

Министерство образования и науки Самарской области
ГБПОУ “ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ КОЛЛЕДЖ”

ПРОЕКТ
САЙТОСТРОЕНИЕ

ОУД-12 ИНФОРМАТИКА
09.02.07 ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Студент

В.А. Сабинин

подпись

14.02.2023

Оценка выполнения и защиты проекта

Руководитель

И.В. Федотова

подпись

23.06.2023

Самара 2023г

Оглавление

ВВЕДЕНИЕ	3
Цель исследования – изучить принципы построения и функционирования веб-сайтов.....	4
ГЛАВА 1. Проектирование создания Веб-сайта.....	6
1.1 Сайт-визитка.....	6
1.2 Коммерческий сайт	6
1.3 Интернет-магазин	7
ГЛАВА 2. Технологии создания сайта.....	9
2.1 Производство контента	9
2.2 HTML – язык развертки	10
2.3 CSS – таблицы стилей	12
2.4 Эффективность сайта	15
3.1 Adobe Brackets	19
3.2 Текстовый редактор Sublime Text	20
ЗАКЛЮЧЕНИЕ.....	22

ВВЕДЕНИЕ

В 1962 г. Дж. Ликлайдером, руководителем исследовательского компьютерного проекта экспериментальной сети передачи пакетов в Управлении перспективных исследований и разработок Министерства обороны США (Defense Advanced Research Project Agency, DAPRA), была опубликована серия заметок, в которых обсуждались концепция «Галактической сети» («Galactic Network»). «Галактическая сеть» представлялась как глобальная сеть взаимосвязанных компьютеров, позволяющая любому пользователю получить доступ к данным и программам на компьютерах, объединенных данной сетью. Можно сказать, что эта идея положила начало развитию сети Интернет.

В последнее время сеть Интернет стала одним из важнейших средств маркетинга, которое имеет значительные преимущества по сравнению с традиционными маркетинговыми каналами. Однако возможности, предоставляемые сетью Интернет, настолько новы и необычны, что большинство фирм плохо представляют, как их правильно использовать. Одним из аспектов сети Интернет, требующим маркетингового подхода, является разработка Веб-страниц.

Создание Веб-сайтов является одной из важнейших технологий разработки ресурсов Интернет. Веб-сайт – это информационный ресурс, состоящий из связанных между собой гипертекстовых документов (Веб-страниц), размещенный на Веб-сервере и имеющий индивидуальный адрес. Веб-страница представляет собой текстовый файл с расширением *.html, который содержит текстовую информацию и специальные команды – HTML-коды, определяющие в каком виде эта информация будет отображаться в окне браузера. Хороший сайт, вбирая в себя всю полезную информацию, является лучшей визитной карточкой и коммерческой фирмы, и образовательного учреждения, работая на них в любое время суток.

С точки зрения маркетинга, Веб-сайт – это набор информационных блоков и инструментов для взаимодействия с одним или несколькими сегментами рынка. Какая информация будет представлена на сайте, какие инструменты будут задействованы и как

они будут взаимодействовать между собой – зависит от целей создания сайта, от краткосрочных и долгосрочных задач, а также от типа сегментов целевой аудитории и наличия возможности взаимодействовать с ней тем или иным образом. Грамотно подготовленный Веб-сайт в состоянии решать вопросы привлечения клиентов и увеличения продаж, создания имиджа и изучения потребительского спроса и т.д. Однако даже незначительные промахи при реализации сайта могут нанести существенный ущерб деятельности фирмы.

Для создания Веб-страниц нужен любой браузер для просмотра и контроля сделанной Веб-страницы, а также любой текстовый редактор для создания Веб-страниц. Чтобы создать Веб-страницу, первым делом создают структуру сайта. Сайт – это сервер, имеющий уникальное имя и постоянный адрес, физически подключенный к WWW, или виртуально размещенный на другом, более крупном сервере.

Цель исследования – изучить принципы построения и функционирования веб-сайтов.

1. Проектирование создания Веб-сайта
2. Технологии разработки Веб-сайта
3. Инструменты для разработки Веб-сайта

Объект исследования – создание веб-сайтов.

Предмет исследования – способы и средства разработки веб-сайтов.

Методами для исследования практическое создание веб-сайтов. Информационная база взята из книжной и учебной литературы.

В основе эффективного использования сети Интернет в деятельности фирмы лежат стратегические решения компании, среди которых должно присутствовать и решение о создании корпоративного Веб-сайта. Такое решение должно базироваться на результатах бизнес-планирования и содержать оценку эффективности Веб-сайтов, а также четко сформулированные концепцию бизнеса и цели создания Веб-сайта и критерии их достижения, характеристику целевой аудитории, описание этапов реализации проекта с указанием ответственных исполнителей. Определение целей создания Веб-сайта и

критериев их достижения играют основополагающее значение для определения как структуры Веб-сайта, так и необходимого программного обеспечения. Распространены следующие варианты:

1. Создание дополнительного информационно-рекламного канала, совершенствование коммуникации с действующими и потенциальными партнерами, формирования положительного имиджа компании как предприятия, использующего в своей деятельности современные технологии.
2. Использование сети Интернет для рекламы и сбыта и продаж.
3. Создание портала, обеспечивающего работу с клиентами, партнерами, поставщиками (онлайн заказы, каталог товаров и услуг, прайс-листы, информационная поддержка), подключение платежным системам, создание и развитие систем складирования и доставки.

На принятие решений по реализации Интернет-проекта существенное влияние оказывает определение целевой аудитории. Целевой сегмент может не совпадать с сегментом, на который фирма ориентируется в реальном бизнесе, поскольку сеть Интернет предоставляет гораздо более широкие возможности, например, в плане географического охвата, а может оказаться значительно уже из-за технических проблем в том или ином регионе. На этом этапе следует определить характеристики, которые описывают целевой сегмент. Кроме традиционных факторов, таких как демографические, поведенческие, психологические и другие, необходимо учитывать характеристики, связанные с особенностями компьютерного обеспечения потребителей в сети (быстродействие компьютера, объем памяти, вид модема, скорость и качество передачи данных по линии связи, качество монитор и т.д.).

