

*Ситников С. Ю.*

**Курс лекций (18)  
по дисциплине «Web-дизайн»  
для группы PCO-1-18**

10.02.2020

**Тема 1**

**Тема: Основные понятия Web-дизайна**

**Тема2**

**Тема: Структура HTML документа**

**Тема 3**

**Тема: Цветовые спецификации. Графические элементы**

**Тема 4**

**Тема: Ввод текстовой информации**

**Тема 5**

**Тема: Гиперссылки**

**Тема 6**

**Тема: Технологии web-дизайна**

**Тема 7**

**Тема: Принципы компоновки и алгоритм создания web- сайта**

**Тема8**

**Тема: Графические изображения. Особенности веб-графики**

**Тема9**

**Тема: Этапы создания web-сайта**

**Тема 10**

**Тема: SAAS и CMS**

**Тема 11**

**Тема: CMS World Press и SEO**

**Тема 12**

**Тема: Способы размещения сайтов в сети Интернет**

## Тема 1

### Тема: Основные понятия Web-дизайна

*Цели: Ознакомить учащихся с основными понятиями web-дизайна; основными задачами web-дизайна в Интернете функции web-дизайнера; определить роль и предмет изучения дисциплины web-дизайна.*

План урока.

1. Организационный момент.
2. Изложение нового материала.
  - 1) Понятие web-дизайна
  - 2) Задачи web-дизайна
  - 3) Функции web-дизайнера
3. Закрепление изученного.
4. Итог урока.

Ход урока.

#### Организационный момент.

Отметить присутствующих. Сообщить тему урока. Настроить учащихся на восприятие материала.

#### Изложение нового материала.

#### Понятие web- дизайна

Термин Web-дизайн состоит из двух частей. Первая часть - Web - это сокращенное написание слов World Wide Web - Всемирная Паутина. Вторая часть термина - слово "дизайн" - происходит от английского слова design, что означает: проектировать, конструировать, планировать, создавать – рисунок. Таким образом, перевод термина Web-design с английского языка на русский язык может быть таким: "проектирование для Интернета".

Web-дизайн - новый жанр изобразительного искусства, основанный на применении специфических инструментов (компьютера и программ-редакторов), специальных языков программирования и разметки (HTML, CSS, JavaScript; PHP и др.) и сетевых технологий (CGI, SSI и др.). Целью Web-дизайна является проектирование объектов (Web-страниц, Java-апплетов, скриптов, баннеров, отдельных элементов Web-страниц), размещаемых в Интернете.

## WEB-разработка

**Front-end** и **Back-end** разработка – это процесс создания сайта, разделенный на две части, клиентскую и серверную.



### Что такое front-end разработка?

Front-end разработка связана с дизайном и интерфейсом сайта. Специалисты, занимающиеся этой частью сайта, называются front-end разработчиками или web-дизайнерами (веб-мастерами).

Клиентская часть (т.е. front-end разработка), к ней можно отнести HTML+CSS-верстку, а также создание PSD-макета и шаблонов для сайта, пользовательского интерфейса и специальных скриптов (js), отвечающих за визуализацию и анимацию.

Начинающего разработчика, освоившего только HTML+CSS, называют *верстальщиком*. А вот **Web developer** – это фронтенд-разработчик, кроме html+css знающий JavaScript, PhotoShop, CMS, SEO-оптимизацию; имеющий представление о back-end.

## Чем занимается Frontend Developer?

за что этим специалистам платят от \$300 до \$3000 в месяц



**1 шаг**  
Получаете "картинку" сайта в PSD

Вам не нужно уметь рисовать. Вы получаете от дизайнера готовую "картинку" сайта в Photoshop.



**2 шаг**  
"Разрезаете" ее на кусочки

"Разрезаете" ее на логические блоки и делаете каркас будущего сайта с использованием языка разметки HTML.



**3 шаг**  
Придаете сайту аккуратный вид

Дорабатываете каркас, придаете сайту аккуратный и привлекательный вид с помощью языка стилей CSS.



**4 шаг**  
Добавляете динамики и интерактива

"Оживляете" статичный сайт, добавляете интерактивные элементы с помощью языка JavaScript.



**PROFIT!**

Сделаете проект и получаете деньги. Но главное – удовольствие от хорошо сделанного проекта, который уже лежит в сети.

## Что такое Back-end разработка?

К серверной части (back-end разработке) относится создание ядра сайта, разработка платформы (движка?), основного функционала и административной части (админка).

Back-end разработка включает в себя написание приложений, которые обеспечивают поддержку front-end. На сайте, back-end будет отвечать за проверку данных пользователей из баз данных, и предоставлять им возможности входа, а также принимать какие-либо предложения, которые могут быть полезны в улучшении сайта. Специалисты, ответственные за ведение веб-сайта называются back-end разработчиками. Они играют большую роль в поддержании динамических веб-сайтов, таких как блоги, которые нуждаются в постоянном обновлении на регулярной основе.

## Front-end против Back-end разработки

Бэкэнд - это всё, что работает на сервере, а не на компьютере пользователя, который соединяется с этим сервером. Различные скрипты обрабатывают данные и действия пользователя, а затем посылают результат на компьютер.

Фронтэнд - всё, что "читает" и выполняет браузер на компьютере пользователя. Сюда относят как простую сборку в html-страницы, которая приходят к вам с сервера по запросу в виде кода, так и простейшую анимацию, несложные подсчеты в онлайн-формах.

### Языки программирования

- Front-end разработчик должен владеть такими языками как HTML, CSS (таблицы стилей) и JavaScript.
- Back-end разработчик должен владеть PHP, Java, Python, Ruby и .NET; MySQL, PostgreSQL (произносится «ПостГр эС-Кью-Эл»).

### Среда разработки

- Front-end разработчики работают через пользовательский интерфейс для разработки внешнего вида сайта, изучают привычки пользователей и изменение дизайна с помощью тестирования.
- Back-end разработчики работают с архитектурой кода, разрабатывают приложения, которые поддерживают пользовательский интерфейс, а также обеспечивают поддержку, безопасность и управление контентом.

### **Автономное обслуживание**

- Front-end разработка не может осуществляться самостоятельно, за исключением создания дизайна сайта, который может продаваться в виде шаблонов.
- Back-end разработка может осуществляться самостоятельно в качестве независимого сервиса.

### **Цель**

- Цель front-end команды состоит в слежении за тем, что сайт доступен для всех пользователей, и реагирует на все представления – для мобильных и настольных компьютеров.
- Back-end команда нужна, чтобы создавать приложения для внешнего интерфейса и поддерживать его. Кроме того, они заботятся о том, чтобы сайт открывался и все функции правильно работали.

Back-end – выполняется на стороне сервера, в то время как front-end работает на стороне клиента. Такая структура значительно ускоряет процесс создания сайта, а также облегчает его обслуживание.

### **(FrontEnd + BackEnd) разработчик = FullStack Developer**

Кроме обязанностей, связанных с дизайном, Веб-мастеру нередко приходится также выполнять обязанности системного администратора и программиста (по крайней мере, иметь хорошее представление о сетевых технологиях и языках программирования).

Профессиональные Web-дизайнеры разрабатывают коммерческие проекты, приносящие прибыль: виртуальные магазины, поисковые системы, каталоги, информационные сайты. Профессионально разработанный Web-сайт является одним из видов эффективной рекламы продукции или услуг, удобным средством для поиска деловых клиентов и партнеров, успешного проведения маркетинговой политики.

Основной задачей профессиональных Web-дизайнеров является привлечение большого числа посетителей на созданный ими сайт. И если на страницы дизайнеров-любителей заходят два-три посетителя в неделю, то коммерческие сайты посещают ежедневно тысячи людей. Этому способствует неразрывное единство художественного оформления сайта и его предметного содержания (единство формы и содержания). Не последнюю роль в создании популярности играет умение "раскрутить" сайт (SEO – поисковая оптимизация).

На Web-страницах присутствуют не только текст и статические изображения, но и анимация, видео- и аудиоклипы. Внешний вид просматриваемых Web-страниц динамично изменяется при выполнении пользователем некоторых действий.

При этом опытный профессиональный разработчик хорошо представляет контингент людей, чаще других посещающих его сайты, учитывает психологию пользователей, подбирает темы, которые наиболее популярны в сети в данный момент времени. Подавляющее большинство Web-страниц является продуктом мультимедиа.

Проектирование страниц или сайтов осложняется еще и тем, что одна и та же страница может отображаться не одинаково разными браузерами (Chrome, FireFox, Safari, Opera, IE и Edge) и в окнах разного размера. Во многих случаях браузер самостоятельно "решает" как будет выглядеть загружаемая страница. Изменить внешний вид страницы может и пользователь, например, путем отключения опции загрузки рисунков, анимации.

При проектировании сайта дизайнеру приходится обращать внимание на правильный выбор цвета шрифта и цвета фона. Рекомендуются отдавать предпочтение синим буквам на белом фоне, черным на желтом, зеленым на белом, черным на белом, белым на синем. При выборе цветовой палитры следует учитывать, что наибольшую чувствительность глаз имеет к желто-зеленому цвету, а наименьшую к фиолетовому и красному цветам. Замечено, что разные цвета оказывают разное воздействие на человека: красный цвет возбуждает, а синий цвет успокаивает.

Данные психологических исследований говорят о том, что человеческий глаз воспринимает красный, оранжевый, желтый цвета (и их оттенки) как теплые. Синий, фиолетовый цвета и их оттенки воспринимаются человеком как холодные. Как правило, теплые тона на рисунке обладают свойством "приближаться", а холодные - "удаляться". Другими словами, предметы, нарисованные теплыми тонами, будут казаться расположенными ближе предметам, изображенных холодными тонами.

Специалистами художественной фотографии установлено, что глаз зрителя обращается прежде всего к светлым тонам снимка и к тем местам, где имеется наивысший тональный контраст.

Белый, черный и серый цвета называются ахроматическими, все прочие - хроматическими. Особенности воздействия цвета на человека изучает наука колористика.

Перечислим некоторые важные рекомендации по созданию Web-сайта. Целесообразно не перегружать страницу большим числом деталей. Это связано с тем, что человек за короткое время способен воспринять и запомнить не более 7-8 объектов.

Считается, что чем меньше размер объекта, тем больше должна быть его контрастность. Замечено, что четко выраженный фоновый рисунок на Web-странице повышает утомляемость глаз и снижает эффективность восприятия текста. Для исключения пестроты страницу рекомендуется создавать с использованием не более четырех основных цветов.

На всех страницах одного сайта рекомендуется использовать однотипные панели навигации, размещенные в одинаковых местах. Каждая Web-страница сайта должна иметь ссылку на главную страницу сайта.

На последней стадии проектирования сайта целесообразно проверить работоспособность всех сделанных гиперссылок. Завершая проектирование сайта, желательно просмотреть его с помощью различных браузеров (Chrome, FireFox, Opera, Safari, IE/Edge).



### Краткие итоги

- Web-дизайн – новый жанр изобразительного искусства, использующий собственные средства, закономерности и приемы.
- Web-дизайнер применяет специфические инструменты: компьютер, прикладные программы и языки программирования.

### Определения в области web-дизайна

**Web-дизайнер** - это специалист, который занимается проектированием и разработкой web-узла; созданием его структуры, подбором цветовой палитры и разработкой пользовательского интерфейса.

**Web-мастер** - это специалист, основной задачей которого является поддержка существующего web-сайта. Поддержка заключается в информационном наполнении страниц сайта и в виртуальном взаимодействии с пользователями интернета, которые посещают этот сайт.

**Web-сайт** - это совокупность web-страниц, объединенных между собой единой композицией, темой информационного содержания, структурой, набором цветов, и находящихся на одном или нескольких серверах.

**Web-страница** - это текстовый документ, составленный с использованием языка HTML-разметки. Web-страница является простым файлом с расширением html или htm. Этот файл содержит текст и изображения.

**Браузер** - это прикладная программа, которая позволяет запрашивать, получать с сервера и отображать web-страницы. Самым популярным браузером в настоящее время (2017 г.) считается Google Chrome.

**Статический web-узел** - это web-сайт, содержимое которого фиксировано и не изменяется до тех пор, пока мастер не заменит данные на сайте.

**Динамический web-узел** - это web-сайт, в котором информация хранится в базе данных, а web-страницы генерируются динамически в ответ на запрос клиента.

**Сервер** - это компьютер, подключенный к сети и предназначенный для управления ресурсами. Если он подключен к глобальной сети интернета, то на нем могут быть размещены web-документы, которые передаются в ответ на запросы клиентов.

**Сетевой робот** - это компьютерная программа, которая сканирует различные web-сайты, извлекает информацию о ключевых словах и сохраняет ее в базе данных поискового сервера.

**Поисковый сервер** - это компьютерная программа, которая организует поиск по ключевым словам в глобальной сети интернета, либо на конкретном web-узле. В ответ на запрос пользователя поисковый сервер возвращает список ссылок на документы. В качестве примера поискового сервера можно назвать "yandex" и "rambler".

**Доменное имя** - это название web-сайта, представленное в виде символов [чаще на английском языке]. Например, [www.designer.ru](http://www.designer.ru), [www.site.com](http://www.site.com) и другие.

**Протокол интернет (IP)** - это протокол передачи информационных пакетов, который лежит в основе функционирования глобальной сети интернета.

**Web-палитра** - это набор из 216 цветов. Web-ориентированная палитра гарантирует, что цветное изображение будет отображаться в неизменном виде большинством браузеров, работающих на разных платформах.

**Протокол** - это набор правил, регламентирующих обмен информацией между компьютерами в сети. В качестве примеров сетевых протоколов можно привести IP, HTTP и FTP.

**Гипертекст** - это информационная структура, которая обеспечивает навигацию между web-документами посредством гипертекстовых ссылок. Фрагменты текста или изображения, при активации которых отображаются связанные с ними документы, называются гипертекстовыми ссылками.

Вопросы для закрепления.

- 1) Что такое «web-дизайн»?
- 2) Дайте понятие «web-страница»
- 3) Назовите допустимый объем web-страницы.
- 4) Назовите основные рекомендации при создании web-страниц

Домашнее задание: Подготовить доклад на тему: «История развития сети Интернет», «Основные web-технологии»

**Тема 2****Тема: Структура HTML документа**

**Цели:** *Ознакомить учащихся с основными понятиями HTML; дать понятие основных элементов HTML; определить структуру HTML документа.*

План урока.

1. Организационный момент.
2. Изложение нового материала.
  - 1) Язык HTML
  - 2) Структура HTML документа
  - 3) Спецификации элементов
3. Закрепление изученного.
4. Итог урока.

**ОСНОВНЫЕ ПОНЯТИЯ HTML**

**HTML** – язык гипертекстовой разметки документов (HyperText Markup Language). Появился благодаря ученому, которого зовут Тим Бернерс-Ли.



Тимоти Джон Бернерс-Ли

Тим – британский учёный, изобретатель URL, HTTP, HTML, создатель Всемирной паутины (совместно с Робертом Кайо) и действующий глава Консорциума Всемирной паутины.

Появление первых версий HTML относят к 1986 году, а в 1991 году HTML был существенно доработан и стал использоваться именно для передачи гипертекста по просторам всемирной паутины. Известны стандартизованные версии HTML2, HTML3.2 и HTML4.01. Этот курс лекций ориентирован на классическую версию HTML4.01 (HTML5, CSS3 и XHTML1.1. не рассматриваются).

HTML – это не язык программирования в традиционном смысле, он является языком разметки. С помощью HTML текстовый документ разбивают на блоки смысловой информации (заголовки, параграфы, таблицы, рисунки и т.п.).

**Гипертекстовый документ** – это документ, содержащий переходы (**гиперссылки**) на другие документы. При использовании гиперссылки происходит перемещение от одного документа к другому (как по цепочке) в Интернете. HTML-документ является гипертекстовым документом.

Особенности HTML-документа:

1. HTML-документ может содержать текст, графику, видео и звук.
2. В общем случае HTML-документ – это один или несколько текстовых файлов, имеющих расширение .htm или .html.
3. Создавать HTML-документ можно как с помощью специальных программ – редакторов HTML (например: Adobe Dreamweaver, Web Builder или Microsoft SharePoint Designer), так и с помощью текстового редактора (например, Блокнота++).
4. Для просмотра HTML-документов существуют специальные программы – **браузеры**. Они **интерпретируют** HTML-документы, т.е. переводят текст документа в Web-страницу, и отображают ее на экране пользователя.
5. Если при интерпретации HTML-документа браузер чего-то не понимает, то сообщения об ошибке не возникает, а это место в HTML-документе игнорируется и не отображается браузером.

HTML-документ состоит из элементов HTML.

**Элемент HTML** – это чаще всего два тега (открывающий и закрывающий) и часть документа между ними. Кроме того, существуют элементы HTML, состоящие только из одного тега.

**Тег** – в переводе с английского – ярлык, этикетка. Тег определяет тип выводимого элемента HTML (например, заголовков, таблица, рисунок и т.п.).

Сам тег не отображается браузером. Тег представляет собой последовательность элементов:

- символ левой угловой скобки (<) – начало тега;
- необязательный символ слеша (/) – символ используется, чтобы обозначить закрывающий тег;
- имя тега;
- необязательные атрибуты в открывающем теге;
- символ правой угловой скобки (>)

**Атрибуты** – необязательный набор параметров, определяющих дополнительные свойства элемента HTML (например, цвет или размер). Атрибут состоит:

- из имени атрибута;
- знака равенства (=);
- значения атрибута – строки символов, заключенной в "кавычки"

#### Пример элемента HTML:

```
<H1 ALIGN= "CENTER">Глава 1</H1> [ə'lain]
```

В этом примере:

`<H1 ALIGN= "CENTER">` – открывающий тег

`</H1>` – закрывающий тег

`H1` – имя тега

`ALIGN= "CENTER"` – атрибут

`ALIGN` – имя атрибута

`"CENTER"` – значение атрибута

#### Правила создания HTML-документов:

1. Теги и атрибуты можно записывать в любом регистре, т.е. `</H1>` и `</h1>` – это одно и то же.

2. Несколько пробелов подряд, символы табуляции и перевода строки при интерпретации браузером заменяются на один пробел. Это позволяет писать хорошо структурированные исходные тексты файлов HTML.

3. Рекомендуется давать имена файлам \*.html строчными английскими буквами. Длина имени – до восьми символов. В принципе, можно не придерживаться данной рекомендации, но тогда пользователи, работающие в операционных системах, отличных от Windows, не смогут воспользоваться вашими HTML-документами.

## СТРУКТУРА HTML-ДОКУМЕНТА

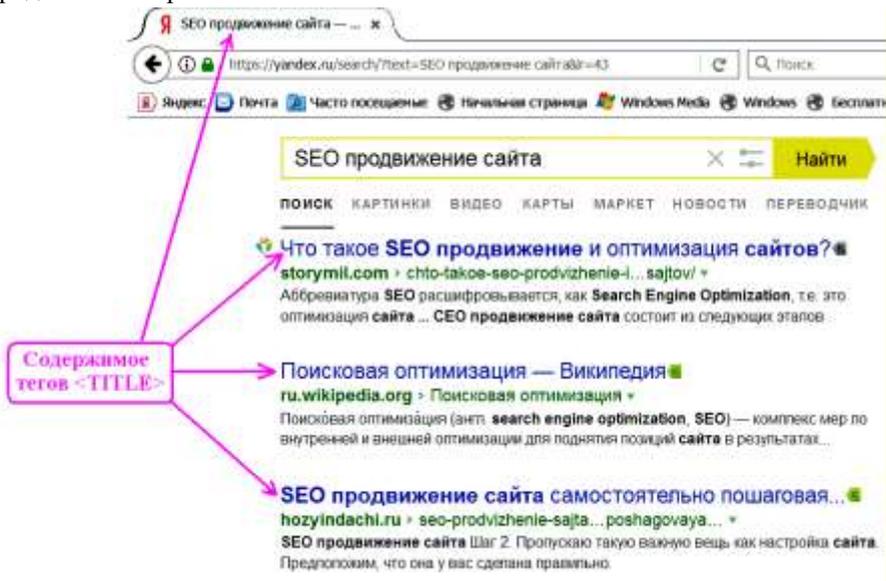
Каждый HTML-документ должен начинаться тегом `<HTML>` и заканчиваться тегом `</HTML>`. Эти теги обозначают, что находящиеся между ними строки представляют единый HTML-документ. Кроме того, можно заметить, что файл HTML в целом является элементом языка HTML.

