Добавьте в код файла стилей **style.css** следующие строки (рисунки 7.14-7.15).

```
.kolvo{
    width:48px; height:22px; font:16px Verdana; color:#000; background:#fafafa;
    border:1px solid #b5b5b5; text-align:center;
}

.sposob-dostavki p{
    font-size:12px; font-family:Verdana; margin-top:7px;
}
.sposob-dostavki{
    margin:0 0 30px 0;
}
#content-zakaz h4{
    font:bold 14px Verdana;
}
#content-zakaz h3{
    font:normal 18px Verdana;
}
```

Рисунок 7.14 – Код файла style.css (продолжение)

```

#content-zakaz h3{
    font:normal 18px Verdana;
}

.zakaz-data{
    width:100%; background:#fafafa; margin:10px 0 40px 0;
}

.zakaz-data td{
    vertical-align:middle; padding:10px 10px 0 10px;
}

.zakaz-txt{
    font:14px Verdana; color:#555555;
}

.zakaz-prim{
    font:12px Verdana; color:#c0c0c0; font-style:italic;
}

.zakaz-inpt{
}

.zakaz-inpt input{
    width:330px; height:28px; border:1px solid #ccc;
}

.zakaz-txtarea{
    padding:10px 10px 20px 10px !important;
}

.zakaz-txtarea textarea{
    width:330px; height:100px; border:1px solid #ccc;
}

```

Рисунок 7.15 – Код файла **style.css** (окончание)

Сохраните изменения, посмотрите результат в браузере (рисунок 7.16).

Сравните полученный результат с макетом, убедитесь в правильности верстки.

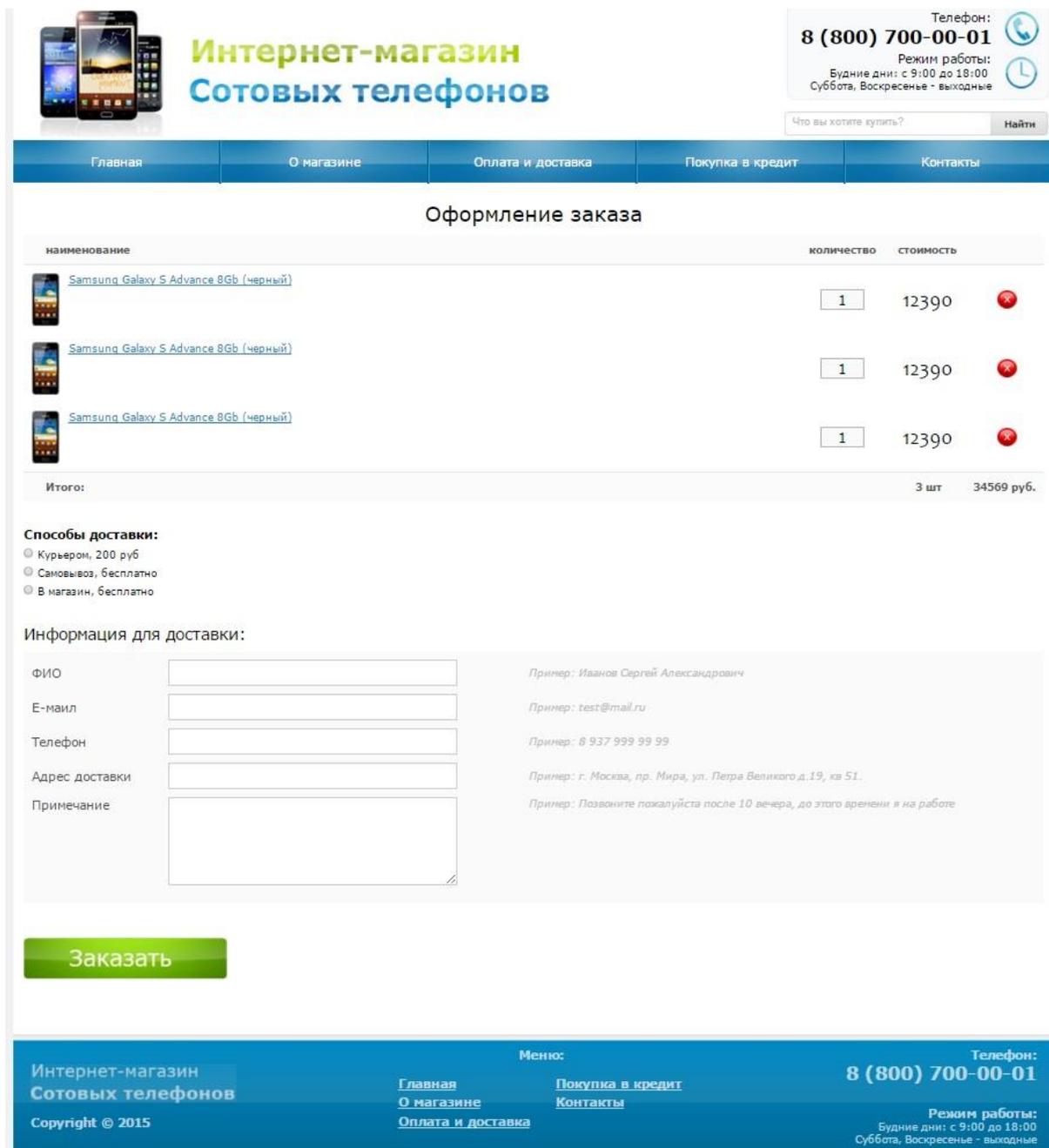


Рисунок 7.16 – Внешний вид готовой страницы «Оформление заказа»

Самостоятельно.

1) Разобраться с приведенным кодом, ответить на любые вопросы преподавателя по оформлению страницы оформления заказа

2) Исправить внешний вид таблицы «Оформление заказа» в соответствии с рисунком 7.17. Нижняя строка («Итого» и далее) должна содержать 4 столбца, а не 2, как на рисунке 7.16.

наименование	количество	стоимость	
Samsung Galaxy S Advance 8Gb (черный)	<input type="text" value="1"/>	12390	
Samsung Galaxy S Advance 8Gb (черный)	<input type="text" value="1"/>	12390	
Samsung Galaxy S Advance 8Gb (черный)	<input type="text" value="1"/>	12390	
Итого:	3 шт	34569 руб.	

Рисунок 7.17 – Исправленный вид таблицы «Оформление заказа»