Знание характеристик потенциальных потребителей позволит привлечь на Веб-страницу максимальное число посетителей. В зависимости от выбранной структуры Веб-сайта, будь то визитка или портал, меняются затраты на реализацию каждого из этапов, но общий подход к проекту остается одинаковым.

ГЛАВА 1. Проектирование создания Веб-сайта

1.1 Сайт-визитка

Сайт-визитка это – одна или несколько страниц в интернете, расположенных на одном доменном имени, предоставляющая общую информацию о владельце, основной перечень услуг, товаров, контактные данные и раздел с ответами на основные вопросы. Главная задача такого сайта – предоставить пользователю основной перечень информации о компании или частном лице.

Стандартная структура проста и состоит всего из двух уровней:

1. Главная
2. Остальные страницы

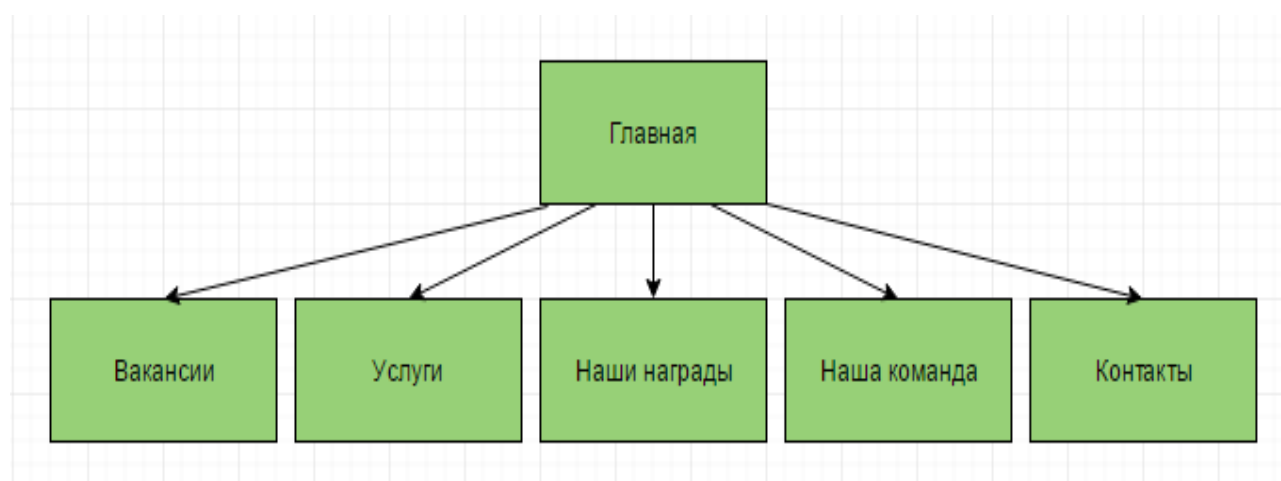


Рис. 1.1. Структура сайта-визитки

1.2 Коммерческий сайт

Здесь уже сложнее – нужно будет внедрять дополнительные уровни страниц. Если вкратце, то основными страницами будут главные направления вашей деятельности, страницами второго уровня – их разновидности. Такое решение идеально для сайтов услуг и любых других коммерческих сайтов без функций интернет-магазина.

Структура продающего сайта:

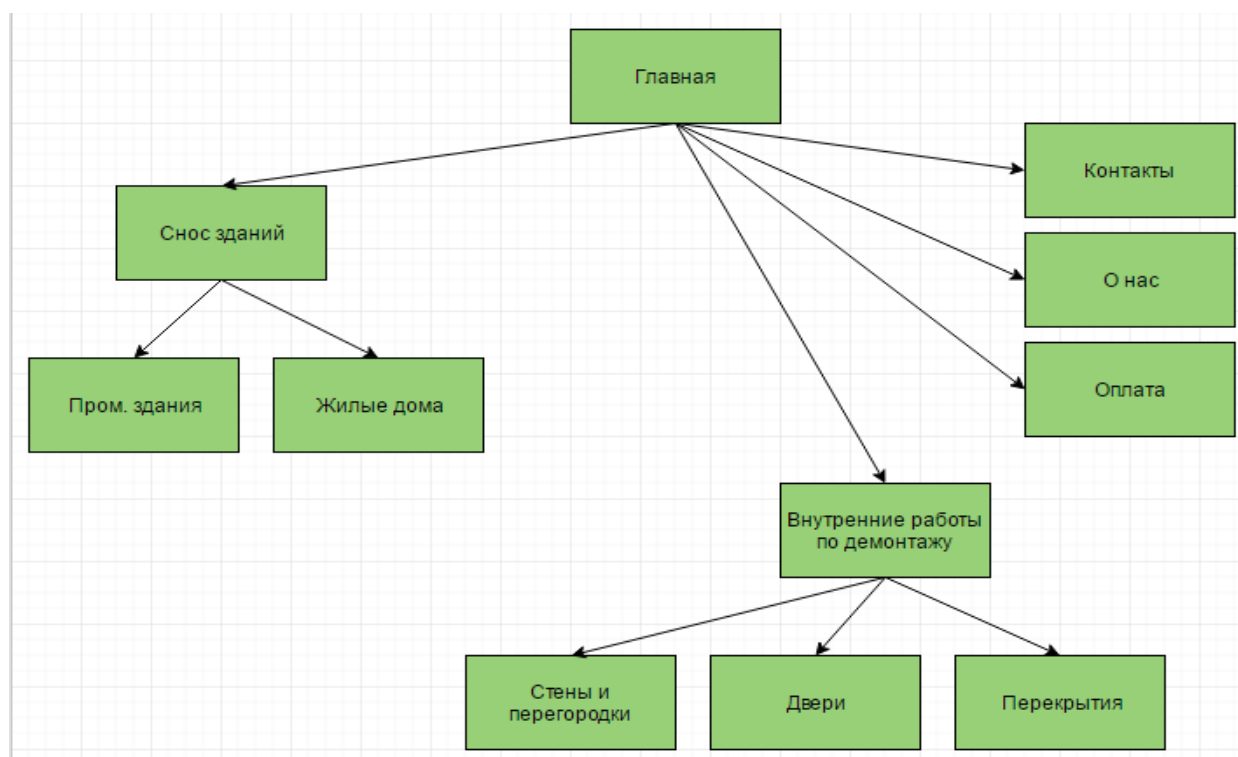


Рис. 1.2. Структура коммерческого сайта

1.3 Интернет-магазин

Интернет-магазин позволяет перевести продажи товаров непосредственно в онлайн-пространство, и многие компании сегодня используют именно такой тип сайтов для получения прибыли через интернет.

Это уже совершенно иной уровень организации сайта. Чтобы охватить все возможные запросы, которые могут ввести в поисковой строке ваши потенциальные покупатели, вам нужно будет внедрить не только систему разделов и подразделов, но и добавить на сайт фильтры. Главное, правильно определить, что пойдет в фильтры, а что будет основным свойством продаваемого товара.

Один из ключевых моментов – грамотная структура каталога товаров. Структура сайта интернет-магазина - это определенная система взаимного расположения веб-страниц и разделов, логичный способ их связи. От того, насколько она удобна, зависит непосредственное перемещение пользователя по сайту.

Задача структуры - быстро сориентировать клиента в каталоге, чтобы он легко мог найти нужный товар, выбрать его по своим критериям, сравнить с другими. Существует два инструмента для создания структуры: категории и фильтр.

Так, например, для магазина диванов логично будет разделить диваны на кожаные и тканевые (материал), а не на синие и красные (цвет). Если фишка – это разноцветные диваны, то будет лучше разделить их по цветам, сделав материал второстепенным признаком и вынести его в фильтр. Фильтры нужны тогда, когда одному товару присуще сразу несколько качеств, которые интересуют людей в запросах, например, «цвет + форма» (кровать белая круглая) или «размер + форма + материал» (большой угловой кожаный диван).

Если не получается собственноручно создать страницы со всевозможными вариантами сочетаний параметров товаров – нужны фильтры.

Если у товаров всего один-два параметра (например, если продается конкретный вид продукции - бамбуковые одеяла, и единственный параметр, который изменяется, это размер), можно обойтись без фильтров. Фильтры могут быть одинаковыми для каждого уровня и раздела каталога, могут разрабатываться отдельно – все зависит от специфики.

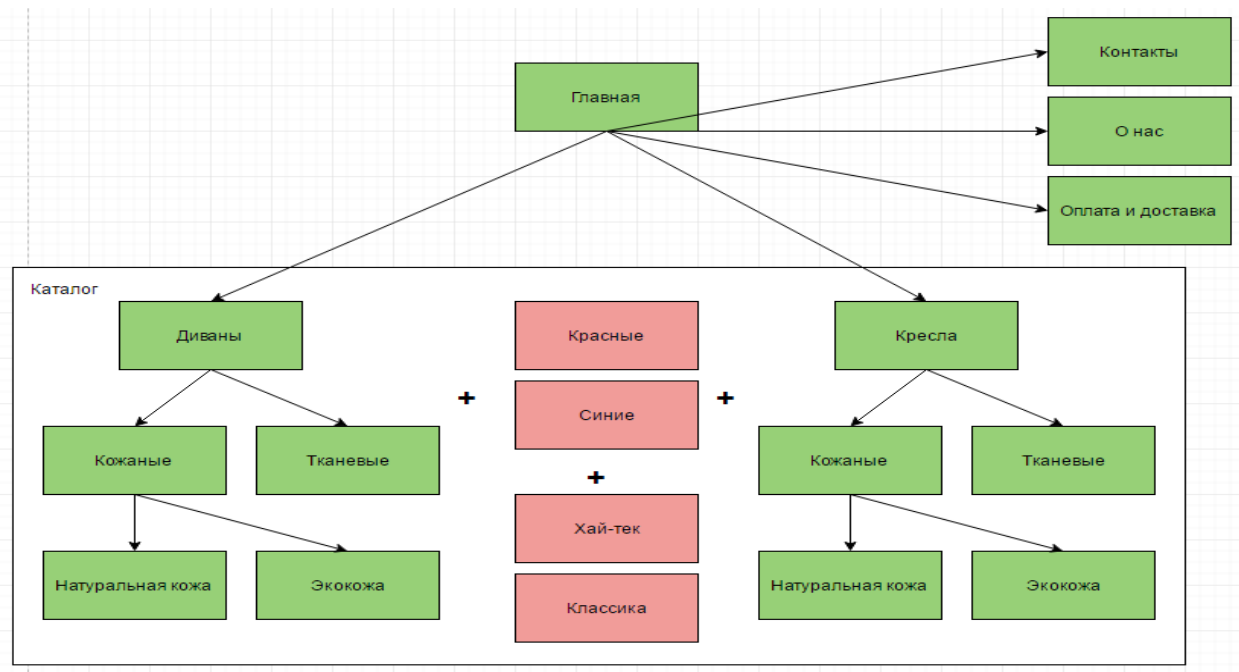


Рис. 1.3. Структура интернет-магазина

ГЛАВА 2. Технологии создания сайта

2.1 Производство контента

Контент – информационное наполнение сайта. Тип и формат представления информации.