Также в HTML-документе должны присутствовать элементы HEAD (информация о документе) и BODY (тело документа).

Для вставки комментариев используется пара тегов `<!-- -->`

### Раздел документа HEAD

Мета-теги располагаются в заголовке html-страницы, между тегами `<head>` и `</head>`, и никак не отображаются браузером (за исключением содержимого тега `<title>`). Тем не менее, мета-теги для сайта играют очень важную роль. Их добавление и оптимизация могут существенно помочь в продвижении и работе сайта.



Рекомендую использовать только те мета-теги, которые реально нужны для работы сайта. Не следует загромождать область «head» лишними инструкциями, т.к. любая ошибка здесь может привести к печальным последствиям.

Раздел документа HEAD определяет его заголовок, а также содержит дополнительную информацию о документе для браузера. Этот раздел не

является обязательным, однако рекомендуется всегда использовать его в своих HTML-документах, так как правильно составленный заголовок может быть весьма полезен.

Раздел заголовка начинается тегом **<HEAD>** и следует сразу за тегом **<HTML>**. Между открывающим и закрывающим тегами элемента HEAD располагаются другие элементы заголовка.

### Название документа TITLE

Для того чтобы дать название HTML-документу, предназначен тег **<TITLE>**. Это название будет выведено в заголовок окна браузера. Название записывается между тегами **<TITLE>** и **</TITLE>** и представляет собой строку текста. Длина этой строки может быть любой, но рекомендуется делать ее не больше 60 символов. Элемент TITLE должен находиться только в разделе HEAD.

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Приветствие</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

.....

```
</BODY>
```

```
</HTML>
```

Кроме заголовка, в разделе HEAD могут присутствовать другие теги:

- Мета-тег «keywords» (ключевые слова)
- Мета-тег «description» (описание страницы)

Важной особенностью мета-тега «description» является то обстоятельство, что в случае его правильного заполнения поисковые системы на его основе формируют **сниппет** страницы в выдаче (краткое описание под ссылкой на страницу).

- Технические мета-теги
  - content-type – тип документа и его кодировка (windows-1251 или utf-8)
  - content-language – указание языка документа (ru)

```

<HTML>
<HEAD>
  <TITLE>Приветствие</TITLE>
  <meta http-equiv="Content-Type" content="text/html; ch
  <meta http-equiv="content-language" content="ru">
</HEAD>
<BODY> Текст страницы </BODY>
</HTML>

```

```

1 <head>
2   <meta charset="UTF-8">
3   <meta name="description" content="Web-портал">
4   <meta name="keywords"
5 content="HTML,CSS,XML,JavaScript">
6   <meta name="author" content="Автор Я">
7 </head>
8 <body>
9   Содержимое страницы
10 </body>

```

### Раздел документа BODY

В этом разделе документа располагается та информация, которая отображается в окне браузера. Раздел BODY должен начинаться тегом **<BODY>** и завершаться тегом **</BODY>**, между которыми располагаются элементы HTML, из которых и состоит содержание документа.

### Спецификация элемента BODY

Тег **<BODY>** имеет ряд атрибутов, определяющих внешний вид документа. Ниже приводится спецификация тега **<BODY>**.

```

<BODY
  TEXT="цвет текста"
  BGCOLOR="цвет фона"
  BACKGROUND="адрес фонового рисунка"
  LINK="цвет непосещенной гиперссылки" (по умолчанию синий)
  VLINK="цвет посещенной гиперссылки" (по умолчанию фиолетовый)
  ALINK="цвет активной гиперссылки" (по умолчанию красный)
>

```

Атрибут `TEXT` задает цвет шрифта для всего документа в формате RGB или в символьной нотации. По умолчанию (если не указан этот атрибут) используются настройки браузера.

Атрибут `BGCOLOR` задает цвет фона окна браузера документа в формате RGB или в символьной нотации. По умолчанию используются настройки браузера.

Атрибут `BACKGROUND` позволяет указать адрес и имя рисунка, используемого в качестве фона. Этот рисунок будет размножен и распределен на заднем плане документа.

Атрибуты `LINK`, `VLINK` и `ALINK` задают цвета гиперссылок в формате RGB или в символьной нотации. По умолчанию используются настройки браузера. Непосещенная гиперссылка – гиперссылка, которая еще не использовалась для перехода к другому документу. Посещенная гиперссылка – гиперссылка, которая уже использовалась для перехода к другому документу. Активная гиперссылка – гиперссылка на документ, к которому в данный момент происходит переход.

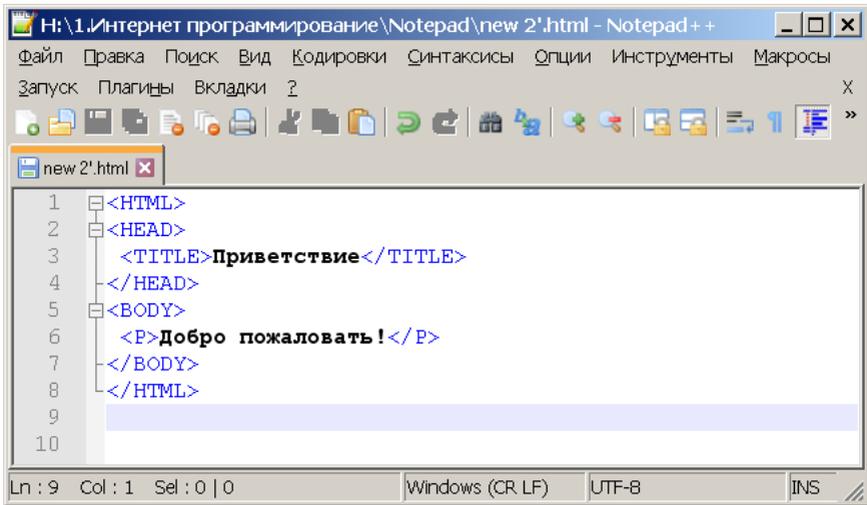
#### Советы по использованию атрибутов тега `BODY`

□ Если вы указываете хотя бы один цвет в теге `BODY`, то укажите и остальные. Это связано с тем, что пользователь может установить настройки цветов своего браузера как ему удобней. Указание только одного цвета может привести к ситуации, что у некоторых пользователей текст сольется с цветом фона. В результате просмотр документа будет затруднен.

□ Выбирайте цвет текста так, чтобы он "работал" вместе с цветом фона или основными цветами изображения. Например, красное на зеленом может вызвать серьезные проблемы у значительного числа людей.

□ В элементе `BODY` можно задать как атрибут `BGCOLOR`, так и атрибут `BACKGROUND`. В этом случае браузер отдает предпочтение `BACKGROUND`, но если изображение фона невозможно загрузить, будет использовано `BGCOLOR`. Поэтому старайтесь задавать цвет фона похожим на цвет фонового рисунка, чтобы не нарушился цветовой баланс документа.

#### **Пример простого HTML-документа**



The screenshot shows a Notepad++ window with the following content:

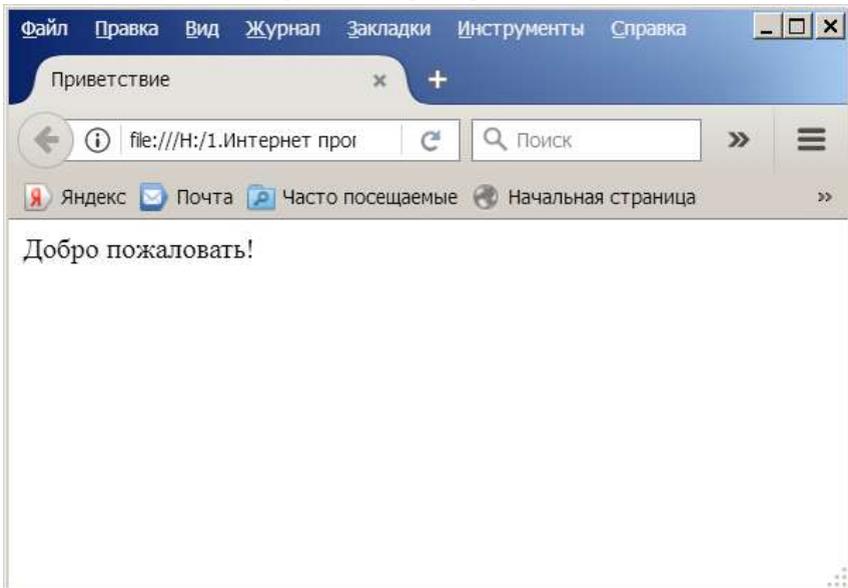
```

1  <HTML>
2  <HEAD>
3    <TITLE>Приветствие</TITLE>
4  </HEAD>
5  <BODY>
6    <P>Добро пожаловать!</P>
7  </BODY>
8  </HTML>
9
10

```

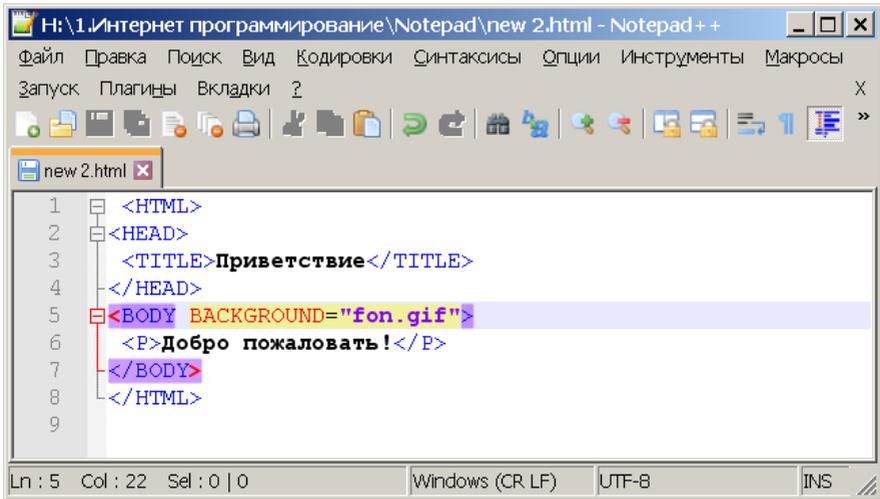
The status bar at the bottom indicates: Ln : 9 Col : 1 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

Этот документ отобразится в браузере так:



В этом примере обратите внимание на то, куда выводится браузером название документа в элементе TITLE.

### Пример использования фонового рисунка



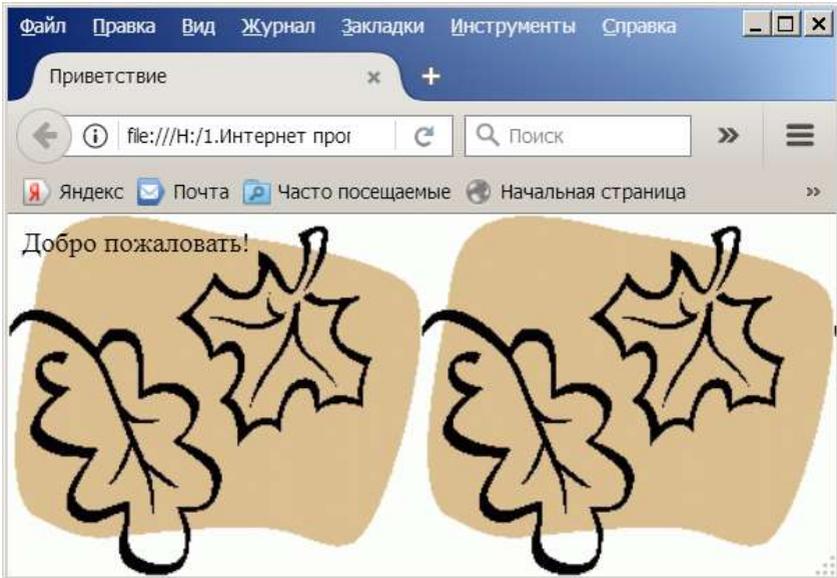
```

1  <HTML>
2  <HEAD>
3    <TITLE>Приветствие</TITLE>
4  </HEAD>
5  <BODY BACKGROUND="fon.gif">
6    <P>Добро пожаловать!</P>
7  </BODY>
8  </HTML>
9

```

Ln : 5 Col : 22 Sel : 0 | 0 Windows (CR LF) UTF-8 INS

Этот документ отобразится в браузере так:



Вопросы для закрепления

- 1) Дайте понятия элемента HTML, тега, атрибутов.
- 2) Что такое браузер?
- 3) Каковы особенности и правила HTML-документа?
- 4) Какие элементы HTML входят в обязательную структуру HTML-документа?
- 5) Для чего используется элемент BODY, и какие он имеет атрибуты?

Домашнее задание: Создать две web- страницы в редакторе Блокнот++, используя язык HTML, с информацией о себе.

### Тема 3

#### Тема: Цветовые спецификации. Графические элементы

**Цели:** *Ознакомить учащихся с основными цветовыми спецификациями; использование цветовых обозначений с помощью символьной нотации.*

План урока.

1. Организационный момент.
2. Изложение нового материала.
  - 1) Цветовые спецификации
  - 2) Символьная нотация
  - 3) Добавление графических объектов с помощью языка HTML
3. Закрепление изученного.
4. Итог урока.

Ход урока.

#### **Организационный момент.**

Отметить присутствующих. Сообщить тему урока. Настроить учащихся на восприятие материала.

#### **Изложение нового материала.**

### **ЦВЕТОВЫЕ СПЕЦИФИКАЦИИ**

Для определения цвета различных элементов HTML-документа необходимо указать значение соответствующих атрибутов. Указывать значения этих атрибутов можно двумя способами:

- определять цвет в формате RGB;
- определять цвет, используя символьную нотацию

#### **Формат RGB**

Формат RGB – это система указания цвета, которая базируется на смешении трех основных цветов: красном (RED), зеленом (GREEN) и синем (BLUE). Итоговый цвет определяется цифрами в шестнадцатеричном коде. Для каждого цвета задается шестнадцатеричное значение в пределах от 0 до FF, что соответствует диапазону 0-255 в десятичном исчислении. Затем эти значения объединяются в одно число, перед которым ставится символ #. Например, число #800080 обозначает фиолетовый цвет. Указывая цвет в формате RGB, можно определить более шестнадцати миллионов цветовых оттенков.

### **СИМВОЛЬНАЯ НОТАЦИЯ**

Задание цвета в формате RGB имеет один недостаток – необходимо помнить совокупности цифр для указания цвета. Этого недостатка лишена символьная нотация. Можно указывать название цвета на английском языке. Но у этого способа указания цвета тоже есть недостаток – определено всего шестнадцать имен цветов.

### СООТВЕТСТВИЕ ФОРМАТА RGB И СИМВОЛЬНОЙ НОТАЦИИ

Ниже приведена таблица соответствий указания цвета в символьной нотации и формате RGB.

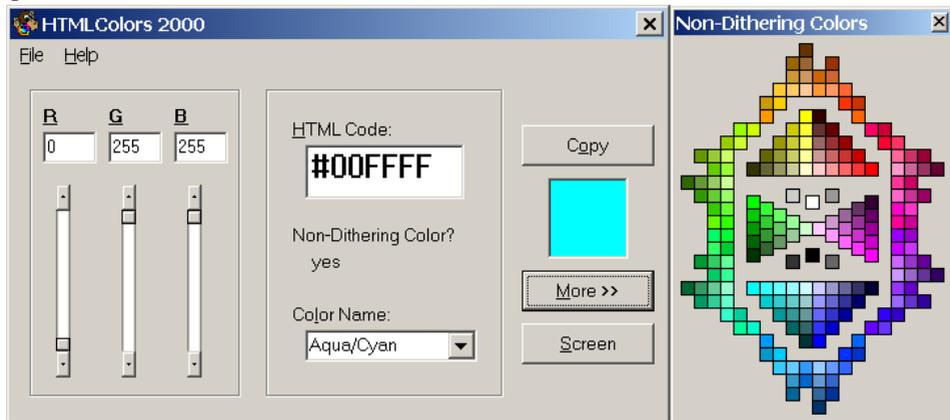
Символьная нотация	Формат RGB	Цвет
Black	#000000	Черный
Silver	#C0C0C0	Серебро
Gray	#808080	Серый
White	#FFFFFF	Белый
Maroon	#800000	Бордовый
Red	#FF0000	Красный
Purple	#800080	Фиолетовый
Fuchsia	#FF00FF	Розовый
Green	#008000	Зеленый
Lime	#00FF00	Желто-зеленый
Olive	#808000	Оливковый
Yellow	#FFFF00	Лимонный
Navy	#000080	Темно-синий
Blue	#0000FF	Синий
Teal	#008080	Бирюзовый
Aqua	#00FFFF	Аквамарин

Таким образом, строка `TEXT="#008000"` и строка `TEXT="Green"` в теге `<BODY>` одинаково определяют цвет шрифта – зеленый.

В эпоху 256-цветных видеокарт широко применялась так называемая «безопасная палитра Netscape», состоящая из всех возможных сочетаний 6 оттенков красного, зеленого и синего. Это однородная палитра  $6 \times 6 \times 6 = 216$  оттенков. В truecolor у безопасных цветов все три компонента – R, G и B – равняются 00, 33, 66, 99, CC и FF (0, 51, 102, 153, 204, 255 с шагом 20%).

Браузер Netscape Navigator гарантировал, что эти цвета будут отображены без приближения или растривания (без имитации полутонов).

Другими словами, 216 цветов стандартной 16-цветовой палитры и палитры 6×6×6 были наиболее предпочтительными для цвета изображения и фона.



К концу 90-х годов, с исчезновением 8-битного цвета с рабочих столов ОС, это требование исчезло. Тем не менее, стоит помнить, что популярный графический формат GIF имеет палитру в  $2^8=256$  цветов. А редактор Adobe Photoshop и поныне оснащён опцией «Только Web-цвета» (англ. Only Web Colors).

## ГРАФИЧЕСКИЕ ЭЛЕМЕНТЫ

Одним из достоинств HTML-документа является возможность использования графических элементов в оформлении. Можно выделить три элемента, чаще всего используемых в HTML-документах: горизонтальные линии, таблицы и рисунки.

### ГОРИЗОНТАЛЬНЫЕ ЛИНИИ

Горизонтальные линии визуально подчеркивают законченность той или иной области документа. Сейчас часто используют рельефную, вдавленную линию, чтобы обозначить “объемность” документа.

Тег **<HR>** позволяет провести рельефную горизонтальную линию в окне большинства браузеров. Этот тег не является контейнером, поэтому не

требует закрывающего тега. До и после линии автоматически вставляется пустая строка. Спецификация тега <HR>:

```
<HR
ALIGN="выравнивание" [ə'lain]
WIDTH="длина линии" [widθ]
SIZE="толщина линии"
NOSHADA >
```

Атрибут ALIGN определяет способ выравнивания линии. Он может иметь следующие значения:

- LEFT – линия выравнивается по левому краю окна браузера. Это значение используется по умолчанию.
- CENTER – линия выравнивается по центру окна браузера.
- RIGHT – линия выравнивается по правому краю окна браузера.
- – выравнивание по ширине (аналогично тексту) для линии не предусмотрено.

Атрибут WIDTH задает длину линии. Значением данного атрибута является число. Это число означает длину линии в пикселях. Если после числа стоит знак %, то это означает длину в процентах от ширины окна. Например:

```
<HR WIDTH="400"> – линия длиной 400 пикселей.
```

```
<HR WIDTH="50%"> – линия длиной 50 процентов от ширины окна.
```

Атрибут SIZE задает толщину линии. Значением этого атрибута является число. Это число означает толщину линии в пикселях.

Атрибут NOSHADA отменяет “трехмерность” линии.

## РИСУНКИ

Без иллюстраций документ скучен, вял и однообразен. HTML позволяет использовать рисунки в формате JPG и GIF для оформления HTML-документов. Для вставки рисунков используется тег <IMG>. Спецификация тега <IMG>:

```
<IMG
SRC="адрес рисунка"
ALIGN="выравнивание"
HEIGHT="высота рисунка"
WIDTH="ширина рисунка"
BORDER="толщина рамки"
>
```

Атрибут SRC определяет URL-адрес рисунка, который будет отображаться браузером.