Информация, представленная на Веб-странице, должна удовлетворять следующим критериям:

- должна соответствовать целям создания Веб-страницы;
- должна учитывать особенности целевого сегмента потребителей;
- должна быть в определенной степени уникальной, чтобы привлечь внимание посетителей. Тем более, что в сети Интернет существует множество похожих Веб-страниц, и конкуренция между ними достаточно сильная;
- должна быть оперативной. Для поддержания интереса к Веб-странице ее необходимо постоянно обновлять и модернизировать. Можно установить, как часто будут обновляться данные. Веб-страницы, информация на которых обновляется раз в месяц, могут рассчитывать на повторные посещения пользователей только через несколько месяцев. Если же новые сведения появляются каждый день, то повторные визиты будут происходить раз в несколько дней. Для того чтобы показать пользователям, что Веб-страница обновляется, можно указать на ней дату ее создания и последнего обновления;
- должна быть объективной и достоверной. Посетители, которые обнаружат ошибочную или неточную информацию, вряд ли захотят повторно обратиться к этой странице;
- не должна носить излишне рекламный, навязчивый характер.

На Веб-странице можно представить разнообразные материалы: информацию о фирме и ее продукции, каталог продуктов и услуг, списки дилеров, адреса торговых представителей, справочные сведения, новости, сетевой магазин, раздел обратной связи, вакансии и т.д. Следует размещать только ту информацию, которая является наиболее интересной и полезной.

Одна из основных задач, которую следует ставить перед собой при составлении текстов для Веб-страниц – это использование технической и развлекательной информации для выделения особенностей фирмы. Низкая стоимость системы WWW приводит к тому, что у любой Веб-страницы существует множество конкурентов. Поэтому тексты на Веб-странице должны четко отвечать на вопросы: Чем отличается компания от всех остальных? Как можно обосновать и подчеркнуть эти отличия?

Некоторые сведения следует указывать на каждой Веб-странице. К ним относятся, в частности, электронный адрес и имя администратора сервера, а также название компании и сведения об авторском праве.

2.2 HTML – язык развертки

HTML получил широкую популярность. На данный момент – это единственный язык, с помощью которого создается разметка сайта.

Происхождение языка HTML начинается с GML, стандарт которого был принят в 80-х годах прошлого века с названием SGML (Standard Generalized Markup Language – стандартный обобщенный язык разметки). Предложенный в 1974 году Чарльзом Голдфарбом, SGML представляет собой метаязык – систему для описания других языков. При всех своих возможностях он слишком сложен для большинства браузеров. Только спецификация SGML занимает свыше 500 страниц. Поэтому на основе SGML был создан HTML – специализированный, гораздо более простой язык. года сообществом WHATWG ведется разработка пятой версии HTML. Теги языка HTML представляют собой служебные слова языка, заключенные в треугольные скобки. Например, многие теги используются парами. Тогда различают открывающий и закрывающий тег. Например, закрывающий тег начинается с косой черты, за которой идет название открывающего тега. Внутри пары обычно располагается информация и другие теги. Поэтому парные теги называют контейнерами.

Так как HTML-документ может представлять собой целый набор файлов, то лучше всего поместить их все в отдельную папку. А файл-запуск назвать стандартным именем “index.html”, известным любому браузеру. После того как документ сохранен, его можно

просмотреть в браузере. Браузер «понимает» и исполняет html-код. Через меню «Вид/Источник» можно снова вернуться к Блокноту с html-кодом, внести изменения, сохранить их в файле. Если в HTML коде содержатся ошибки, то браузер просто проигнорирует их и не выдаст Вам или посетителю страницы сообщений об ошибках. Это делает процедуру отладки более трудоемкой. Любой браузер “понимает” язык тегов. Веб-страница – это еще одно название HTML-документа. Оно дается тем документам, которые размещаются в сети Интернет. Набор веб-страниц, объединенных общей темой и гиперсвязями и расположенных на одном сервере, называют сайтом. Как правило, сайт содержит гиперссылки на другие сайты. Огромное множество взаимосвязанных веб-сайтов и составляют Всемирную паутину сети Интернет. Еще одно применение HTML-документы нашли в сфере разработки электронных мультимедийных учебников и обучающих программ. Одними из самых распространенных HTML-редакторов являются: Adobe Brackets.

```
<body>
  <div class="container">
    <div class="line" id="header">
      
      <ul class="menu" style="margin: 50px 167px">
        <li><a href="index.html" id="menu-a">Home</a></li>
        <li><a href="products.html">Products</a></li>
        <li><a href="#">History</a></li>
        <li><a href="#">Showroom</a></li>
        <li><a href="contacts.html">Contact</a></li>
        <li><a href="#" class="search"></a></li>
      </ul>
      <div class="text" style="margin: 288px 0 0 98px; width: 495px;">
        <h2 class="text1">TRENDING</h2>
        <h1 class="text2">FISHNET CHAIR</h1>
        <p class="text3">Seat and back with upholstery made of cold cure foam. Steel frame, available in matt powder-coated black</p>
        <a href="details.html" class="orderus">ORDER US </a>
      </div>
    </div>
    <div class="line">
      <ul class="menu-featured">
        <li id="menu-featured-a"><a href="#">Featured</a></li>
        <li><a href="#">New</a></li>
      </ul>
      <a href="details.html" class="button-explore">EXPLORE ALL </a>
      <div class="catalog">
        <a href="details.html" class="product1"></a>
        <a href="details.html" class="product1"></a>
        <a href="details.html" class="product1"></a>
        <a href="details.html" class="product1"></a>
      </div>
    </div>
    <div class="line">
      <div class="product4" style="background-image: url(img/products/promo-img1.jpg);">
        <div class="text" style="margin: 228px 51px; width: 400px;">
          <h2 style="color: #1e88e5;">TRENDING</h2>
          <h1 style="font-weight: 700; font-size: 36px; letter-spacing: 0px;">Fishnet Chair</h1>
          <p>Seat and back with upholstery made of cold cure foam. Steel frame, available in matt</p>
          <a href="details.html" class="orderus">ORDER US </a>
        </div>
      </div>
    </div>
  </div>
```

Рис. 2.2. Пример кода HTML

2.3 CSS – таблицы стилей

Таблицы стилей предназначены в первую очередь для визуального оформления структурных элементов страницы. Структурные же элементы создают все те теги, которые рассматривались ранее. Правда некоторые теги (<FRON>, и т.д.) из рассмотренных использовались для стилевого форматирования элементов, но в версии HTML 4.0 они считаются устаревшими и не рекомендуются для дальнейшего использования на вновь создаваемых страницах.

Каскадные таблицы стилей позволяют задать уже известным тегам авторский стиль оформления. Например, возможно всем заголовкам первого уровня придать зеленый цвет и больший размер или у всех элементов маркированного списка изменить маркер на красивое графическое изображение.

Таким образом, таблицы стилей снимают ограничения старых версий HTML на набор свойств тех или иных структурных элементов документа. Применение CSS начинается с помещения в раздел заголовка HTML-документа следующего тега.

```
<head>  
<meta http-equiv="Content-Style-Type" content="text/css">  
</head>
```

Существует несколько способов определения стиля элемента.

Способ 1. Атрибут style

Стиль задается непосредственно для какого-либо элемента в основном разделе документа с помощью параметра STYLE. Например:

```
<p style="font-size: 20px; font-style: italic; color: blue">
```

Здесь слово STYLE выступает в роли атрибута тега. Из примера видно, что значение параметра STYLE представляет собой последовательность свойств и их значений, разделенных точкой с запятой. Очередное свойство и его значение в этой последовательности разделяются двоеточием.

Способ 2. Таблица стилей с тегом <style>

Атрибут STYLE в предыдущем способе задает свойства только одного элемента, а с помощью тега <STYLE> можно задать свойства всех одноименных элементов. Контейнер <STYLE> размещается в заголовке <HEAD></HEAD>.

Таблица стилей может состоять из одного или нескольких правил, определяющих внешний вид описываемых элементов. Правило состоит из 2-х частей: селектора и описания.

Стиль для вложенных элементов Р В будет использоваться браузером только тогда, когда элемент В окажется внутри элемента Р. При этом свойство font-size будет унаследовано от предыдущего описания стиля тега В.

Последняя строка тега STYLE показывает, как можно одно описание свойства применить сразу к нескольким элементам, перечисленным через запятую.

Способ 3. Классы

Предыдущий способ позволяет присвоить любому элементу только один стиль. Иногда удобнее назначать для стилей имена и указывать их вместе с элементами. Такие универсальные стили называют классами.

При желании можно создать любое количество классов, каждый из которых будет иметь свой вариант форматирования для одного элемента.

Способ 4. Селекторы id и контекстные селекторы id

Этим способом можно создавать классы, связанные с нестандартными html-элементами, то есть элементами веб-разработчика.

С атрибутом id тесно связан структурный элемент <DIV> (от англ. Division – часть, раздел). Тег <DIV> не описывает базовых элементов html-документа, таких как заголовки, абзац, список, таблица и т.д. Он предназначен для описания какой-либо структуры, созданной самим дизайнером.

Типичным примером такого элемента является навигационная панель имя такой структуре дает тоже разработчик через атрибут id. Таблица стилей используется для придания внешнего вида структурному элементу.

```
@import url('https://fonts.googleapis.com/css?
family=Montserrat:100,400,700');

* {
    margin: 0;
    padding: 0;
}

body {
    font-family: 'Montserrat', sans-serif;
}

a {
    text-decoration: none;
}

ul {
    list-style-type: none;
}

input {
    font-family: 'Montserrat', sans-serif;
}

.container {
    margin: auto;
    width: 1366px;
    background-color: rgb(251, 251, 251);
    box-shadow: 0px 2px 2px 0px rgba(0, 0, 0, 0.25);
}

div.line {
    display: inline-block;
    position: relative;
    width: 100%;
    outline: 0;
    border: 0;
}
```

Рис. 2.3. Пример кода CSS

2.4 Эффективность сайта

Необходимо заботиться о читабельности страницы. Веб-дизайнер в процессе верстки постоянно держит перед глазами текст, так что почти выучивает его наизусть; к тому же содержимое текста ему обычно неинтересно, а зачастую и непонятно – Веб-дизайнер по специальности чаще всего является художником.

Отсюда и возникают Веб-страницы, переливающиеся всеми цветами радуги, которые очень красиво смотрятся, но которые совершенно невозможно читать.

Не следует делать предположений о каких-либо параметрах экрана, на котором будет отображаться страничка и о браузере посетителя. Необходимо как можно меньше параметров указывать в абсолютных единицах (пикселах), если это не вызвано необходимостью – при другом разрешении экрана это приведет к выходу изображения за границу видимой области или к неполному использованию экранной площади.

Не следует навязывать без необходимости своих установок пользователю – тип, размер и цвет шрифта, а также цвет остальных элементов страницы должны остаться на усмотрение пользователя, чтобы не нарушать привычной ему обстановки и не вызывать проблем адаптации к привычкам. Что касается шрифта – если указать конкретный тип, то это часто приводит к проблемам русификации, если указанный шрифт оказался в другой кодировке или вообще нерусифицированным; конкретный размер затруднит чтение (обычно человек подбирает наиболее удобный для себя размер).

Не стоит делать широких полей слева и справа от текста – если пользователь любит узкий текст, он может сузить окно браузера, а вообще надо максимально полно задействовать экранное пространство.