Атрибут ALIGN определяет способ выравнивания рисунка. Он может иметь следующие значения:

- TOP – рисунок выравнивается по верхнему краю текущей строки.
- MIDDLE – рисунок выравнивается серединой по текущей строке.
- BOTTOM – рисунок выравнивается по нижнему краю текущей строки.
- LEFT – рисунок прижимается к левому краю окна браузера и обтекается текстом.
- RIGHT – рисунок прижимается к правому краю окна браузера и обтекается текстом.

Атрибут HEIGHT определяет высоту рисунка в пикселях.

Атрибут WIDTH определяет ширину рисунка в пикселях.

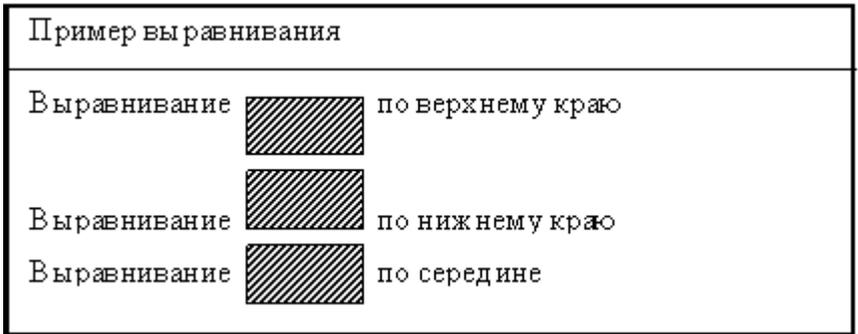
Используя атрибуты HEIGHT и WIDTH можно увеличивать или уменьшать рисунок. Если указать только один из этих атрибутов, то рисунок будет увеличен или уменьшен пропорционально и по ширине, и по высоте.

Атрибут BORDER позволяет задавать рамку вокруг рисунка. Значение этого атрибута – толщина рамки в пикселях. По умолчанию = 1.

Пример выравнивания рисунков:

```
<HTML>
<HEAD>
<TITLE>Пример выравнивания</TITLE>
</HEAD>
<BODY>
<P>Выравнивание<IMG SRC="ris1.jpg" ALIGN="TOP">по
верхнему краю</P>
<P>Выравнивание<IMG SRC="ris1.jpg" ALIGN="BOTTOM">
по нижнему краю</P>
<P>Выравнивание<IMG SRC="ris1.jpg" ALIGN="MIDDLE">
по середине</P>
</BODY>
</HTML>
```

Браузер отобразит данный пример так:



### Рисунок-ссылка

Можно использовать рисунки в качестве гиперссылок. Для этого нужно поместить изображение (тег IMG) между тегами <A> и </A>. Например:

```
<A href="sample.html">
```

```

```

```
</A>
```

Вопросы для закрепления.

- 5) Чем отличается символьная нотация от формата RGB?
- 6) Какие значения в символьной нотации можно использовать для указания цвета?

Домашнее задание: Подготовить доклад на тему: «Дополнительные web-технологии (Java, CGI, SSI, CSS, PHP, ASP, VBScript и др.)»

## Тема 4

### Тема: Ввод текстовой информации

**Цели:** *Ознакомить учащихся с основными элементами форматирования текста: заголовок, абзац, перевод строки, списки, форматирование шрифта; способ создания таблиц с помощью языка HTML.*

План урока.

1. Организационный момент.
2. Изложение нового материала.
  - 1) форматирование текста
  - 2) форматирование шрифта
  - 3) создание таблиц
3. Закрепление изученного.
4. Итог урока.

Любые тексты в HTML4 имеют определенную структуру. Элементами такой структуры являются заголовки, подзаголовки, абзацы, списки и др.

Разбиение всего текста на структурные элементы называется логическим форматированием. В HTML-документе логическое форматирование достигается с помощью специальных тегов.

#### АБЗАЦЫ

Одним из первых правил составления любых документов является разбиение его текста на отдельные абзацы, выражающие законченную мысль. В HTML-документе разделение на абзацы производится с помощью специального тега абзаца `<P>` или тега `<DIV>`. Отличие между абзацем и контейнером `<div>` в том, что в начале и конце параграфа (тег `<p>`) появляется вертикальный отступ, чего нет в случае использования тега `<div>`.

Атрибуты тегов аналогичны.

Синтаксис тега `<P>` таков:

```
<P  
ALIGN="выравнивание"> /лайн/
```

Атрибут ALIGN определяет способ выравнивания абзаца. Он может иметь следующие значения:

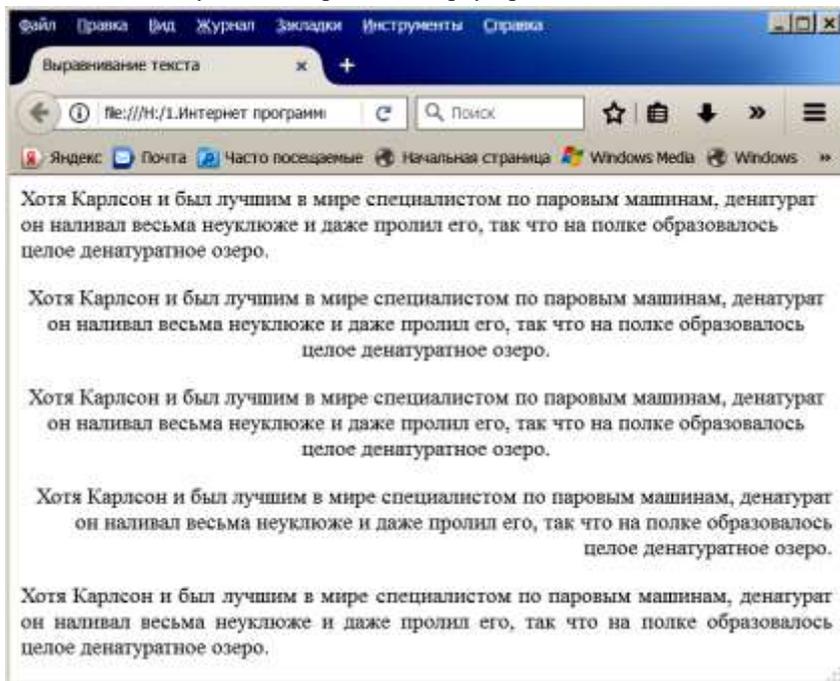
- LEFT – текст выравнивается по левому краю окна браузера. Это значение используется по умолчанию, т.е. когда атрибут не указан.
- CENTER – текст выравнивается по центру окна браузера.
- RIGHT – текст выравнивается по правому краю окна браузера.

□ JUSTIFY – текст выравнивается по ширине.

Пример использования тега <P>:

```
<HTML>
<HEAD>
  <TITLE>Выравнивание текста</TITLE>
</HEAD>
<BODY>
  <P align="LEFT">Хотя Карлсон и был лучшим в мире специалистом по паровым машинам, денатурат он наливал весьма неуклюже и даже пролил его, так что на полке образовалось целое денатуратное озеро.</P>
  <P ALIGN="CENTER">Хотя Карлсон и был лучшим в мире специалистом по паровым машинам, денатурат он наливал весьма неуклюже и даже пролил его, так что на полке образовалось целое денатуратное озеро.</P>
  <CENTER>Хотя Карлсон и был лучшим в мире специалистом по паровым машинам, денатурат он наливал весьма неуклюже и даже пролил его, так что на полке образовалось целое денатуратное озеро.</CENTER>
  <P ALIGN="RIGHT">Хотя Карлсон и был лучшим в мире специалистом по паровым машинам, денатурат он наливал весьма неуклюже и даже пролил его, так что на полке образовалось целое денатуратное озеро.</P>
  <P ALIGN="JUSTIFY">Хотя Карлсон и был лучшим в мире специалистом по паровым машинам, денатурат он наливал весьма неуклюже и даже пролил его, так что на полке образовалось целое денатуратное озеро.</P>
</BODY>
</HTML>
```

Этот документ отобразится в браузере так:



Кроме тегов <P> и <DIV>, выровнять по центру можно и так:

`<center>Добро пожаловать!</center>` (третий способ центрирования)

Браузер автоматически формирует абзацы в зависимости от ширины окна браузера или размера шрифта, перенося слова из строки в строку и отделяя абзацы друг от друга пустой строкой.

### УПРАВЛЕНИЕ ПЕРЕВОДОМ СТРОКИ

Так как браузер автоматически определяет места переноса строк, иногда возникают ситуации запретить перевод строки в каком-нибудь месте или, наоборот, принудительно сделать перевод строки в каком-то определенном месте. Для этого существуют особые теги, управляющие переводом строк.

Когда необходимо сделать принудительный перевод строки, используют тег <BR>. Этот тег не имеет атрибутов и закрывающего тега. Пример использования принудительного перевода строки:

```
<HTML>
<HEAD>
<TITLE>Приветствие</TITLE>
</HEAD>
<BODY>
<P>Добро<BR>позаловать!</P>
</BODY>
</HTML>
```

Этот пример будет выглядеть так:

Приветствие
Добро позаловать!

При использовании тега <BR> пустая строка не образуется, т.е. абзац не прерывается.

В некоторых случаях, наоборот, бывает необходимо сделать так, чтобы браузер не производил перевода строки. Например, не рекомендуется отрывать

буквы инициалов от фамилии. В таких случаях тот участок текста, в котором нельзя переводить строку, следует поместить в элемент **NOBR** (тег понимается всеми браузерами, хотя не входит в спецификацию HTML и его наличие приводит к невалидному коду).

Пример:

```
<P>Это стихотворение написал <NOBR>А.С.
Пушкин</NOBR> – великий русский поэт.</P>
```

В браузере участок текста “А.С. Пушкин” всегда будет отображаться на одной строке.

Если получится так, что строка, расположенная в элементе NOBR, будет выходить за пределы окна браузера, то внизу окна появится горизонтальная полоса прокрутки.

Другой вариант – использовать специальный символ неразрывного пробела **&nbsp;**;

Пример:

```
<P> Это стихотворение написал А.С. &nbsp; Пушкин –
великий русский поэт.</P>
```

Кстати, если надо поставить несколько пробелов подряд, то ничего из этого не получится. Пробелы, количество которых превышает одного, автоматически сокращаются браузером, давайте рассмотрим пример такого сокращения:

```
<p>Здесь какой-то текст и много пробелов</p>
```

После открытия мы получаем строку, в которой нет больше одного пробела. Как побороть это? Есть специальный парный тег, который сохраняет форматирование так, как нам требуется. Этим тегом является **<pre>...текст...</pre>**.

Другой способ – использовать рассмотренный выше спецсимвол **&nbsp;**;

```
<p>Немного текста &nbsp; &nbsp; &nbsp; &nbsp;
&nbsp; &nbsp; много &nbsp; &nbsp; &nbsp; &nbsp; &nbsp;
пробелов</p>
```

Не забываем в конце ставить точку с запятой «;» иначе пробелы не появятся, а вместо них увидим код пробела.

Бывают случаи, когда обычный пробел оказывается слишком «большим». Тогда на смену ему приходит тонкий пробел. Это пробел, ширина которого составляет четверть кегля используемого шрифта. Обозначается тонкий пробел следующим образом: **&thinsp;**

и используется, по большей части, для разбиения разрядов чисел, например, "15 000 000 долларов" стоит записать так:

```
15&thinsp;000&thinsp;000 долларов
```

## Заголовки

Почти в каждом тексте используются заголовки для отдельных частей документа. Эти заголовки представляют собой фрагменты текста, которые выделяются на экране при отображении страницы браузером.

Для разметки заголовков используются теги `<H1>`, `<H2>`, `<H3>`, `<H4>`, `<H5>` и `<H6>`. Эти теги требуют соответствующего закрывающего тега. Заголовок с номером 1 является самым крупным (заголовок верхнего уровня), а с номером 6 – самым мелким. Теги заголовка нельзя использовать для выделения отдельных слов текста с целью увеличения их размера. При использовании тегов заголовков происходит вставка пустой строки до и после заголовка, поэтому тегов абзаца и перевода строки здесь не требуется.

Синтаксис тегов заголовков:

```
<Hn
```

```
ALIGN="выравнивание">
```

Атрибут ALIGN определяет способ выравнивания заголовка. Он может иметь те же значения, что и аналогичный атрибут у тега абзаца `<P>`.

Пример использования разных заголовков:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Пример</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<H1>Заголовок 1</H1>
```

```
<H2 ALIGN="CENTER">Заголовок 2</H2>
```

```
<P>Простой текст</P>
```

```
</BODY>
```

```
</HTML>
```

Вот так браузер отобразит данный пример:

Пример
<p><b>Заголовок 1</b></p> <p style="text-align: center;"><b>Заголовок 2</b></p> <p>Простой текст</p>

## СПИСКИ

В языке HTML предусмотрен специальный набор тегов для представления информации в виде списков. Списки являются одним из наиболее часто употребляемых форм представления данных как в электронных документах, так и печатных. В языке HTML предусмотрены маркированные, нумерованные списки и списки определений.

### Маркированный список

Этот список еще называется нумерованным или неупорядоченным. В маркированном списке для выделения его элементов используются специальные символы, называемые маркерами списка. Вид маркеров определяется браузером, причем при создании вложенных списков браузеры автоматически разнообразят вид маркеров различного уровня вложенности.

Для создания маркированного списка необходимо использовать тег-контейнер `<UL> </UL>`, внутри которого располагаются все элементы списка. Открывающий и закрывающий теги списка обеспечивают перевод строки до и после списка, отделяя, таким образом, список от основного содержимого документа, поэтому нет необходимости использовать теги абзаца или принудительного перевода строки.

Каждый элемент списка должен начинаться тегом `<LI>` и заканчиваться тегом `</LI>`.

Спецификация элемента UL:

`<UL`

`TYPE="вид маркера">`

Атрибут TYPE задает вид маркера, которым выделяются элементы списка. Он может имеет следующие значения:

□ DISK – маркеры отображаются закрашенными кружочками, это значение используется по умолчанию;

- CIRCLE – маркеры отображаются не закрашенными кружочками;
- SQUARE – маркеры отображаются квадратиками.

Спецификация элемента LI для маркированного списка:

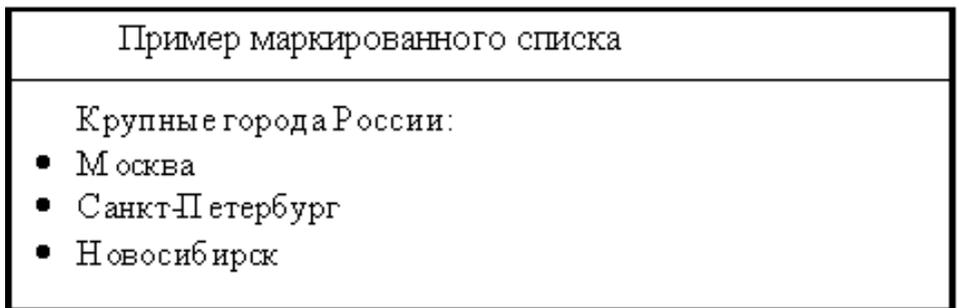
```
<LI
TYPE="вид маркера">
```

Атрибут TYPE задает вид маркера, он может принимать такие же значения, что и одноименный атрибут элемента UL. Значение по умолчанию – DISK.

Пример использования маркированного списка:

```
<HTML>
<HEAD>
<TITLE>Пример маркированного списка</TITLE>
</HEAD>
<BODY>
<UL>
Крупные города России:
<LI>Москва</LI>
<LI>Санкт-Петербург</LI>
<LI>Новосибирск</LI>
</UL>
</BODY>
</HTML>
```

Вот так браузер отобразит данный пример:



### Нумерованный список

Нумерованные списки иногда называют упорядоченными. Списки данного типа представляют собой упорядоченную последовательность отдельных элементов. Отличием от маркированных списков является то, что в

нумерованном списке перед каждым его элементом автоматически проставляется порядковый номер. Вид нумерации зависит от браузера и может задаваться атрибутами тегов списка. В остальном реализация нумерованного списка во многом похожа на реализацию маркированного списка.

Для создания нумерованного списка следует использовать тег-контейнер `<OL>` `</OL>`, внутри которого располагаются все элементы списка.

Каждый элемент списка должен начинаться тегом `<LI>` и заканчиваться тегом `</LI>`.

Спецификация элемента OL:

`<OL`

`TYPE="вид нумерации"`

`START="начальная позиция">`

Атрибут TYPE задает вид нумерации, которой выделяются элементы списка. Он может иметь следующие значения:

A – маркеры в виде прописных латинских букв;

a – маркеры в виде строчных латинских букв;

I- маркеры в виде больших римских цифр;

i- маркеры в виде маленьких римских цифр;

1- маркеры в виде арабских цифр, это значение используется по умолчанию.

Атрибут START определяет позицию, с которой начинается нумерация списка. Используя этот атрибут, можно начать нумерацию, например, с цифры 5 или буквы E, в зависимости от вида нумерации. Значением атрибута START является число, вне зависимости от вида нумерации.

Спецификация элемента LI для нумерованного списка:

`<LI`

`TYPE="вид нумерации"`

`VALUE="номер элемента">`

Атрибут TYPE задает вид нумерации, он может принимать такие же значения, что и одноименный атрибут элемента OL. По умолчанию значение этого атрибута – 1.

Атрибут VALUE позволяет изменить номер данного элемента, при этом изменятся номера и последующих элементов.

Пример использования нумерованного списка:

`<HTML>`

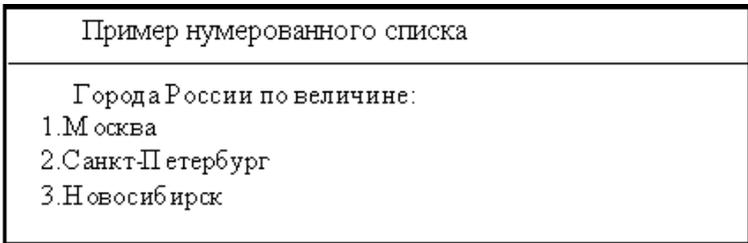
`<HEAD>`

```

<TITLE>Пример нумерованного списка</TITLE>
</HEAD>
<BODY>
<UL>
Города России по величине:
<LI>Москва</LI>
<LI>Санкт-Петербург</LI>
<LI>Новосибирск</LI>
</UL>
</BODY>
</HTML>

```

Вот так браузер отобразит данный пример:



### Список определений

Списки определений, также называемые словарями определений специальных терминов, являются особым видом списков. В отличие от других типов списков каждый элемент списка определений всегда состоит из двух частей. В первой части элемента списка указывается определяемый термин, а во второй части – текст в форме словарной статьи, раскрывающий значение термина.

Списки определений задаются с помощью тега-контейнера **<DL>**. Внутри него тегом **<DT>** отмечается определяемый термин, а тегом **<DD>** – абзац с его определением. Внутри элемента **<DT>** нельзя использовать абзацы (P) и заголовки (H1-H6), но их можно использовать внутри элемента **<DD>**. Атрибутов для элементов списка определений нет.

В общем случае список определений записывается следующим образом:

```

<DL>
<DT>Термин</DT>
<DD>Определение термина</DD>
</DL>

```

Пример использования списка определений:

```
<HTML>
<HEAD>
<TITLE>Пример списка определений</TITLE>
</HEAD>
<BODY>
<DL>
<H2 ALIGN="CENTER">Состав Microsoft Office</H2>
<DT>Microsoft Word</DT>
<DD>Многофункциональный текстовый процессор</DD>
<DT>Microsoft Excel</DT>
<DD>Программа для работы с электронными
таблицами</DD>
<DT>Microsoft Access</DT>
<DD>Система управления базами данных</DD>
</DL>
</BODY>
</HTML>
```

Этот пример отобразится в браузере так:

Пример списка определений	
<b>Состав Microsoft Office</b>	
Microsoft Word	Многофункциональный текстовый процессор
Microsoft Excel	Программа для работы с электронными таблицами
Microsoft Access	Система управления базами данных

## ФОРМАТИРОВАНИЕ ТЕКСТА

В языке HTML предусмотрены специальные теги, предназначенные для форматирования текста. Они позволяют изменять вид шрифта, цвет, размер и др.

Чтобы отобразить текст полужирным шрифтом, используют тег **<B>**.

Например:

```
<P>Это <B>полужирный</B> шрифт</P>
```

Тег **<I>** отображает текст курсивом. Например:

```
<P>Выделение <I>курсивом</I></P>
```

Используя тег **<TT>**, можно отобразить текст шрифтом, в котором все буквы имеют одинаковую ширину. Это так называемый моноширинный шрифт.

Например:

```
<P>Это <TT>моноширинный</TT> шрифт</P>
```

Для отображения текста подчеркнутым используется тег **<U>**.

Например:

```
<P>Пример <U>подчеркивания</U> текста</P>
```

Тег **<S>** отображает текст, перечеркнутый горизонтальной линией.

Например:

```
<P>Пример <S>зачеркивания</S> текста</P>
```

Тег **<BIG>** выводит текст шрифтом большего (чем непомеченная часть текста) размера. Например:

```
<P>Шрифт <BIG>большого</BIG> размера</P>
```

Тег **<SMALL>** выводит текст шрифтом меньшего размера. Например:

```
<P>Шрифт <SMALL>меньшего</SMALL> размера</P>
```

Тег **<SUB>** сдвигает текст ниже уровня строки и выводит его (если возможно) шрифтом меньшего размера. Удобно использовать для математических индексов. Например:

```
<P>Шрифт <SUB>нижнего</SUB> индекса</P>
```

Тег **<SUP>** сдвигает текст выше уровня строки и выводит его (если возможно) шрифтом меньшего размера. Удобно использовать для задания степеней чисел в математике. Например:

```
<P>Шрифт <SUP>верхнего</SUP> индекса</P>
```

Теги форматирования могут быть вложенными друг в друга. При этом нужно внимательно следить, чтобы один контейнер находился целиком в другом контейнере. Например:

```
<P>Этот текст <B><I>полужирный и курсивный</I></B></P>
```

Вот так браузер отобразит приведенные выше примеры:

Форматирование текста
Это <b>полужирный</b> шрифт
Выделение <i>курсивом</i>
Это моноширинный шрифт
Пример <u>подчеркивания</u> текста
Пример <del>зачеркивания</del> текста
Шрифт <b>большого</b> размера
Шрифт <b>маленького</b> размера
Шрифт <small>нижнего</small> индекса
Шрифт <sup>верхнего</sup> индекса
Этот текст <i>полужирный и курсивный</i>

Тег <FONT>

Тег <FONT> позволяет изменить шрифт для блока текста. Этот тег имеет следующую спецификацию:

<FONT

FACE="тип шрифта"

COLOR="цвет шрифта"

SIZE="размер шрифта"

>

Атрибут FACE служит для указания типа шрифта, которым браузер будет выводить текст (если такой шрифт имеется на компьютере). Значением данного атрибута служит название шрифта, которое должно в точности совпадать с названием шрифта, имеющимся у пользователя. Если такой шрифт не найдется, то данное указание проигнорируется и будет использован шрифт, установленный по умолчанию.

Можно установить как один, так и несколько названий шрифтов, разделяя их запятыми. Тогда список шрифтов будет просматриваться слева направо так: если на компьютере пользователя нет шрифта, указанного в списке первым, то делается попытка найти следующий и т.д. Если ни одного шрифта найти не удалось, то будет использоваться шрифт, установленный браузером по умолчанию.

Атрибут COLOR устанавливает цвет шрифта. Значение этого атрибута может быть указано в формате RGB или символьной нотацией.

Атрибут SIZE служит для указания размера шрифта. Указывать размер шрифта можно двумя способами: абсолютной величиной или относительной величиной. При указании размера абсолютной величиной значением атрибута является число от 1 до 7. Размер 1 – самый маленький шрифт, 7 – самый большой. При указании размера *относительной величиной* значением атрибута является число со знаком + или -. В данном случае шрифт будет увеличен (+) или уменьшен (-) от размера, принятого по умолчанию. Оперативно в браузере это делается клавишами [Ctrl] [+/-].

Пример использования тега <FONT>:

```
<HTML>
<HEAD>
<TITLE>Пример изменения шрифта</TITLE>
</HEAD>
<BODY>
<P>Шрифт по умолчанию<BR>
<FONT COLOR="GREEN">зеленый шрифт</FONT><BR>
<FONT FACE="Arial">другая форма шрифта</FONT><BR>
<FONT SIZE="6">размер шрифта – 6</FONT><BR>
<FONT SIZE="+1">размер увеличен на 1</FONT></P>
</BODY>
</HTML>
```

Этот пример отобразится так:

Пример изменения шрифта

Шрифт по умолчанию  
 зеленый шрифт  
 другая форма шрифта  
 размер шрифта -6  
 размер увеличен на 1

## Контейнер DIV

Иногда бывает необходимо произвести выравнивание большого блока документа, содержащего не только текст, но и рисунки, таблицы и т.п. Для этих целей используется элемент-контейнер **DIV**. Тег `<div>` является блочным, его содержимое всегда начинается с новой строки и занимает всю доступную ширину, но отступов от других элементов он не имеет, то есть элементы, расположенные до и после него, будут вплотную прилегать сверху и снизу.

Спецификация элемента DIV:

`<DIV`

`ALIGN="выравнивание">`

Атрибут ALIGN определяет тип выравнивания содержимого и может иметь те же значения, что и тег абзаца `<P>`. Тег `<div>` не делает отступов снизу и сверху блока текста, так как он не помещает его в параграф, в отличие от тега `<P>`.

Будьте внимательны и не допускайте таких ошибок:

`<p align="center">`

`<p>текст</p>`

`<p>текст</p>`

`<p>текст</p>`

`</p>`

- помещать внутрь тега `<p>...</p>` еще теги `<p>` или `<div>` нельзя, так как этот код будет не правильно распознан

**И**

`<p align="right">`

`<div>текст</div>`

`<p>текст</p>`

`<div>текст</div>`

`</p>`

- неправильный код

Но, ошибкой не будет, если Вы поместите внутрь тега `<div>` тег `<p>`, так как он может содержать несколько абзацев. В следующем примере все абзацы будут размещены по правому краю страницы:

```
<div align="right">
<p> первый абзац</p>
<p> второй абзац</p>
<p> третий абзац</p>
</div>
```

---

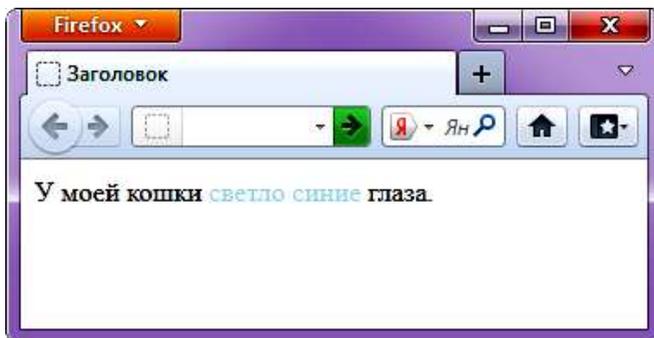
Тег `<span>` является строчным элементом и представляет из себя контейнер для текста. Он используется в ситуациях, когда нужно изменить стиль для части текста или отдельного слова, не помещая их при этом в блочный элемент `<div>`. Сам по себе он не имеет ни какого-либо визуального представления, ни семантического значения. Может использоваться, например, для:

- выделение части текста цветом, изменение размера или шрифта
- добавление цветного фона или фоновое изображение к тексту
- заключение фрагмента текста в рамку, подчеркивание или добавление тени

Пример.

```
<p>У моей кошки <span style="color: lightblue;">
    светло синие</span> глаза.
</p>
```

Результат данного примера в окне браузера:



### Короткие и длинные цитаты

Существует множество случаев, когда в тексте используются двойные кавычки, но в случае применения HTML тега `<q>` (quote), это означает, что используется именно цитата. Он является строчным элементом и определяет короткие цитаты в HTML-документе, то есть определяет текст цитаты непосредственно в тексте.

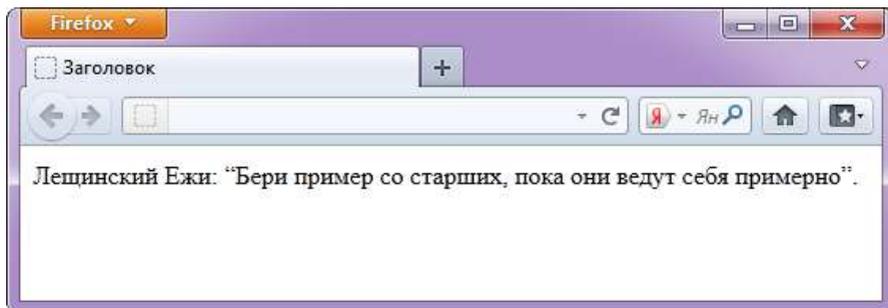
Другими словами, вы должны указать браузеру, что хотите добавить именно цитату. Если вы просто поставите двойные кавычки, то браузер решит, что есть абзац текста с парой двойных кавычек в нем. А если использовать тег `<q>`, то браузер интерпретирует часть этого текста как реальную цитату.

И вот теперь, когда браузер знает, что это цитата, он может отобразить ее наилучшим из возможных способов. Некоторые браузеры отобразят двойные кавычки вокруг текста, некоторые не отобразят, а в отдельных случаях могут использоваться и другие методы. Этот тег полезен и в других ситуациях, например при работе поисковых систем, просматривающих Сеть и выбирающих страницы с цитатами.

#### Пример.

- 1 `<p>Лещинский Ежи:`
- 2 `<q>Бери пример со старших, пока они ведут себя примерно</q>.`
- 3 `</p>`

Результат данного примера в окне браузера:

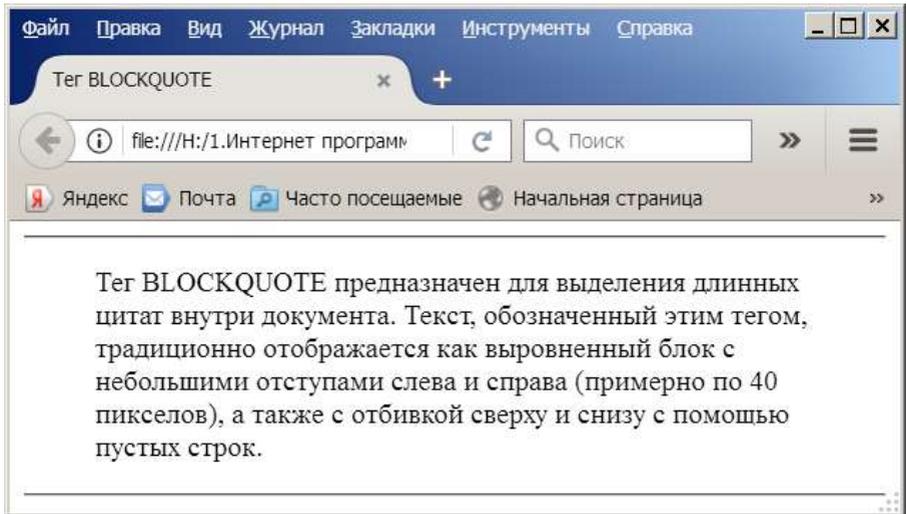


---

Иногда требуется отобразить блок текста с отступом. Для этого блок текста помещают в элемент-контейнер **BLOCKQUOTE**.

Тег `<blockquote>` предназначен для выделения длинных цитат внутри документа, занимающих несколько строк или целый абзац. Текст, обозначенный этим тегом, традиционно отображается как выровненный блок с небольшими отступами слева и справа (примерно по 40 пикселей), а также с *отбивкой* сверху и снизу с помощью пустых строк.

```
<html>
<head>
  <meta charset="utf-8">
  <title>Тег BLOCKQUOTE</title>
</head>
<body>
  <hr>
  <blockquote>
    <p>Тег BLOCKQUOTE предназначен для выделения длинных цитат внутри документа.
    Текст, обозначенный этим тегом, традиционно отображается как выровненный блок
    с небольшими отступами слева и справа (примерно по 40 пикселей), а также с
    отбивкой сверху и снизу с помощью пустых строк.</p>
  </blockquote>
  <hr>
</body>
</html>
```



## ТАБЛИЦЫ

Одним из наиболее мощных и широко применяемых в HTML средств являются таблицы. Они используются не только традиционно как метод представления данных, но и как средство форматирования Web-страниц. Документ HTML может содержать произвольное число таблиц, причем допускается вложенность таблиц друг в друга.

Каждая таблица начинается тегом `<TABLE>` и заканчивается тегом `</TABLE>`. Внутри этой пары тегов располагается описание содержимого таблицы. Любая таблица состоит из одной или нескольких строк, в которых задаются данные для отдельных ячеек.

Каждая строка начинается тегом `<TR>` и заканчивается тегом `</TR>`. Отдельная ячейка в строке обрамляется парой тегов `<TD>` и `</TD>` или `<TH>` и `</TH>`. Тег `<TH>` используется для ячеек заголовка таблицы, а `<TD>` – для ячеек данных. Отличие этих тегов в том, что в заголовке по умолчанию используется полужирный шрифт, а для данных – обычный.

Теги `<TD>` и `<TH>` не могут появляться вне описания строки таблицы `<TR>`.

Пример таблицы:

```
<HTML>
<HEAD>
<TITLE>Пример таблицы</TITLE>
</HEAD>
<BODY>
<TABLE>
<TR>
<TD>Ячейка 1</TD>
<TD>Ячейка 2</TD>
</TR>
<TR>
<TD>Ячейка 3</TD>
<TD>Ячейка 4</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

Этот пример отобразится в браузере так:

Пример таблицы		(в FF без рамочки)
Ячейка 1	Ячейка 2	
Ячейка 3	Ячейка 4	

Спецификация тега <TABLE>:

```
<TABLE
ALIGN="выравнивание"
BORDER="толщина рамки"
CELLPADDING="расстояние"
CELLSPACING="расстояние"
HEIGHT="высота"
VALIGN="вертикальное выравнивание"
WIDTH="ширина"           width
>
```

Атрибут ALIGN определяет выравнивание таблицы в окне просмотра браузера. Он может иметь одно из двух значений: LEFT (по левому краю) и RIGHT (по правому краю). По умолчанию – LEFT.

Атрибут BORDER управляет толщиной рамки. Значением этого атрибута является число. Это число определяет толщину рамки таблицы в пикселях. По умолчанию толщина рамки – 1.

Атрибут CELLPADDING определяет расстояние в пикселях между рамкой и содержимым ячейки. По умолчанию – 1.

Атрибут CELLSPACING определяет расстояние в пикселях между ячейками таблицы. По умолчанию – 2.

Атрибут HEIGHT определяет высоту таблицы в пикселях.

Атрибут VALIGN определяет вертикальное выравнивание содержимого таблицы. Он может иметь следующие значения: TOP (по верхнему краю), MIDDLE (посередине) и BOTTOM (по нижнему краю). По умолчанию – MIDDLE.

Атрибут WIDTH определяет ширину таблицы в пикселях или процентах от ширины окна браузера.

Спецификация тега <TR>

```
<TR
```

**ALIGN="выравнивание"**

**BGCOLOR="цвет фона"**

**VALIGN="вертикальное выравнивание"**

**>**

Атрибут ALIGN определяет выравнивание содержимого всех ячеек строки. Он может иметь одно из трех значений: LEFT (по левому краю), RIGHT (по правому краю) и CENTER (по центру). По умолчанию – CENTER.

Атрибут BGCOLOR определяет цвет фона для всех ячеек строки. Его значение можно указывать в символьной нотации или в формате RGB.

Атрибут VALIGN определяет вертикальное выравнивание содержимого всех ячеек строки. Он может иметь следующие значения: TOP (по верхнему краю), MIDDLE (посередине) и BOTTOM (по нижнему краю). По умолчанию – MIDDLE.

Спецификация тега <TD>

**<TD**

**ALIGN="выравнивание"**

**BGCOLOR="цвет фона"**

**COLSPAN="количество ячеек"**

**HEIGHT="высота ячейки"**

**ROWSPAN=" количество ячеек "**

**VALIGN="вертикальное выравнивание"**

**WIDTH="ширина ячейки"**

**>**

Атрибут ALIGN определяет выравнивание содержимого ячейки. Он может иметь одно из трех значений: LEFT (по левому краю), RIGHT (по правому краю) и CENTER (по центру). По умолчанию – CENTER.

Атрибут BGCOLOR определяет цвет фона для ячейки. Его значение можно указывать в символьной нотации или в формате RGB.

Атрибут COLSPAN позволяет объединить несколько соседних ячеек по горизонтали. Значение этого атрибута – количество объединяемых ячеек.

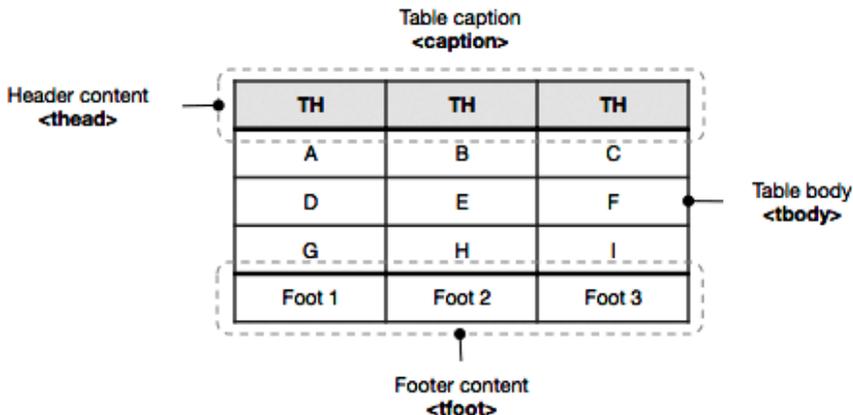
Атрибут HEIGHT определяет высоту ячейки в пикселях.

Атрибут ROWSPAN позволяет объединить несколько соседних ячеек по вертикали. Значение этого атрибута – количество объединяемых ячеек.

Атрибут VALIGN определяет вертикальное выравнивание содержимого ячейки. Он может иметь следующие значения: TOP (по верхнему краю),

MIDDLE (посередине) и BOTTOM (по нижнему краю). По умолчанию – MIDDLE.

Атрибут WIDTH определяет ширину ячейки в пикселях.



HTML тег `<caption>` определяет заголовок или описание к таблице и должен быть вставлен сразу после открывающего тег `<table>`. Для каждой таблицы может быть задан только один заголовок.

По умолчанию, заголовок выводится над таблицей и выравнивается по центру. Для выравнивания заголовка по левому или правому краю используется CSS свойство `text-align`.

HTML тег `<colgroup>` определяет группу из одного или нескольких столбцов таблицы для форматирования.

Обычно тег `<colgroup>` используется для применения стиля на целые столбцы, вместо того, чтобы повторять стили для нужной ячейки в каждой строке.

Элемент `<colgroup>` должен быть расположен после открывающего тега `<table>` и после тега `<caption>`, если он присутствует, и перед тегам `<thead>`, `<tbody>`, `<tfoot>` и `<tr>`.

Чтобы определить различные свойства для столбцов в `<colgroup>`, используйте тег `<col>` в элементе `<colgroup>`.