Не указывать кодировку в тексте (заголовке) HTML-страницы – ее могут конвертировать как серверы, так и сами пользователи в приемлемую для чтения кодировку, а указание на кодировку в тексте HTML-страницы останется неизменным и будет затруднять ее чтение при несоответствии указанной и действительной кодировок.

Не надо использовать картинки там, где можно обойтись без них:

- обычно картинка занимает гораздо больше места, чем текст той же информативности, а значит, замедляется загрузка (по крайней мере старайтесь сделать файл с картинкой поменьше, выбирайте между gif и jpeg);
- картинка-ссылка без бордюра (BORDER=0) не выделяется так явно, как текст-ссылка, а с бордюром выглядит некрасиво;
- размер шрифта пользователь подбирает под свое зрение, расположение текста браузер подгоняет под экран и шрифт пользователя, а картинка всегда имеет фиксированный размер;
- движущаяся картинка забирает ресурсы процессора и отвлекает внимание посетителя от полезной информации;
- на случай, если у пользователя отключена загрузка картинок, картинки должны иметь подписи (ALT=«...»);
- если необходимо опубликовать большую картинку, не надо внедрять ее в информационную страницу – лучше поместить туда уменьшенную копию, а саму страницу загружать по желанию пользователя.

Размер информационных страниц надо подобрать таким, чтобы они загружались за приемлемое время, и чтобы переходы к следующей страницы были не слишком часты. Ориентировочно размер страницы (без учета картинок) для российских линий – 10 KB. Исключения возможны для страниц, составляющих единое целое, если предполагается, что пользователь готов сначала загрузить ее целиком, а потом читать – для них пределом можно считать 100 KB.

Размер и компоновка индексных (навигационных) страниц должны подбираться из предположения, что страница целиком уместится на экране, максимум – на двух экранах. Если это не соблюдается, навигационная страница обязательно должна иметь внутреннюю индексацию.

Не стоит злоупотреблять таблицами, по крайней мере, не делать их большими – таблицы начинают выводиться на экран только после того, как загрузятся целиком, а это создает ощущение медленности загрузки.

Таблицы помогают красиво разместить информацию и эффективнее использовать экранное пространство, но не надо загонять все содержимое Веб-страницы в одну таблицу.

Не стоит полагаться на «наглядные» (WYSIWYG) редакторы HTML – именно они дают подавляющее большинство нарушений вышеизложенных принципов и делают страницы нечитабельными.

Дело в том, что HTML как раз рассчитан на отображение на произвольном (в разумных пределах) устройстве, а при «наглядном» редактировании создается страница, которая красиво (по крайней мере с точки зрения ее создателя) выглядит на экране редактора, но никак не проверяется вид страницы при других характеристиках экрана: разрешении, количестве цветов, типе и размере шрифта. Часто встречаются такие недочеты в HTML-страницах, подготовленных «наглядными» редакторами:

- явное указание типа и размера шрифта;
- тэги перевода строки (
) в тех местах, где происходил переход на новую строку при отображении в окне редактора – т.е. принудительное разбиение на строки;
- явное указание размеров таблиц в пикселях;
- выравнивание текста по горизонтали с помощью жестких пробелов (), а по вертикали – с помощью параграфов (<P>) или переводов строки (
) – это, конечно, вина Веб-дизайнера, но провоцировала его именно «наглядность» редактирования;
- кодирование русского текста в виде &#число;, причем числа немного больше 1000; по-видимому, имеется в виду кодировка UniCode, а если OS не поддерживает UniCode, то пользователь не сможет прочитать страницу.

Помимо того, что «наглядные» редакторы могут сделать страницу, привязанную к характеристикам экрана, они часто внедряют в нее лишние тэги, делая ее гораздо больше (в худшем случае – раза в полтора), чем необходимо.

Нужно делать страницу так, чтобы HTML-текст был читабелен даже без браузера, в исходном виде – это облегчит последующее внесение исправлений и вылавливание ошибок.

Не стоит злоупотреблять фреймами – фрейм иногда облегчает навигацию по сайту, зато отнимает экранную площадь, на которой мог бы разместиться полезный текст.

К тому же фрейм не позволяет определить URL страницы, заинтересовавшей посетителя – в строке «Location» всегда отображается URL фрейм-страницы, (и это почти логично – ведь на экране присутствуют несколько страниц; хотя можно было бы отображать URL активного фрейма, но авторы браузеров этого не сделали).

Ссылка со страницы, включенной во фрейм, на страницу другого сайта должна содержать указание загружать ее на полный экран (TARGET=_top), ибо показывать чужую страницу в своем фрейме – дурной тон.

Если дается ссылка на другой сайт, можно внедрить в страницу логотип того сайта или той страницы, с которой он берется; лучше не копировать картинку на свой сайт, а сделать ссылку () прямо на тот сайт – это экономит место на диске, а если пользователь переходит по ссылке, то браузеру при загрузке той страницы не понадобится грузить ту же картинку из другого места. При тестировании создается впечатление, что локальная картинка грузится гораздо быстрее, чем картинка с того сервера; помните, что это относится только к локальной сети.

Не стоит злоупотреблять новомодными технологиями – Java, JavaScript, ActiveX, фреймами, включением звука: все это может не поддерживаться браузером и машиной посетителя (или может быть отключено), а без какой-либо части эффект от всей композиции окажется утерян.

С помощью программирования на разных языках можно по всякому издеваться над посетителем вплоть до того, что он не сможет уйти со страницы. Помните, что пользователь всегда может перезагрузить компьютер; на такую страницу он, скорее всего, никогда не вернется, а информацию, расположенную на странице, забудет в процессе борьбы с злобной программой.

ГЛАВА 3. Инструменты для разработки сайта

3.1 Adobe Brackets

Adobe Brackets представляет из себя редактор с открытым исходным кодом.