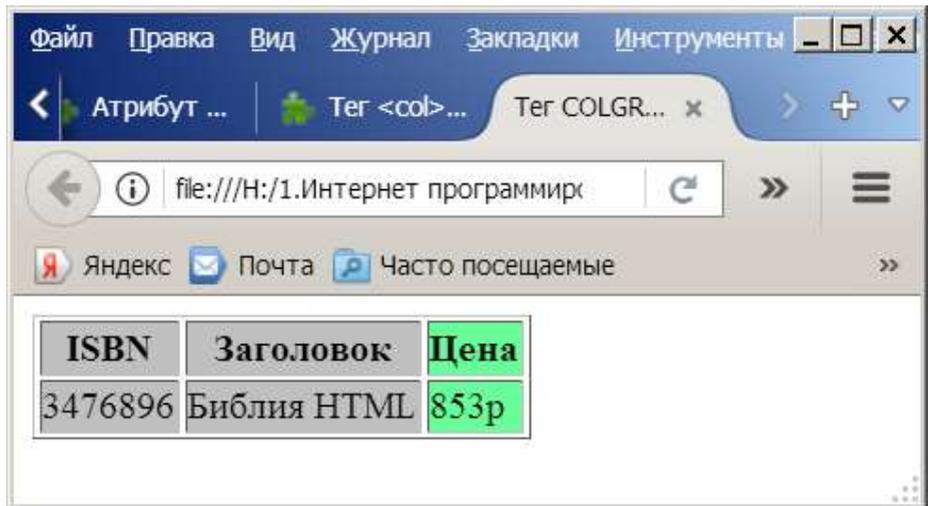
Атрибуты: `span`

Положительное целое число, указывающее количество следующих друг за другом столбцов, которое представляет элемент `<colgroup>`.

```

просмотр colgroup.html
<html>
<head>
  <meta charset="windows-1251">
  <title>Тег COLGROUP</title>
</head>
<body>
<table border="1">
  <colgroup>
    <col span="2" style="background-color: silver">
    <col style="background-color: #66FF99">
  </colgroup>
  <tr>
    <th>ISBN</th>
    <th>Заголовок</th>
    <th>Цена</th>
  </tr>
  <tr>
    <td>3476896</td>
    <td>Библия HTML</td>
    <td>853p</td>
  </tr>
</table>
</body>

```



HTML тег `<col>` представляет один или несколько столбцов в элементе `<colgroup>`.

Тег `<col>` используется для применения стилей на целые столбцы, вместо того, чтобы повторять стили для нужной ячейки в каждой строке. Для применения к столбцам таблицы разных стилей, нужно использовать несколько элементов `<col>`, каждый из которых будет влиять на то количество столбцов, которое задано атрибутом `span`.

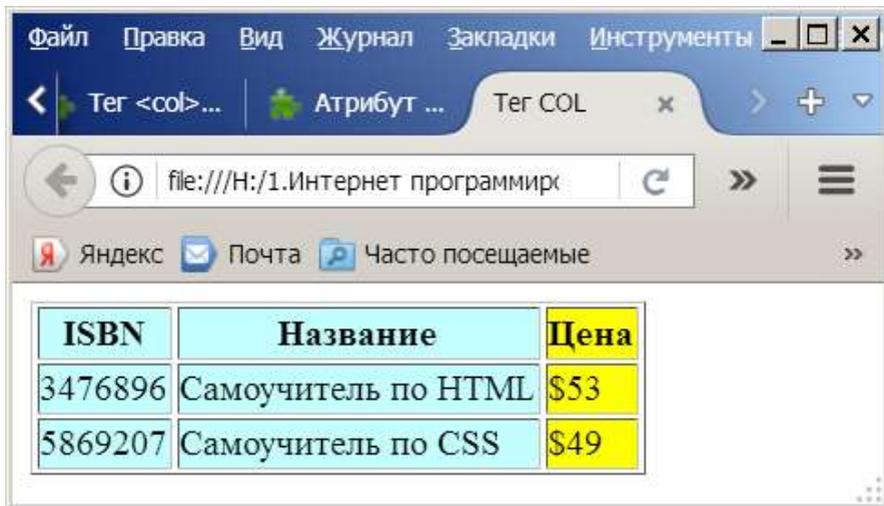
Атрибуты: `span`

Содержит положительное целое число, указывающее количество следующих друг за другом столбцов, которое представляет элемент `<col>`.

```

<html>
<head>
  <meta charset="windows-1251">
  <title>Тег COL</title>
</head>
<body>
<table border="1">
  <colgroup>
    <col span="2" style="background-color:#C2FFFF">
    <col style="background-color:yellow">
  </colgroup>
  <tr>
    <th>ISBN</th><th>Название</th><th>Цена</th>
  </tr>
  <tr>
    <td>3476896</td><td>Самоучитель по HTML</td><td>$53</td>
  </tr>
  <tr>
    <td>5869207</td><td>Самоучитель по CSS</td><td>$49</td>
  </tr>
</table>
</body>
</html>

```



Тег **<THEAD>** применяется для группирования рядов (строк) таблицы. Такое группирование используется с целью задания единого оформления группе рядов, чтобы повысить качество восприятия и удобство пользования таблицей. Естественно, к тегу **<THEAD>** можно применить не только его атрибуты, но и стили (CSS).

Тег **<THEAD>** создает так называемую «шапку» таблицы, то есть внутри этого элемента располагаются ряды, которые будут выведены в начале таблицы. Причем таблица может иметь только одну «шапку».

Еще ряды таблицы можно сгруппировать в ее тело (тег **<TBODY>**) и футер (тег **<TFOOT>**). Во всех группах рядов обязательно должно быть одинаковое количество столбцов (без учета объединения ячеек).

Модель тега: может располагаться внутри элемента **<TABLE>**, но только в одном экземпляре перед тегами **<TFOOT>** и **<TBODY>**.

Должен содержать: один или более тегов **<TR>**.

## Синтаксис

```

<table>
  <thead>
    <tr>
      <td> ... </td>
    </tr>
  </thead>
  <tfoot> ... </tfoot>
  <tbody> ... </tbody>
</table>

```

## Атрибуты

**align**

Определяет выравнивание текста в строке.

**char**

Выравнивание содержимого ячеек относительно заданного символа.

**charoff**

Смещение содержимого ячеек относительно указанного символа.

**bgcolor**

Цвет фона ячеек, которые расположены внутри контейнера <thead>.

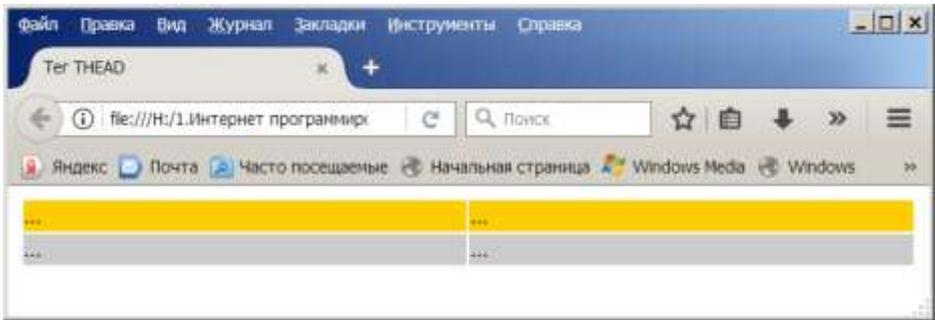
**valign**

Выравнивание содержимого ячеек по вертикали.

## Закрывающий тег

Не обязателен.

```
<html>
<head>
  <meta charset="windows-1251">
  <title>Ter THEAD</title>
</head>
<body>
  <table style="width: 600px;">
    <thead style="background: #fc0">
      <tr>
        <td> ... </td>
        <td> ... </td>
      </tr>
    </thead>
    <tbody style="background: #ccc">
      <tr>
        <td> ... </td>
        <td> ... </td>
      </tr>
    </tbody>
  </table>
</body>
</html>
```



---

HTML тег **<tfoot>** используется для группировки содержимого в нижнем колонтитуле таблицы.

Элемент `<tfoot>` используется в сочетании с элементами `<thead>` и `<tbody>` для разделения таблицы на логические части (тело, заголовок и нижний колонтитул).

Браузеры могут использовать эти элементы для включения прокрутки тела таблицы, независимо от заголовка и нижнего колонтитула. Так же при выводе на печать большой таблицы, которая занимает несколько страниц, эти элементы могут включить заголовок и нижний колонтитул, которые будут напечатаны в верхней и нижней части каждой страницы.

Элемент `<tfoot>` должен быть расположен после элементов `<caption>` и `<colgroup>` (если таковые присутствуют) и `<thead>`, но до любого элемента `<tbody>`.

Элемент `<tfoot>` должен быть расположен перед элементом `<tbody>` в таблице, чтобы браузер мог отобразить нижний колонтитул в самом низу таблицы, перед обработкой всех строк с данными таблицы.

В пределах таблицы разрешается использовать только один элемент `<tfoot>`.

## Синтаксис

```
<table>
```

```
<thead> ... </thead>  
<tfoot>  
  <tr>  
    <td> ... </td>  
  </tr>  
</tfoot>  
<tbody> ... </tbody>  
</table>
```

## Атрибуты

### align

Определяет выравнивание текста в строке.

### bgcolor

Цвет фона ячеек, которые расположены внутри контейнера <tfoot>.

### char

Выравнивание содержимого ячеек относительно заданного символа.

### charoff

Смещение содержимого ячеек относительно указанного символа.

### valign

Выравнивание содержимого ячеек по вертикали.

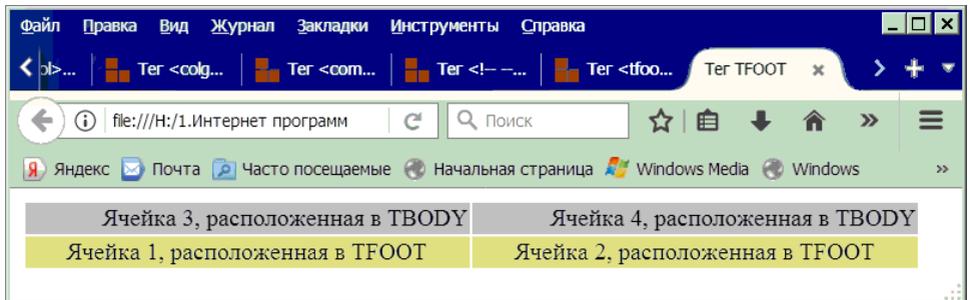
## Закрывающий тег

Не обязателен.

```

<html>
<head>
  <meta charset="windows-1251">
  <title>Ter TFOOT</title>
</head>
<body>
  <table width="600">
    <tfoot align="center" style="background: #ffc">
      <tr>
        <td>Ячейка 1, расположенная в TFOOT</td>
        <td>Ячейка 2, расположенная в TFOOT</td>
      </tr>
    </tfoot>
    <tbody align="right" style="background: silver">
      <tr>
        <td>Ячейка 3, расположенная в TBODY</td>
        <td>Ячейка 4, расположенная в TBODY</td>
      </tr>
    </tbody>
  </table>
</body>
</html>

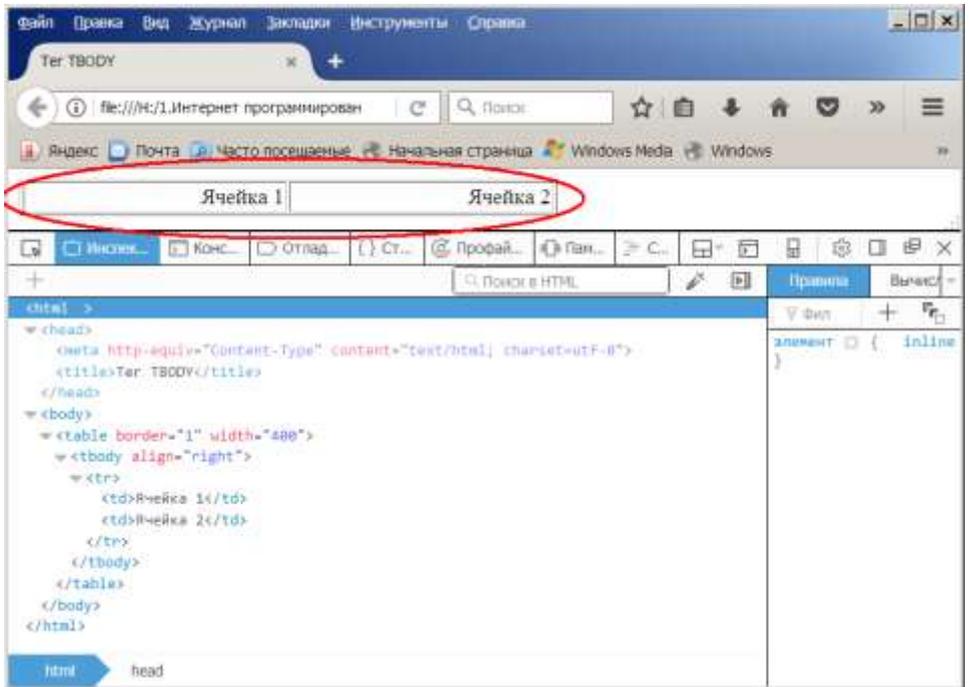
```



Элемент `<tbody>` предназначен для хранения одной или нескольких строк таблицы. Это позволяет создавать структурные блоки, к которым можно применять единое оформление. Допускается применять несколько тегов `<tbody>` внутри контейнера `<table>`.

Пример: выравнивание по правому краю применено ко всем ячейкам.

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Тег TBODY</title>
</head>
<body>
  <table width="400" border="1">
    <tbody align="right">
      <tr>
        <td>Ячейка 1</td>
        <td>Ячейка 2</td>
      </tr>
    </tbody>
  </table>
</body>
</html>
```



Правильная последовательность вышеперечисленных табличных тегов:

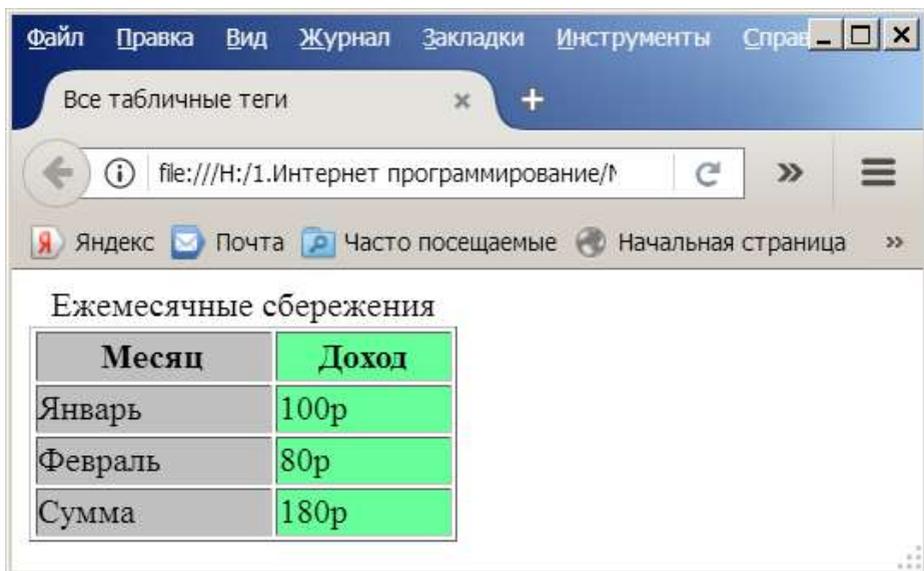
```

<table>
<caption>
<colgroup>
<col>
<thead>
<tfoot>
<tbody>

```

Пример с использованием всех табличных тегов:

```
<html>
<head>
  <meta charset="windows-1251">
  <title>Все табличные теги</title>
</head>
<body>
<table border="1" style="width: 200px;">
  <caption>Ежемесячные сбережения</caption>
  <colgroup>
    <col span="1" style="background-color: silver">
    <col style="background-color: #66FF99">
  </colgroup>
  <thead>
    <tr>
      <th>Месяц</th>
      <th>Доход</th>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <td>Сумма</td>
      <td>180р</td>
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <td>Январь</td>
      <td>100р</td>
    </tr>
    <tr>
      <td>Февраль</td>
      <td>80р</td>
    </tr>
  </tbody>
</table>
</body>
</html>
```



### Вопросы для закрепления

- Чем отличаются абзацы от заголовков?
- Какие бывают списки?
- Какие теги используются для изменения шрифта?  
Что такое «информация»?

### Домашние задание:

- Создайте таблицу, которая в первой строке содержит одну ячейку, во второй – две, в третьей – три ячейки.
- На практике наблюдайте эффект использования различных значений атрибута ALIGN тега <IMG>.

## Тема 5

### Тема: Гиперссылки

**Цели:** *Ознакомить учащихся с основными понятиями гиперссылки; правилами создания ссылок на различные типы документов.*

План урока.

1. Организационный момент.
2. Изложение нового материала.
  - 1) Гиперссылки
  - 2) Универсальный идентификатор ресурсов URL
  - 3) Правила записи ссылок
3. Закрепление изученного.
4. Итог урока.

Ход урока.

#### **Организационный момент.**

Отметить присутствующих. Сообщить тему урока. Настроить учащихся на восприятие материала.

#### **Изложение нового материала.**

Как уже было сказано, гиперссылки – переходы к другим документам. Они являются основой HTML. Гиперссылки можно использовать для перехода не только к другим HTML-документам, но и другим объектам, которые можно разместить на компьютере, например, к видео- и аудиофайлам, архивам, рисункам и т.п.

Каждая ссылка состоит из двух частей. Первая – это то, что отображается браузером. Она называется **указателем ссылки**. Вторая часть – адресная, содержащая адрес объекта, к которому должен происходить переход. Эту часть называют **URL** (универсальный идентификатор ресурсов).

Когда вы щелкаете мышью по указателю ссылки, браузер загружает документ, адрес которого указан в URL.

Указателем ссылки может быть слово, группа слов или рисунок. Если указатель текстовый, то он обычно отображается браузером подчеркнутым синим шрифтом. При наведении курсора мыши на указатель курсор принимает форму руки, указывая, что это ссылка и можно произвести переход. Если указатель графический, внешне он не отличается от других рисунков, но при наведении курсора мыши на такой указатель, курсор также принимает форму руки.

## УНИВЕРСАЛЬНЫЙ ИДЕНТИФИКАТОР РЕСУРСОВ URL

По своей сути URL – это адрес файла, к которому происходит переход. Указание адреса может быть относительным или абсолютным.

Когда в URL указывается полный путь к файлу, независимо от того, где этот файл находится (в Интернете или на жестком диске компьютера), то это абсолютное указание. Например: **http://www.mysite.ru/page.htm** – абсолютный адрес документа, находящегося в Интернете, **c:\web\doc1.htm** – абсолютный адрес документа на диске **c**.

Если в URL указывается не полный путь, а путь относительно адреса документа, в котором находится ссылка, то это относительное указание. Например, браузер отображает документ, абсолютный адрес которого **c:\web\doc1.htm**, в этом документе имеется ссылка с адресом **pict/ris1.jpg**, то это означает, что на самом деле ссылка будет на документ по адресу **c:\web\pict\ris1.jpg**.

Когда надо сослаться на файл, находящийся на другом компьютере, тогда следует пользоваться абсолютным указанием, а если ссылаться на файлы, находящиеся на том же компьютере, где и документ, содержащий ссылку, то лучше использовать относительное указание.

### Правила записи ссылок

Для организации ссылки необходимо указать браузеру, что является указателем ссылки, а также определить адрес документа, на который происходит ссылка. Оба этих действия выполняются с помощью тега **<A>**.

Тег **<A>** имеет следующую спецификацию:

```
<A  
HREF="URL-адрес"  
NAME="имя ссылки"  
TARGET="объект для вывода"  
>
```

Атрибут **HREF** используется для задания адреса файла, к которому производится переход. Значением этого атрибута является текстовая строка, содержащая абсолютный или относительный URL-адрес.

Атрибут **NAME** предназначен для задания ссылки имени. Значением этого атрибута является короткая текстовая строка. Этот атрибут используется для ссылок внутри одного HTML-документа.

Атрибут **TARGET** позволяет определить, куда будет выводиться документ, на который происходит переход. Этот атрибут может иметь значение **\_blank** – это означает, что документ будет выводиться в новом окне браузера.

Пример ссылки:

```
<A HREF="doc1.htm">Документ 1</A>
```

Браузер отобразит эту строку так:

Документ 1

При нажатии мышью на этой строке браузер загрузит и отобразит файл doc1.htm, т.е. “Документ 1” – это указатель ссылки, а “doc1.htm” – URL-адрес.

### ВНУТРЕННИЕ ССЫЛКИ

Кроме ссылок на другие документы, часто бывает полезно включить ссылки на различные части текущего документа. Например, большой документ читается лучше, если он имеет оглавление со ссылками на соответствующие разделы.

Для построения внутренней ссылки сначала нужно создать указатель, определяющий место назначения. Для этого в месте, куда потом будет производиться ссылка, надо поместить тег <A> с атрибутом NAME, определив этим атрибутом имя указателя. Например:

```
<A NAME="glava5"></A>
```

Обратите внимание, что в этом примере отсутствует содержимое тега <A>. Обычно так и делают, поскольку здесь нет необходимости как-то выделять текст, а требуется лишь указать местоположение.

После того как место назначения определено, можно приступить к созданию ссылки на него. Для этого в атрибуте HREF тега <A> помещается имя ссылки с префиксом #, говорящим о том, что это внутренняя ссылка. Например:

```
<A HREF="#glava5">Глава 5</A>
```

Теперь, если пользователь щелкнет кнопкой мыши на словах “Глава 5”, то браузер выведет соответствующую часть документа в окне просмотра.

Можно совмещать внутренние ссылки со ссылками на другие документы. Например:

```
<A HREF="doc1.htm#glava5">Глава 5 Документа 1</A>
```

При нажатии на эту ссылку браузер откроет файл doc1.htm, найдет в этом файле указатель glava5 и выведет в окне просмотра соответствующую информацию.

### ССЫЛКИ НА ДОКУМЕНТЫ РАЗЛИЧНЫХ ТИПОВ

Когда пользователь щелкает мышью на ссылке, указывающей на другой HTML-документ, то документ выводится непосредственно в окне браузера. Если же ссылка указывает на документ иного типа, программа просмотра принимает документ и затем решает, что с ним делать дальше.

□ Браузер знает этот тип документа и умеет с ним обращаться. Например, если создать ссылку на графический файл формата GIF и щелкнуть мышью на ней, то браузер очистит окно просмотра и отобразит указанное изображение.

□ Браузер не распознает тип принятого документа и не знает, что с ним делать дальше. В этом случае он обратится к вспомогательным программам, имеющимся на компьютере пользователя. Если подходящая программа найдется, браузер запустит ее и передаст ей полученный документ для обработки. Например, если пользователь щелкнет на ссылке на видеофайл формата AVI, браузер загрузит файл, найдет программу для демонстрации AVI-файлов и запустит ее. Видеофайл будет показан в дополнительном небольшом окне.

### Ссылки на ресурсы Интернета

Основное назначение HTML-документов – это глобальная компьютерная сеть Интернет. HTML-документ, размещенный в Интернете, становится Web-страницей. Чтобы обратиться к Web-странице, надо указать URL-адрес в такой форме: **http://sitename/docname**, где sitename – имя сайта, docname – имя документа. Например: <http://kotoz.newmail.ru/autor.htm>.

Можно на Web-странице разместить адрес электронной почты. Для этого URL-адрес указывается так: **mailto:address**, где address – это адрес почтового ящика. Например: <mailto:vasya@mail.ru>.

Также существуют специальные форматы URL-адреса для других ресурсов Интернета (FTP, TelNet, Newsgroup, Gopher, WAIS).

Вопросы для закрепления.

1. Что такое URL?
2. Что такое внутренняя ссылка и как она создается?
3. На какие файлы можно ссылаться в HTML-документах?

Домашнее задание: Создать HTML- документ с ссылками на три различных типа документов.

## Тема 6

### Тема: Технологии web-дизайна

**Цели:** *Ознакомить учащихся с технологическими и художественными решениями применяемых в современном web-дизайне.*

План урока.

1. Организационный момент.
2. Изложение нового материала.
  - 1) Логическая и физическая структура сайта
  - 2) Особенности верстки документов сайта.
  - 3) Виды модульных сеток
3. Закрепление изученного.
4. Итог урока.

Ход урока.

#### **Организационный момент.**

Отметить присутствующих. Сообщить тему урока. Настроить учащихся на восприятие материала.

#### **Изложение нового материала.**

#### **Логическая и физическая структура сайта**

Каждый ресурс Интернета, от любительской домашней странички до большого информационного портала, содержит несколько тематических рубрик, соединенных между собой гиперсвязями. Как правило, ссылки на все разделы сайта с краткими анонсами их содержимого приводятся на первой, так называемой стартовой странице, которой присваивается имя index.htm (.html). Если тематические рубрики содержат собственные подразделы, каждая из них также имеет свою стартовую страницу, называющуюся index.html.

**ПРИМЕЧАНИЕ** Такое имя файла рекомендуется присваивать всем стартовым документам сайта, поскольку в противном случае при обращении к какому-либо разделу посредством сокращенного URL без указания названия стартовой страницы (например, вместо <http://www.mysite.ru/photos/startpage.html> указать <http://www.mysite.ru/photos/>), браузер отобразит не саму web-страницу, а перечень хранящихся в данной папке файлов.

Подобный набор тематических рубрик с распределенными по соответствующим разделам документами и заранее спроектированными гиперсвязями между всеми страницами ресурса и называется логической структурой сайта. Физическая структура, напротив, подразумевает алгоритм размещения физических файлов по поддиректориям папки, в которой

опубликован ваш сайт. Пример сравнения логической и физической структур одного и того же ресурса Интернета показан на рис. 2.1.

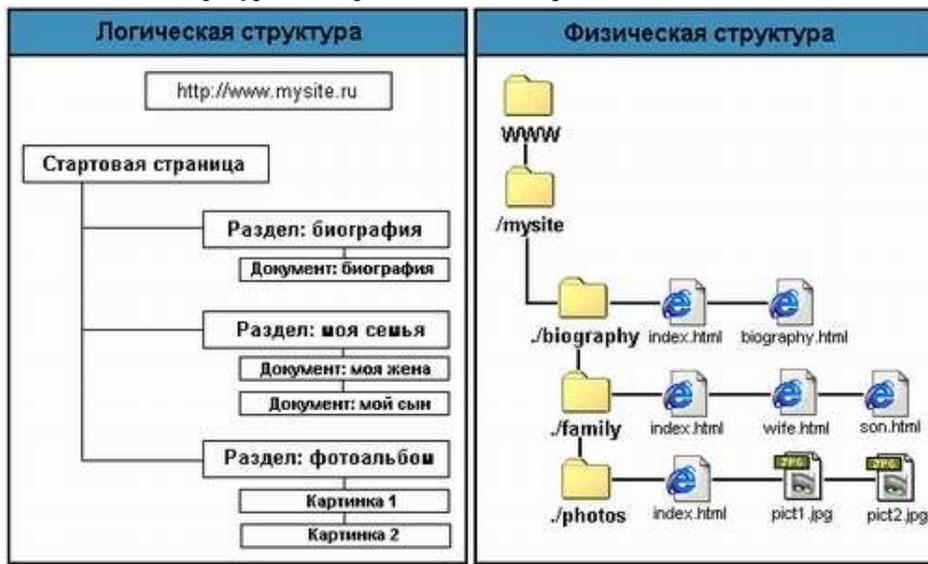


Рис. 2.1. Сравнение логической и физической структуры сайта

Очевидно, что логическая и физическая структуры могут не совпадать, поскольку в общем случае физическая структура ресурса разрабатывается, исходя из удобства размещения файлов. Однако более или менее точное сохранение порядка следования логических разделов в физической структуре сайта позволит вам избежать путаницы при последующем дополнении и обновлении материалов.

**СОВЕТ:** Рекомендуется размещать все графические изображения, являющиеся элементами проекта, в отдельной папке с названием "Images", расположенной в корневой директории сайта. Такой подход позволит обновлять хранящиеся в других тематических разделах документы HTML без переноса графики, использовать одни и те же графические файлы во всех разделах сайта и при необходимости удалять целые директории.

Для того чтобы все гиперссылки на вашей домашней страничке или web-сайте работали корректно, все документы открывались правильно и браузер не выдавал ошибок при обращении к каким-либо разделам ресурса, при создании его физической структуры следует соблюдать несколько простых правил.

**СОВЕТ:** Назначайте имена директорий, имена и расширения документов HTML и графических файлов с использованием символов только латинского алфавита и только в строчном регистре. Старайтесь, чтобы имена созданных вами файлов и директорий не превышали по длине восьми символов.

**СОВЕТ:** При присвоении имен файлов документам HTML старайтесь следить за тем, чтобы эти имена были "смысловыми": впоследствии вы легко можете забыть содержимое и назначение какой-либо web-страницы, если имена файлов будут выглядеть, например, как 1.htm, 2.htm, 3.htm и т.д.

Для того чтобы облегчить процесс обновления web-страниц, дополнения разделов или создания новых рубрик, заведите средство документирования проекта – любую электронную таблицу, созданную, например, в Microsoft Excel, или просто разграфленную тетрадку, в которую записывайте соответствие элементов физической структуры вашего проекта его логической структуре. До тех пор пока количество составляющих ваш сайт файлов относительно мало, это может показаться излишним, когда же оно перевалит за первые два десятка, в обилии html-документов и графических элементов будет легко запутаться, особенно если вы создаете несколько проектов одновременно.

Пример оформления такого средства документирования показан в табл. 2.2.

Таблица 2.2. Пример оформления средства документирования проекта

<b>Имя файла</b>	<b>Директория</b>	<b>Описание</b>	<b>Дата создания/ последнего изменения</b>
index.html	/mysite	Стартовая страница сайта <a href="http://www.mysite.ru">http://www.mysite.ru</a>	1.07.2017
index.html	/mysite/family	Стартовая страница раздела <моя семья>	5.07.2017
wife.htm	/mysite/family	Рассказ о моей жене	5.07.2017
son.htm	/mysite/family	Рассказ о моем сыне	5.07.2017
pict.jpg	/mysite/photos	Моя фотография на пляже в Сочи	6.07.2017

Из всего сказанного становится очевидным, что физическая структура сайта скрыта от посетителей вашего ресурса: они могут наблюдать только логическую структуру, причем именно так, как она представлена при помощи элементов навигации. Отсюда следует вполне логический вывод: строение

системы навигации должно если не полностью повторять, то хотя бы максимально соответствовать разработанной вами логической структуре сайта.

### *Заглавная страница*

Один из критериев, руководствуясь которым можно разделить различные web-сайты на две основные категории – это наличие заглавной страницы (splash) или отсутствие таковой.

Заглавная страница представляет собой html-документ, который не включает в себя какую-либо содержательную информацию, имеет мало ссылок и элементов навигации. Файлу заглавной страницы присваивается имя index.html, при этом стартовая страница называется иначе (index1.html) и вызывается посредством организации гиперссылки с заглавной страницы, загружающейся при обращении к сайту первой. Заглавная страница содержит, как правило, логотип компании-владельца данного ресурса, счетчик посещений и предложение выбора кодировки кириллицы, либо выбора между английской и русской версиями сайта.

Пример web-ресурса, оснащенного заглавной страницей, показан на рис. 2.3.



Рис. 2.3. Пример сайта, оснащенного заглавной страницей

При обращении к сайтам, не оснащенным заглавной страницей, первой отображается стартовый документ, включающий какое-либо информационное наполнение, элементы навигации и иногда анонсы составляющих данный ресурс тематических рубрик.

На рис. 2.4 показан пример сайта, не оснащенного заглавной страницей.

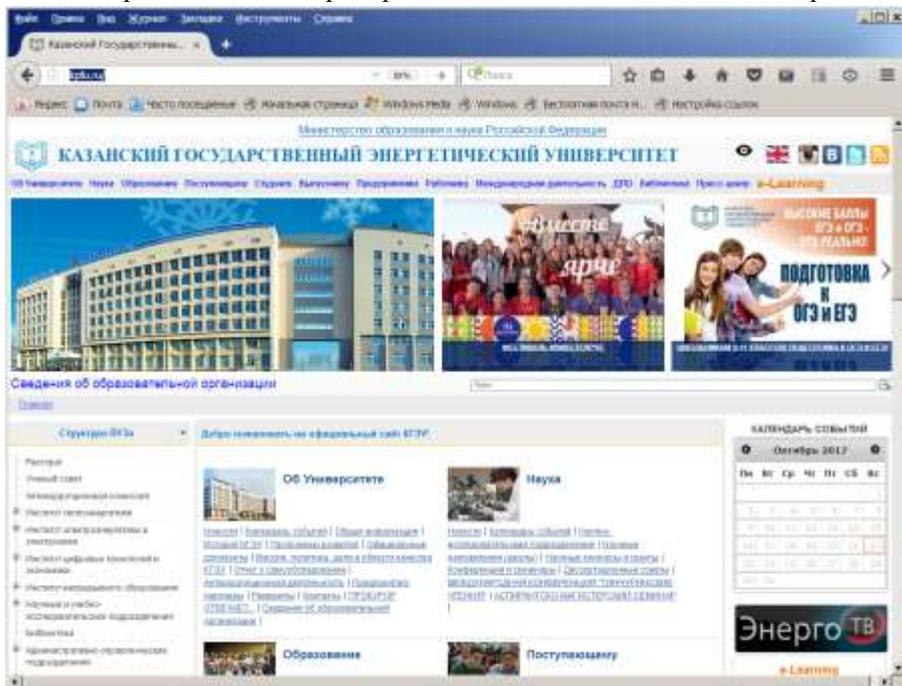


Рис. 2.4. Пример сайта, не оснащенного заглавной страницей

Использовать или не использовать заглавную страницу при создании собственного проекта в сети Интернет – дело вкуса каждого web-мастера. Дать какие-либо исчерпывающие рекомендации на этот счет трудно, поскольку окончательное решение зависит прежде всего от ваших художественных предпочтений и иногда – от желания заказчика, оплачивающего вашу работу.

### **Верстка web страниц**

Веб-страница запускается и отображается на компьютере под управлением клиентской программы – браузера. Понятно, что операционная система, ее настройки и собственно сам браузер отличается от компьютера к компьютеру. Из чего напрашивается вывод, что один и тот же документ сайта по-разному отображается у каждого пользователя.

При дизайне и верстке макета сайта, предназначенного для большой аудитории, специалисты должны учитывать следующие различия между посетителями:

- разрешение экрана;
- браузер пользователя;
- развернут ли браузер на весь экран;
- дополнительные инструменты браузера (закладки, история и т.д.);
- операционную систему и оборудование пользователя.

Прежде всего, необходимо решить, на какое разрешение монитора пользователя будет ориентирован сайт.

Напрашивается вопрос, можно ли сделать так, чтобы веб-страница отображалась одинаково? Учитывая, сколько придется принять во внимание разных неоднозначных факторов, следует сказать, что нет, нельзя. Поэтому задача верстки веб-страниц формулируется так: сформировать документ, который бы корректно отображался с небольшими различиями на основных платформах и в браузерах. Корректно означает, что соблюден исходный замысел автора, воплощен требуемый дизайн документа, и он показывается в браузере без ошибок.

Чтобы реализовать указанную задачу надо понимать, как вообще происходит верстка веб-страниц и соизмерять свои идеи с их исполнением.

### **Особенности верстки документов сайта**

#### **Ширина документа**

Изначально разработчику сайта ширина окна браузера пользователя неизвестна, поскольку она может меняться в самых широких пределах. Ширина зависит от разрешения монитора, длины его диагонали, размера окна и еще некоторых варьируемых данных. Иными словами предугадать ее заранее простыми средствами не представляется возможным. С учетом этой особенности утвердилось три способа верстки: фиксированный, «резиновый» и эластичный.

## 1) Фиксированный макет

В данном случае действуем от обратного и устанавливаем общую ширину макета жестко заданной и равной определенной величине. Если взять некоторую обобщенную статистику посетителей сайтов и посмотреть, какое разрешение монитора они преимущественно используют, то узнаем, что это 1024 x 768 пикселей и выше. Получается, что ширина монитора пользователей в основном  $\geq 1024$  пиксела.



Блоки внутри фиксированы по ширине, а 960 (или 1000)-пиксельный контейнер стал стандартом для пользователей интернет, которые предположительно имеют разрешение 1024\*768 или выше.

### Преимущества макетов с фиксированной шириной

- Фиксированные макеты легче использовать;
- Ширина фиксируется для всех браузеров, так что возникает меньше проблем с картинками, формами, видео и другим контентом, имеющим фиксированную ширину;
- Нет необходимости задавать минимальную (min-width) или максимальную (max-width) ширину, которые, кроме того, до сих пор поддерживаются не всеми браузерами;
- Даже если макет ориентирован на минимальное разрешение 800\*600px, контент будет все равно достаточно широким, чтобы быть читабельным;
- В некоторых случаях использовать макеты с фиксированной шириной проще для создания определенных JavaScript эффектов или дизайнерских решений. Некоторые макеты дизайна **разумно**

верстаются только с использованием фиксированной ширины основного контейнера;

- Фиксированные макеты верстаются быстрее и легче, что очень важно на срочных проектах.

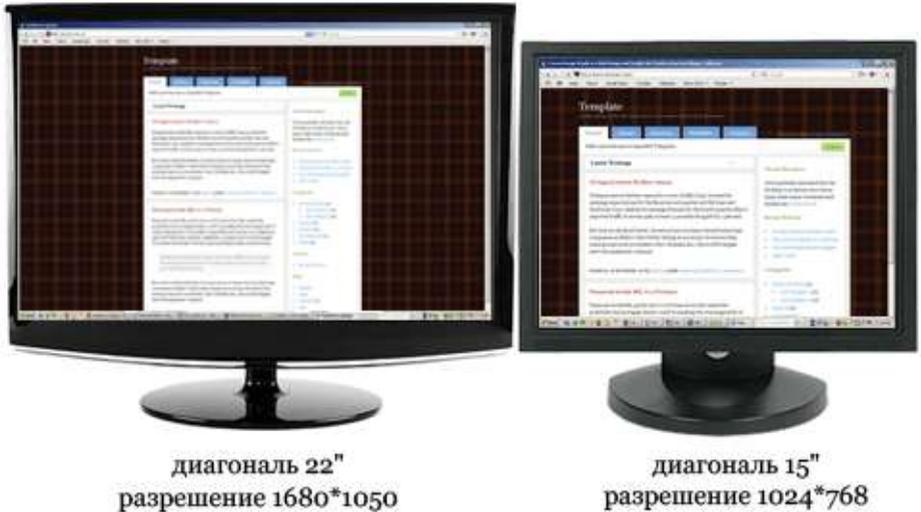
### Недостатки макетов с фиксированной шириной

- Фиксированный макет будет оставлять много пустого пространство у пользователей с большим разрешением экрана;
- При разрешениях, меньших чем тот, на который рассчитан макет, в браузере появится горизонтальная прокрутка;
- Для пользователей с большим разрешением экрана может понадобиться бесшовный повторяющийся фон;
- Фиксированный макет несколько уступает по юзабилити остальным.



*Рис. 2.5. Макет фиксированной ширины с размещением по левому краю*

Справа появляется широкая пустая полоса, размер которой зависит от разрешения экрана пользователя и диагонали его монитора. Чтобы хоть как-то уменьшить пустое пространство, макет обычно помещают по центру окна браузера (рис. 2.6).



*Рис. 2.6 Макет фиксированной ширины с размещением по центру*

## 2) «Резиновый» макет

Резиновый макет получается, если у основного контейнера не задавать ширину вообще (по умолчанию будет 100%) или задать ширину в процентах. Другими словами, не учитывая границы и отступы, сайт, имеющий ширину 100%, будет занимать всю видимую часть экрана, не создавая горизонтальной полосы прокрутки.

### Преимущества резиновых макетов

- Резиновые макеты более дружелюбны, так как они подстраиваются под пользователя;
- Количество свободного места подобно для всех браузеров и разрешений экрана, что визуально более привлекательно;
- Хорошо спроектированные резиновые макеты могут исключать появление горизонтальной прокрутки на меньших разрешениях;
- При грамотной верстке резиновый макет позволяет достичь большой универсальности в отображении, независимо от ширины окна

пользовательского клиента. На практике это встречается довольно редко, но, теоретически, если все делать правильно, то такое возможно;

- Резиновый макет совпадает с макетом, применяемым браузером по умолчанию, что позволяет считать его «идеологически правильным»;
- Резиновый макет сайта позволяет использовать доступное разрешение экрана по максимуму. Большая часть содержания страницы будет видимой без дополнительной прокрутки страницы.

### **Недостатки резиновых макетов**

- Разработчик в меньшей степени контролирует то, что видит пользователь, и может не увидеть проблемы на своем разрешении;
- Дизайнеру в некоторых случаях приходится рисовать несколько вариантов макета для разных разрешений;
- Картинки, видео и другие типы контента с фиксированными размерами желательно предоставить в нескольких вариантах для разных разрешений;
- На очень больших разрешениях контент может создать лишнее пустое пространство, что негативно повлияет на внешний вид сайта;
- На очень маленьких разрешениях, если не задана минимальная ширина макета (для этого используется CSS-свойство `min-width`, которое не поддерживается Internet Explorer (IE) 6 версии и раньше), при слишком маленьком разрешении блоки начинают наезжать друг на друга или перескакивать друг под друга;
- Слишком много доступного для чтения текста страницы и его острое желание заполнить всю доступное место может служить во вред удобству пользователей. Слишком много текста может «давить» на пользователя и делать сайт перегруженным или хаотичным. Существует выражение: “White space sells” (имеется в виду, что пустое пространство нужно для увеличения акцента на содержании), и оно справедливо применительно к вебсайтам;
- Если для ограничения ширины макета использовать CSS свойство `max-width` (которое не поддерживается Internet Explorer (IE) 6 версии и раньше) можно в результате стремления к «правильной идеологии» получить, сайт, которым будет тяжело пользоваться. Представьте себе, что нужно прочитать строку длиной более 1000 пикселей, и вы поймете проблему.

Резиновый макет необходимо использовать строго по назначению, не позволяя своим стремлениям сделать все «идеологически правильно» возобладать над здравым смыслом. Должны быть какая-то достаточно веская причина, чтобы использовать эту разновидность макета. Если вы все же решено использовать именно ее, необходимо убедиться, что резиновый макет сайта нормально смотрится на большинстве экранных разрешений. Иначе игра не стоит свеч.

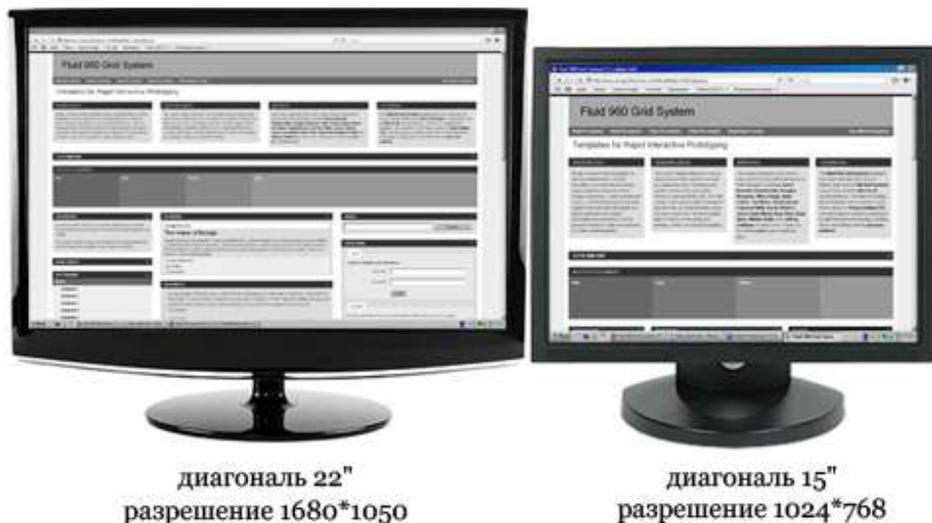


Рис. 2.7. Пример резинового макета

Чем меньше резиновый макет зависит от сложной и графической техники, тем легче будет его использовать и обслуживать. Он также будет более совместим с различными разрешениями экрана. С более чистым кодом и дизайном проблемы совместимости легче предотвратить, найти и устранить.

### 3) Эластичный макет

При использовании эластичного макета исходный размер шрифта определяется не в пикселях, а в em. Размер основного контейнера и других важных элементов/модулей также задается в **em** (не путать с тегом `<em>`, т.е. с

курсивом). EM – это относительная единица длины, равная размеру текущего шрифта. По умолчанию 1em равняется 16px, соответственно  $10px = 0,625em$ . Em прямо пропорциональны размеру текста или шрифта. Следовательно, при увеличении размера текста на странице, размеры разделов, заданные в em, увеличатся пропорционально. Таким образом, разработчик получает эластичную систему изменения размеров сайта при изменении размера шрифта.



## Преимущества эластичного макета

- В случае правильного исполнения макет может быть очень и очень дружелюбным пользователю (user-friendly). Цель – увеличивать или уменьшать все в зависимости от настроек пользователя;
- Эластичный макет понравится и разработчикам, предпочитающим фиксированный макет, и разработчикам, предпочитающим резиновый макет, так как он объединяет в себе преимущества обоих.
- Макет целиком или отдельные его части легко масштабировать, подгоняя под комфортный для восприятия размер.
- Макет будет одинаково смотреться на разных операционных системах, имеющих различия в выборе размера и типа шрифта.

## Недостатки эластичного макета

- Даже с учетом всех преимуществ, эластичный макет может стать источником проблем с юзабилити. Много времени уйдет на разработку и тестирование макета, чтобы сделать его подходящим для всех пользователей;

- Трудоемкость. Для разработки такого макета приходится приложить намного больше усилий и времени, чем для двух других, что не всегда оправдано;
- Зависимо от специфики, некоторые эластичные макеты требуют дополнительных стилей и «хаков» для устаревшего IE6.
- В современных браузерах функция масштаба страницы уже встроена, и пользоваться ей довольно удобно.
- В действительности сфера применения этого макета очень ограничена.