Предназначен для создания веб-приложений. Это редактор кода общего назначения и предназначен для людей, владеющих языками программирования. Имеет множество расширений, которые можно загрузить с сайта программы. Как заявляют создатели Brackets, их цель – создать лучший редактор кода для JavaScript, HTML и CSS и связанных с ними открытых веб-технологий.

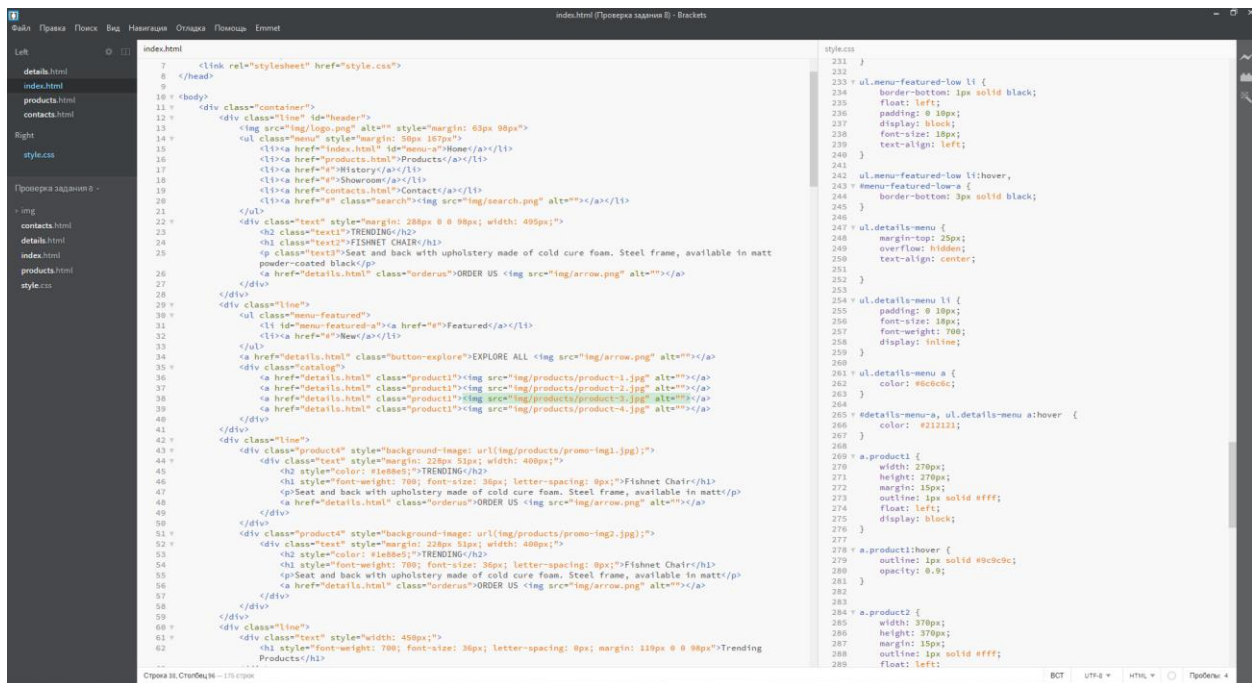


Рис. 3.1. текстовый редактор Adobe Brackets

В редакторе нет необходимости переключения между документами HTML и CSS – их можно поставить одновременно в окне программы. Также возможно быстрое редактирование правил CSS при установке курсора в любом теге html.

Обычно для просмотра страницы в браузере нужно предварительное сохранение файла. В Brackets это не нужно. Эта программа открывает прямое соединение с локальным браузером и направляет изменения CSS, как только вы их напечатаете.

Здесь нет нужды копировать и вставлять финальный CSS обратно в редактор. Код запускается в браузере, но живет в редакторе. Программа поддерживает «Интерактивный просмотр».

В программе легко понимать, как изменения в HTML и CSS отразятся на странице. Когда ваш курсор находится на CSS-правиле, редактор подсветит все затронутые элементы в браузере. То же самое и с редактированием HTML-файла, редактор будет подсвечивать соответствующие HTML-элементы в браузере.

Brackets позиционируется как текстовый редактор, но по факту он больше напоминает полноценную IDE. Огромное количество плагинов позволяет превратить данный текстовый редактор в мощный комбайн для ВЕБ-разработки.

Установка плагинов происходит в окне Менеджера расширений, там же можно установить и темы оформления. Плагины в Brackets позволяют расширить стандартный функционал и облегчить разработку проектов.

3.2 Текстовый редактор Sublime Text

Sublime Text является бесплатным редактором текстовых файлов и поддерживает синтаксис огромного количества языков программирования (более пятидесяти) в среде Windows. Программа потребляет достаточно мало системных ресурсов и, соответственно, совсем не требовательна к железу компьютера, при этом она имеет огромные возможности — как для текстового редактора, и неплохую скорость работы.

Sublime Text часто сравнивают со встроенным в Windows блокнотом, описывая преимущества и недостатки обеих программ. Эти программы сильно отличаются друг от друга. Блокнот сделан для небольшой коррекции текста, сохранения каких-то записок, текстовой информации, просмотра содержимого файла Sublime Text, в свою очередь, сделан для того чтоб облегчить работу программисту (причем достаточно широкого профиля), предлагая универсальный редактор кода.

Чаще всего Sublime Text используют для веб-программирования, то есть как HTML, Python или JavaScript-редактор.

Начинающие веб-программисты изначально часто пользуются Windows-блокнотом, как средством, которое всегда есть под рукой. Впоследствии, когда объем кода увеличивается, разобраться в нем становится все труднее и труднее и тут встает вопрос об установке программы, которая смогла бы облегчить работу.