### **Высота документа**

Исторически сложилось, что пролистывание большого документа на компьютере происходит сверху вниз. Для удобства листания предназначены вертикальные полосы прокрутки, клавиатурные комбинации, колесо прокрутки мыши. А вот перемещение по горизонтали происходит не так удобно, поэтому горизонтальной полосы прокрутки быть не должно. Из чего следует, что веб-страница должна вписываться в окно браузера по ширине, но никак не по высоте, которая может изменяться в очень широком диапазоне. На рис. 2.8 видно, что реальная высота документа не ограничена рамками браузера, но в окне показывается лишь часть страницы.



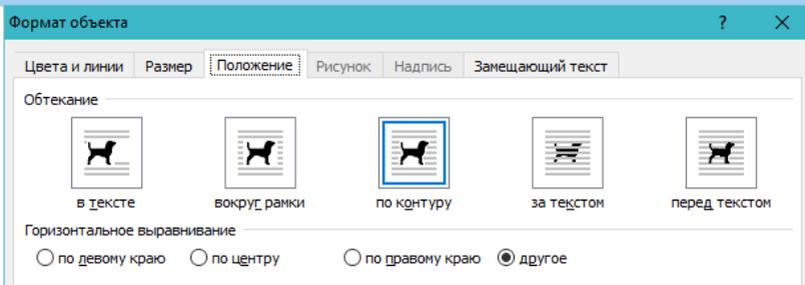
Рис. 2.8. Отображение документа в окне браузера по высоте

В принципе, чем больше на странице информации, и, следовательно, больше высота документа, тем сложнее находить нужные данные. Поэтому текст структурируют, разбивают на блоки и каждому из них дают свой заголовок, чтобы взгляду читателя было за что зацепиться.

Также следует учесть, что объем статей на сайте может достаточно сильно различаться между собой. При этом будет меняться и высота страницы, поэтому следует заранее побеспокоиться о том, чтобы макет отображался без ошибок, несмотря на различное значение высоты.

## Объекты веб-страницы прямоугольны

Все объекты на веб-странице имеют только прямоугольную форму. Этот простой постулат не всегда согласуется с тем, что мы видим в действительности, поэтому следует сделать пояснения. В отличие от традиционной верстки (речь идет о полиграфических материалах), где в документ можно вставлять любые объекты, в том числе и векторные фигуры произвольной формы, верстка веб-страниц имеет ряд ограничений. К числу ограничений относится и то, что добавляемые объекты прямоугольны. В том же WORD'e, например, можно сделать обтекание по контуру изображения:



Прямоугольность рисунков в web относится к их форме, но не к содержанию, благодаря чему требуемый нам дизайн можно конструировать с

помощью набора изображений. На рис. 2.6 показана картинка весьма неправильной формы (солнце с лучами). Но из-за того, что фон у этого рисунка совпадает с фоном веб-страницы, исходная прямоугольность изображения не видна.



*Рис. 2.6. Изображение на рисунке может иметь произвольную форму*

Однако стоит добавить вокруг рисунка рамку, как становится понятно, что изображение все-таки прямоугольно (рис. 2.7). Например, если включить обтекание картинки текстом, то он будет обходить ее именно по границе рамки.



*Рис. 2.7. Но сам рисунок остается прямоугольным*

Данная особенность породила некоторые техники связанные с версткой, которые перечислены далее.

### **Активное использование рисунков**

Рисунки не только применяются для иллюстрации текста, но и выполняют на сайте много различных ролей, например, используются для создания привлекательного дизайна, служат распоркой между ячейками таблицы, создают градиентные заливки, фоновые изображения и т.д.

### **Разрезание изображения на фрагменты**

Один рисунок может занимать слишком большую исходную площадь. Разрезав его на прямоугольники, получим конструктор, в котором одни фрагменты изображения допускается заменять текстом, другие – анимацией, а третьи модифицировать «на лету». Таким образом, имеем в своем распоряжении средство для обхода прямоугольной природы изображений, ведь в «склеенном» рисунке может не хватать некоторых фрагментов, углов, например.

### **Применение фонового рисунка**

Фоновый рисунок удобен тем, что он может заполнять всю отведенную ширину или высоту под блоком. Это позволяет создавать линии или другие декоративные элементы, которые привязываются к ширине или высоте текста и не зависят от размеров окна. К тому же поверх фона можно накладывать текст, что также расширяет возможности по дизайну веб-страниц.

### **Картинки вместо текста**

Если средства верстки имеют определенные ограничения, то почему бы не создать текст в графическом редакторе и не вставить его в качестве картинки или Flash? Это гарантирует, что текст сохранит свой вид и начертание несмотря ни на какие внешние условия. Однако здесь имеется и обратная сторона – рисунки занимают больший объем, чем рядовой текст, их сложнее править, они не индексируются поисковыми машинами, их показ пользователи могут отключить. В общем, недостатков масса, поэтому изображения на сайте хотя и применяют вместо текста, но достаточно ограниченно. Например, для заголовка сайта.

## **МОДУЛЬНАЯ СЕТКА**

Модульная сетка представляет собой набор невидимых направляющих, вдоль которых располагаются элементы веб-страницы. Это облегчает размещение данных в документе, обеспечивает визуальную связь между отдельными блоками и сохраняет преемственность дизайна при переходе от одной страницы к другой.

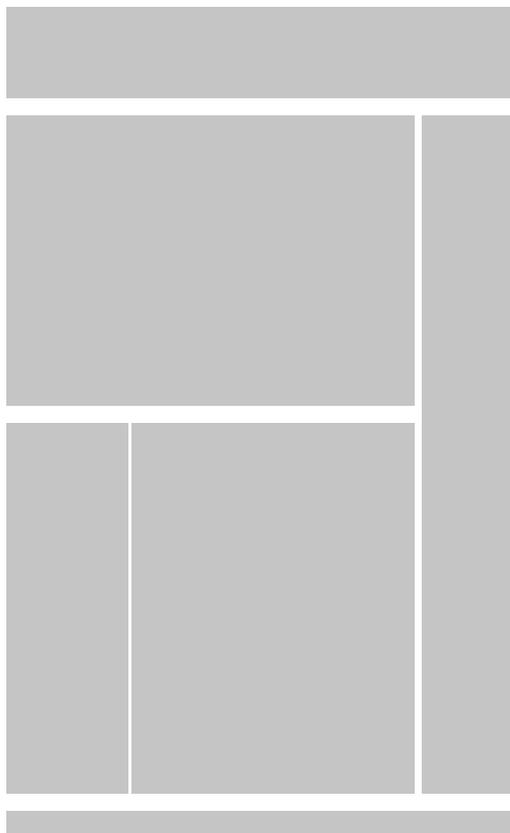
Веб-страница фактически рассматривается как набор прямоугольных блоков, которые выкладываются в определенном порядке. При этом, как правило, данные располагаются по колонкам, поэтому при верстке применяют

термин одно-, двух-, трехколонный макет и т.д. Для примера рассмотрим главную страницу сайта deviantART.com (рис. 2.8).



Рис. 2.8. Главная страница deviantART.com

Каждый блок этой страницы четко отделен от других с помощью пустого пространства, рамки или разделителя, в качестве которого выступает цветной прямоугольник с текстом заголовка. Возникает вопрос, действительно ли мы имеем дело с тремя колонками? В некоторых случаях сразу определить, сколько колонок содержит макет, бывает затруднительно. В таком случае следует понимать, что колонки могут объединяться, а также содержать не только сплошной текст, но и графические вставки. Если представить основные блоки страницы в виде однотонных прямоугольников, то получим наглядную модульную сетку, по которой сразу становится понятно, как сверстан документ (рис. 2.9).



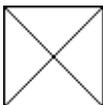
*Рис. 2.9. Модульная сетка для главной страницы deviantart.com*

По данному рисунку видно, что верхний блок с названием сайта, формой поиска и кнопками навигации занимает всю ширину страницы. Далее идут три колонки, причем первые две колонки предварительно объединены в одну для удобного представления необходимой информации. Внизу завершает макет блок с контактной и правовой информацией.

Перейдем к принципам построения модульной сетки. Вначале макет веб-страницы разрабатывают на листе бумаги. Это позволяет, не тратя зря времени, быстро сделать серию набросков и уже из них выбрать подходящий эскиз. Сколько вариантов можно создать за десять минут в графическом редакторе и сколько за это же время с помощью карандаша и бумаги? При этом не имеет значения степень владения программой, все равно на бумаге выйдет быстрее. Тем более что набросок может быть и корявым, главное чтобы автор сам понял, что он нарисовал. Обычно вместо текста и рисунков применяют схематические значки. Например, текст обозначается несколькими горизонтальными линиями (рис. 2.10), а рисунки изображаются затемненными блоками или перечеркнутыми прямоугольниками (рис. 2.11).



*Рис. 2.10. Обозначение текста в макетах*



*Рис. 2.11. Обозначение изображений в макетах*

Далее эти обозначения будем применять при рассмотрении наиболее распространенных модульных сеток.

### **Одноколонная сетка**

Текст в одну колонку чаще всего встречается в академическом дизайне, при фиксированном макете и публикации большого текста.

#### **Замечание**

Академический дизайн характеризуется минимализмом оформления и даже аскетизмом. Основной упор делается на содержательную часть, а дизайну как таковому практически не уделяется внимания. Преимущественно академический дизайн встречается в научной среде.

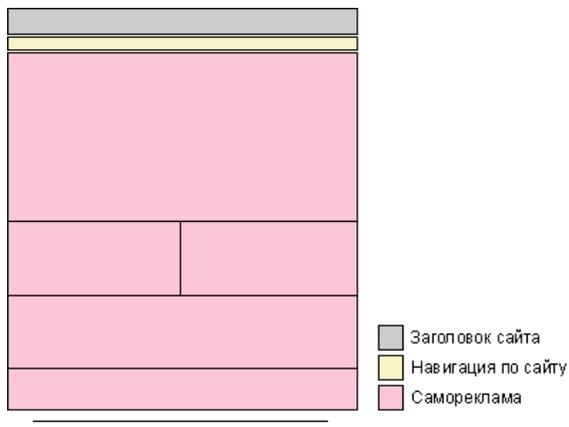
На рис. 2.12 показана типичная схема одноколонной модульной сетки. Как правило, наблюдается четыре основных блока – заголовок страницы, набор ссылок на другие страницы сайта (навигация), собственно сам текст и в самом низу контактная информация. Если высота страницы достаточно велика, то блок навигации дублируют внизу или делают ссылку «Наверх», которая перемещает к началу документа.



*Рис. 2.12. Одноколонная модульная сетка*

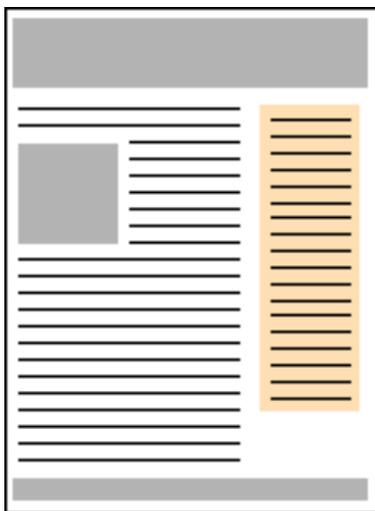
Иллюстрации в тексте встречаются по ходу, при этом текст их обычно обтекает ~~по контуру~~ вокруг рамки.

При активном применении изображений на сайте удобнее воспользоваться фиксированным макетом, ширина которого точно известна. Тогда рисунки можно готовить заданного размера, которые точно впишутся в макет страницы фиксированной ширины и практически полностью состоит из набора рисунков. Модульная сетка для этого сайта представлена на рис. 2.13. Вначале идет заголовок сайта с формой поиска, ниже следует навигация, а отдельные фотографии представляют собой саморекламу со ссылками на соответствующие разделы сайта. В самом низу расположены ссылки на информацию о сайте.



### Двухколонная сетка

Это один из самых распространенных вариантов при использовании на сайтах. При такой модульной сетке используется две колонки – одна отводится под основной текст, а вторая используется для навигации и другой полезной информации (рис. 2.14).



*Рис. 2.14. Двухколонная модульная сетка*

Принципиального значения не имеет, слева или справа располагается колонка с навигацией, встречается и тот и другой вариант.

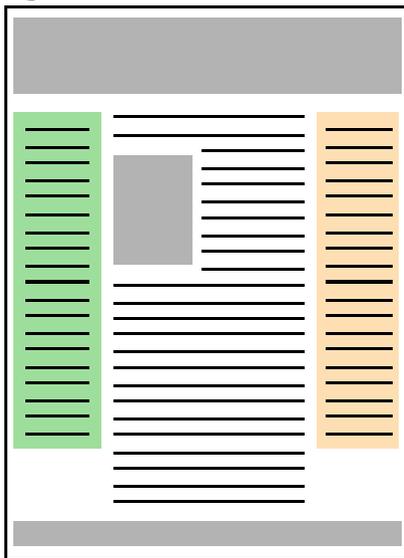
В каком-то смысле двухколонная сетка стала стандартом де-факто для информационных сайтов из-за своего удобства. Действительно, все «под

рукой» – и текст и ссылки, к тому же данная сетка не исключает подключения горизонтальной навигации, как это принято в одноколонной сетке.

Двухколонные сетки удобны при создании самых разнообразных сайтов и не требуют особых знаний по верстке веб-страниц. Единственный недостаток, который им вменяют, что подобные сайты выглядят достаточно однообразно. Но с другой стороны пользователям удобнее работать с сайтом привычного вида, без лишних «наворотов».

### Трехколонная сетка

Такие сетки часто применяются на главных страницах сайтов, где одновременно требуется показать пользователю множество возможностей, которые он обнаружит на данном сайте. Также трехколонная сетка используется и на внутренних страницах, если для размещения различной информации двух колонок уже не хватает (рис. 2.15).

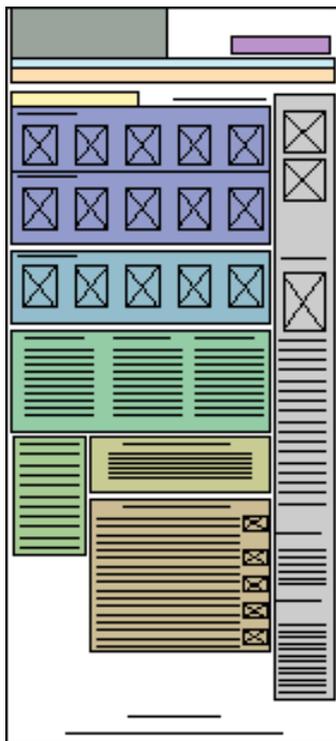


*Рис. 2.15. Трехколонная модульная сетка*

Одна из колонок отдается под навигацию, вторая, самая широкая – под основной текст, а в третью колонку добавляют рекламу, ссылки, текст и т.д.

Трехколонная сетка обеспечивает больше простора для дизайна, ведь в некоторых местах можно объединять колонки, разбивать материал на отдельные фрагменты и визуально отделять один блок от другого. Макет при

этом может получиться достаточно сложным, но результат обычно того стоит. Возвращаясь к главной странице сайта deviantart.com, приведем более детальную модульную сетку (рис. 2.16). Отдельные блоки выделены разным цветом.



*Рис. 2.16. Модульная сетка для главной страницы deviantart.com*

На сайте deviantart.com применяется три колонки, две из них часто объединяются для получения более широкой области. Это оправданно, поскольку требуется разместить 5 фотографий или три колонки с текстом.

К недостаткам трехколонной сетки относится достаточная сложность верстки макета. Чтобы получить нужный результат приходится иной раз затратить много времени на создание стилевого файла и отладку документа в разных браузерах.

Конечно, существуют и другие виды модульных сеток, например, многоколонные. Однако они практикуются достаточно редко из-за

ограниченной ширины окна браузера. При таком раскладе текст приходится делать мелким или применять другие способы, чтобы не возникло горизонтальной полосы прокрутки. От этого может пострадать удобство восприятия информации пользователем.

Модульная сетка не является единственным вариантом при верстке веб-страниц. Можно воспользоваться перспективой, хаотичностью или другой структурой, главное, чтобы она была эстетичной и привлекательной. Отказ от правил верстки характерен для дизайн-студий, задача которых состоит в том, чтобы поразить своим сайтом воображение посетителей и привлечь к себе их внимание. Но даже они делают для заказчика сайты на основе рассмотренных выше 3-х типов формальных сеток, поскольку так проще добавлять информацию и вносить разнообразие в дизайн макета.

Модульные сетки позволяют упростить верстку сайта, поскольку все материалы разбиваются на отдельные блоки, которые выравниваются по невидимым направляющим линиям. Такие блоки хотя и взаимосвязаны друг с другом, но обычно позволяют независимое редактирование данных, что упрощает оформление элементов.

Прежде чем переходить к созданию веб-страницы в HTML-редакторе, сделайте ее набросок на листе бумаги. Это позволит сократить время на выбор модульной сетки, которая будет применяться в дальнейшем. В зависимости от целей сайта и количества материала применяют одно-, двух-, трех или многоколоночную модульную сетку. Чем больше колонок применяется, тем шире возможности по управлению видом дизайна, но за это приходится платить сложностью верстки документа. Не стоит также забывать, что наиболее популярными на сайтах являются двух и трехколоночные сетки.

### **Вопросы для закрепления**

- 1) Чем отличается логическая и физическая структура сайта
- 2) Дайте понятие фиксированному макету.
- 3) Дайте понятие «резиновому» макету.
- 4) Какие виды модульных сеток вы изучили?

Домашнее задание: Составить схему (модульную сетку) для будущего сайта.

## Тема 7

### Тема: Принципы компоновки и алгоритм создания web- сайта

**Цели:** *Ознакомить учащихся с принципами компоновки сайта: статической и динамической; основными компонентами web – сайта.*

План урока.

1. Организационный момент.
2. Изложение нового материала.
  - 1) Принципы компоновки сайта
  - 2) Динамический сайт
  - 3) Статический сайт
  - 4) Компоненты сайта
3. Закрепление изученного.
4. Итог урока.

Ход урока.

#### **Организационный момент.**

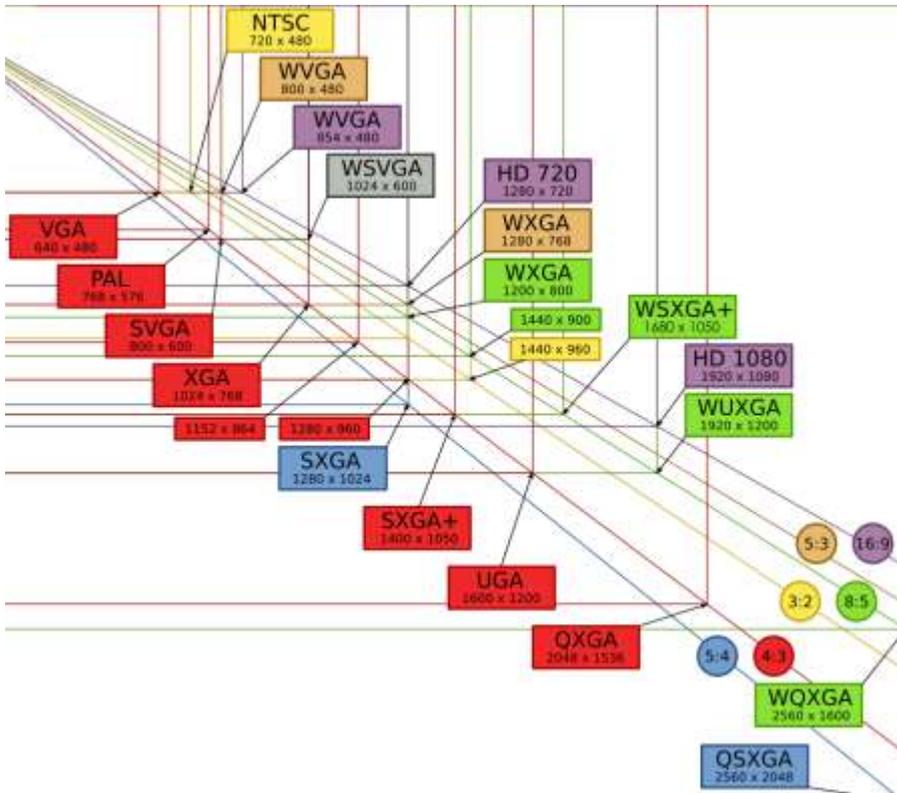
Отметить присутствующих. Сообщить тему урока. Настроить учащихся на восприятие материала.

#### **Изложение нового материала.**

### **Принципы компоновки web- сайта**

Современные видеокарты поддерживают несколько произвольно устанавливаемых видеорежимов, характеризующихся разным экранном разрешением и количеством цветов, используемых при отображении информации на экране компьютера. С помощью встроенных функций операционной системы пользователь может установить несколько стандартных значений экранного разрешения, например 640x480, 800x600, 1024x768, 1152x864, 1280x1024 или 1600x1200 точек.

Разумеется, при открытии в одном и том же браузере какой-либо web-страницы она может отображаться не одинаковым образом в зависимости от используемого посетителем сайта экранного разрешения.



Для того чтобы избежать "съезжания" элементов html-документа друг относительно друга и, как следствие, деформации web-страницы в целом при изменении параметров экрана, применяется достаточно простой и действенный прием: все компоненты web-страницы заключаются в соответствующие ячейки невидимой таблицы, при этом каждому объекту назначается одно, строго определенное положение. Таким образом, появляется второй критерий, по которому можно разделить все существующие web-сайты на две условные категории. Данной таблице можно назначить строго определенную ширину в пикселях, например, 640 точек, после чего жестко позиционировать ее по центру экрана или "прижать" к левому его краю.

*Для построения сложных таблиц можно применять вложение одной таблицы в другую: например, первичная таблица шириной 640 точек задает горизонтальное разбиение страницы, в каждую из ее ячеек вкладывается еще одна, равная ей по ширине невидимая таблица, обеспечивающая вертикальное разбиение. Такой подход позволяет дробить различные участки web-страницы*

*независимо друг от друга, создавая, к примеру, в одном ее поле четыре вертикальные колонки различной ширины, а в другом – две, оперируя при этом различными параметрами вложенных таблиц в пределах одного и того же документа – размером пространства между колонками, толщиной границ и т. д.*

Такой вариант компоновки сайта можно назвать **статическим**, поскольку ширина таблицы не меняется в зависимости от экранного разрешения. Разумеется, при изменении параметров экрана не происходит ни малейшего смещения элементов дизайна страницы.

Иной подход - когда ширину невидимой таблицы, содержащей фрагменты web-страницы, задают в процентах от текущей ширины экрана. При увеличении экранного разрешения таблица "растягивается" по горизонтали, и все размещенные в ее ячейках элементы, позиционированные либо по центру, либо по краям столбцов, смещаются согласно установленному алгоритму. В силу того, что параметры таблицы изменяются в зависимости от настроек экрана, такой принцип компоновки html-документа можно назвать **динамическим**. И тот и другой подход обладает как достоинствами, так и недостатками, которые перечислены ниже.

#### **Статическая компоновка страницы**

Достоинства. Простота алгоритма верстки документа. Оптимизировав страницу для отображения с экраным разрешением 640x480 точек, вы можете быть уверены, что при изменении пользовательских экранных настроек элементы дизайна не "поплывут". Кроме того, данный вариант компоновки сайта в большинстве случаев (при соблюдении ряда дополнительных условий) позволяет добиться идентичности отображения html-документа в различных браузерах.

Недостатки. При отображении документа на компьютере с высоким экраным разрешением по краям экрана или с правой его стороны остается заметное пустое поле.

#### **Динамическая компоновка страницы**

Достоинства. Документ растягивается по всей ширине экрана, не остается не-использованных пустых полей.

Недостатки. Сложность верстки и отладки страницы – весьма часто проявляется неадекватность отображения таких документов в различных браузерах.

**ПРИМЕЧАНИЕ** Используемые в командах HTML значения пикселей подразумевают физические размеры точки экрана пользователя и зависят от типа его монитора.

Примеры реализации сайтов со статической и динамической компоновкой страниц приведены на рис. 3.1 и 3.2. Данные изображения сделаны с фиксированным экранном разрешением 800x600 точек.

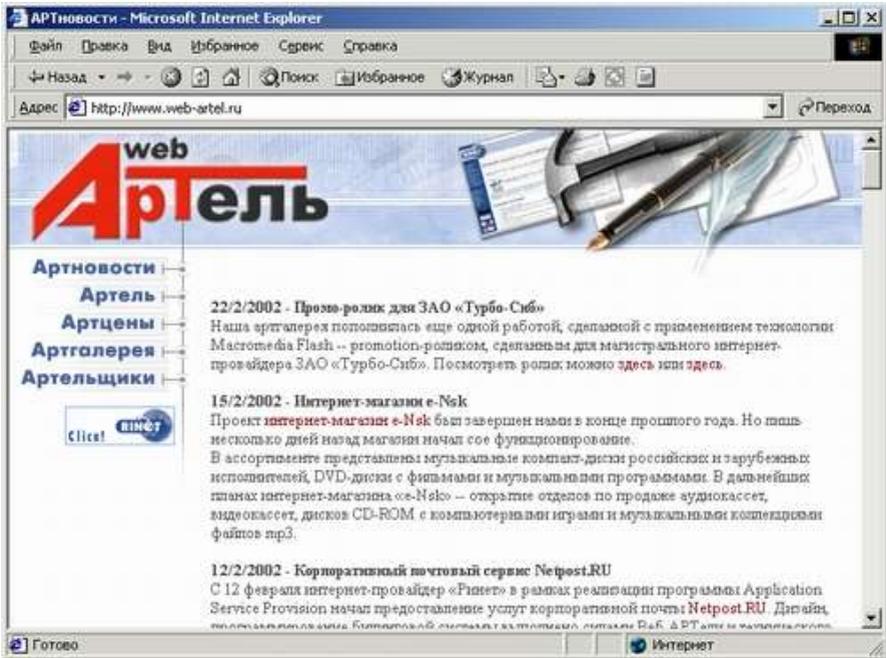


Рис. 3.1. Пример сайта со статической компоновкой страниц



Рис. 3.2. Пример сайта с динамической компоновкой страниц

### *Элементы web-страницы*

Любая web-страница содержит определенный набор стандартных элементов, являющихся обязательными компонентами каждого ресурса Интернета. Безусловно, ассортимент и количество подобных объектов могут варьироваться в зависимости от тематической направленности сайта, объема опубликованных на нем материалов, а также от целей и задач, которые ставит перед собой создатель данного ресурса. Компоновка таких элементов, проектирование их взаимного расположения и составляет одну из главных задач web-мастера.

*Первым* элементом web-страницы, который нам предстоит рассмотреть, является ее заголовок. Он может быть выполнен как в текстовом, так и в графическом варианте, однако и в том и в другом случае он должен располагаться в верхней части документа. Иногда с заголовком совмещают меню выбора кодировки кириллицы и кнопки для перехода с русскоязычной на англоязычную версию сайта, если данный web-ресурс представлен на двух языках. Непосредственно под заголовком документа, как правило, располагается пространство, отведенное для размещения рекламного баннера. Включение баннера именно в верхнюю часть web-страницы в большинстве случаев является обязательным условием регистрации сайта в службах

баннерного обмена – системах, рекламирующих созданный вами ресурс в обмен на показ на страницах вашего сайта рекламы других участников баннерообменной сети. Стандартный размер баннеров, публикуемых под заголовком документа, составляет обычно 468x60 точек. Типичный баннер выглядит так:



Код для его вставки будет выглядеть примерно так:

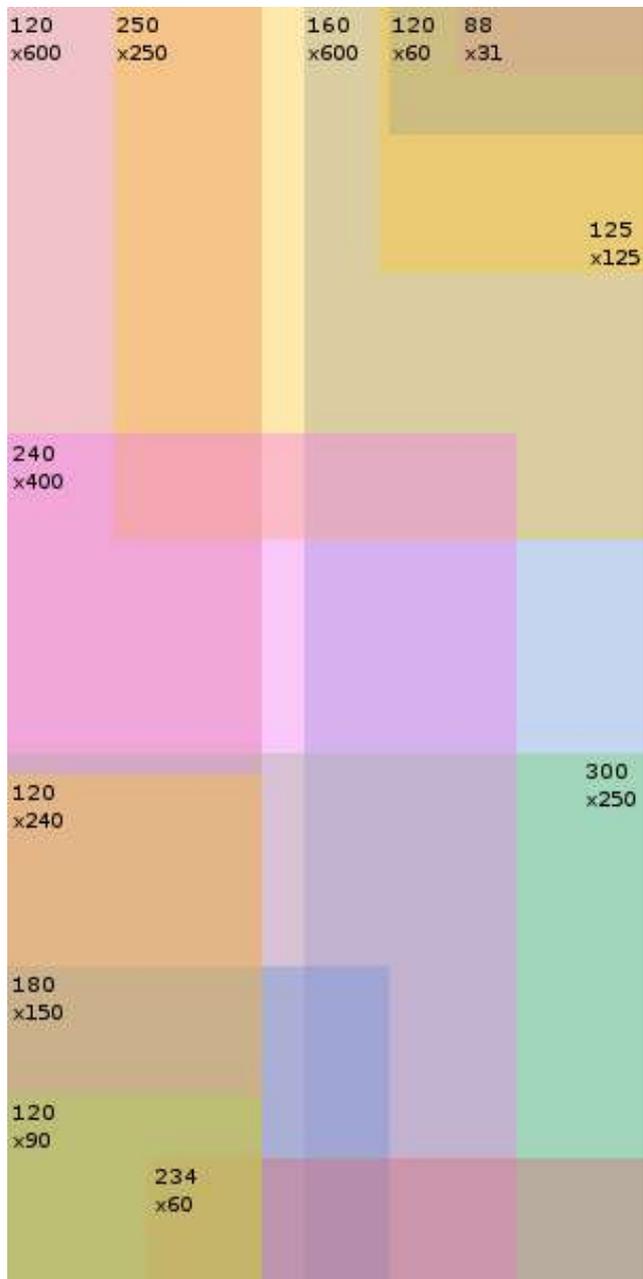
```
<a target="_blank" href="http://webeffector.ru/">