У этой программы есть масса преимуществ по сравнению с другими довольно известными приложениями, например, такими как Dreamweaver или Microsoft FrontPage. Это более сложные программы с довольно большими возможностями, но они занимают определенное место на жестком диске и потребляют системные ресурсы не сравнимые с работой Sublime Text. Во-вторых они не бесплатны. Так же Sublime Text легче и быстрее по сравнению со многими тяжеловесными редакторами.

Sublime Text программа, как и любая другая, требует ее установки, но есть и портативная версия Sublime Text Portable.

Поддержка синтаксиса большого количества языков программирования заключается в подсветке управляющих элементов и командных слов. Также при необходимости можно вызвать выпадающее меню комбинацией «Ctrl + пробел» в котором будут перечислены все атрибуты и теги языка программирования, при наборе командного слова оно будет подсвечиваться в этом меню.

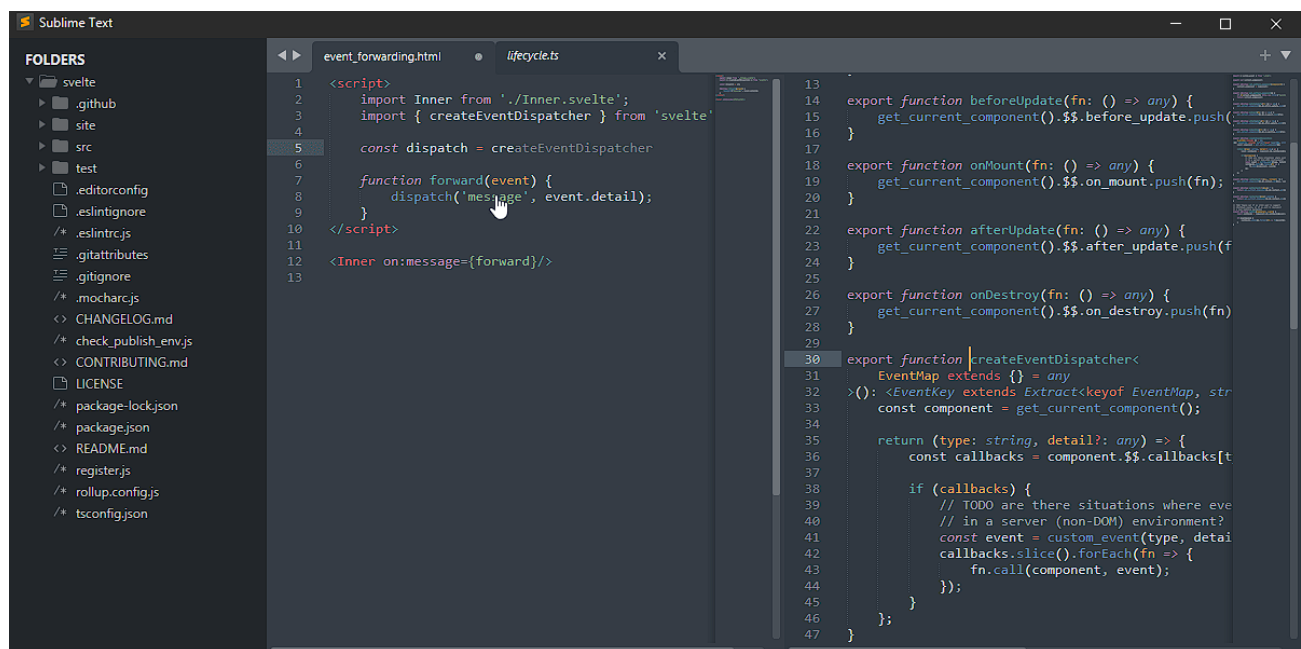


Рис. 3.2. текстовый редактор Sublime Text

ЗАКЛЮЧЕНИЕ

В данной проектной работе мной были изучены основные принципы при разработке Веб-сайта. Узнали, что прежде, чем приступить к разработке сайта необходимо определить цель его создания. Обычно их бывает три:

- создание дополнительного информационно-рекламного канала, совершенствование коммуникации с действующими и потенциальными партнерами, формирования положительного имиджа компании как предприятия, использующего в своей деятельности современные технологии;
- использование сети Интернет для рекламы и сбыта и продаж;
- создание портала, обеспечивающего работу с клиентами, партнерами, поставщиками (онлайн заказы, каталог товаров и услуг, прайс-листы, информационная поддержка), подключение.

В зависимости от цели выбирается структура сайта, их бывает много, но были выделены три соответственно: сайт-визитка, коммерческий сайт, интернет-магазин. Где сайт-визитка самый простой в построении сайт, в то время как интернет-магазин самый многофункциональный и труднореализуемый вид сайта.

После того, как определена структура сайта предстоит выбрать каталоги, подкаталоги и их наполнение контентом. Для лучшей работы сайта контент должен быть отфильтрован согласно рекомендациям. Текст должен быть понятным и хорошо заметным, изображения необходимо доводить до минимального размера без потери видимого качества. А также не перебарщивать с наполнением веб-страниц, чтобы не отпугивать посетителей сайта.

Познакомились с языком разметки HTML, который и позволяет написать код веб-страницы. Дизайн веб-страниц позволяет настроить таблица стилей CSS. Код стилей CSS может быть написан как внутри HTML-документа, так и в отдельном файле *.css формата.

Для редактирования кода можно использовать любой текстовый редактор, например, Sublime Text. Либо специализированную среду разработки, самой популярной является – Adobe Brackets.

В случае, когда у разработчика недостаточно знаний или мало времени можно использовать специальные онлайн-конструкторы для создания веб-сайтов. Они позволяют быстро и легко создавать веб-страницы. Конструкторы так же можно использовать для наброски веб-страницы, чтобы определить дизайн будущего сайта, прежде, чем приступить к полноценному его созданию.