</a>
```

#### **Типы и размеры баннеров.**

- Горизонтальные прямоугольники: 728 x 90, 468 x 60, 234 x 60, 88 x 31, 120 x 60, 120 x 90.
- Вертикальные прямоугольники: 300 x 600, 160 x 600, 120 x 600, 120 x 240, 240 x 400.
- Квадраты: 250 x 250, 125 x 125.
- «Почти квадраты»: 336 x 280, 300 x 250, 180 x 150.

А вот как они выглядят:



Если вы используете статический принцип компоновки страницы, ширина заголовка вашего документа будет составлять приблизительно 640 пикселей: это значение обусловлено, прежде всего, необходимостью обеспечить корректное отображение документа на мониторах с экраным разрешением 640x480 точек и избежать появления горизонтально полосы прокрутки, затрудняющей его просмотр. Очевидно, что ширина баннера в этом случае будет значительно меньше ширины заголовка, благодаря чему в той части страницы, где вы планируете отвести место под рекламу, образуется незанятое пространство. Его можно заполнить логотипом компании-владельца данного сайта или ссылкой на сервер, осуществляющий web-хостинг. Разумеется, логотип необходим далеко не всегда: как правило, он включается в состав web-страницы лишь в случае, если сайт имеет коммерческую направленность.

**Основную часть** документа занимает так называемое текстовое поле - участок, где и размещается смысловое наполнение страницы: содержательный информационный текст и иллюстрации. Перечисленные элементы еще называют "контент" (от англ. content - содержание). Расположение текстового поля зависит в первую очередь от того, каким образом web-дизайнер разместит остальные элементы документа.

**Следующей** обязательной составляющей частью web-страницы являются элементы навигации - гиперссылки, связывающие данный документ с другими разделами сайта. Элементы навигации могут быть выполнены в виде текстовых строк, графических объектов, то есть кнопок, либо активных компонентов, например Java-апплетов. Последние представляют собой те же кнопки, которые, в отличие от своих "традиционных" сестер, умеют реагировать на движения мыши, выполняя при наведении на них курсора какие-либо несложные действия (включение подсветки, создание эффекта "нажатия", изменение формы и т. д.). Как мы уже говорили, располагать элементы навигации следует таким образом, чтобы они всегда были "на виду", "под рукой", то есть так, чтобы пользователю не приходилось "отматывать" страницу назад, если текстовое поле занимает по высоте несколько физических экранов, после чего подолгу искать ссылки на другие разделы. Наиболее устоявшимся подходом является размещение элементов навигации у левой границы страницы.

**В нижней части** документа принято публиковать информацию о разработчиках сайта и адрес электронной почты, по которому посетители ресурса могут направить владельцам странички свои отклики, предложения и пожелания. Если web-страница является стартовым документом, в нижней ее

части также размещают счетчик посещений - небольшой сценарий, вызывающий установленный на сервере CGI-скрипт<sup>1</sup>, который фиксирует каждое открытие документа в браузере пользователей, изменяя значение индикатора счетчика. Благодаря этому web-мастер без труда определит количество посетителей, навестивших его страничку в течение какого-либо времени. Отмечу, что счетчик посещений устанавливается только на первой странице, вызываемой при обращении к сайту, в остальных документах ресурса он отсутствует. Не рекомендуется также размещать на одной странице несколько разных счетчиков.

Итак, мы разобрали все основные компоненты web-страницы и их возможное расположение относительно друг друга. Пример компоновки сайта, содержащего полный набор описанных выше составляющих, показан на рис. 3.3. В нем выбрано позиционирование элементов навигации по левой границе документа.

---

<sup>1</sup> CGI (от англ. Common Gateway Interface – «общий интерфейс шлюза») – стандарт интерфейса, используемого для связи внешней программы с веб-сервером. Программу, которая работает по такому интерфейсу совместно с веб-сервером, принято называть шлюзом, хотя многие предпочитают названия «скрипт» (сценарий) или «CGI-программа».

Позволяет использовать консоль ввода и вывода для взаимодействия с клиентом. Сам интерфейс разработан таким образом, чтобы можно было использовать любой язык программирования, который может работать со стандартными устройствами ввода-вывода. Такими возможностями обладают даже скрипты для встроенных командных интерпретаторов операционных систем, поэтому в простых случаях могут использоваться даже командные скрипты.

Все скрипты, как правило, помещают в каталог `cgi-bin` сервера. CGI является одним из наиболее распространённых средств создания динамических сайтов.



Рис. 3.3. Пример компоновки web-страницы с левым позиционированием элементов навигации

На практике часто встречаются web-сайты, в дизайне которых элементы навигации позиционированы по правой границе экрана. В этом случае текстовое поле смещается налево, остальные компоненты документа располагаются, исходя из принципа максимальной эстетичности их сочетания. Пример такого исполнения сайта показан на рис. 3.4.



Рис. 3.4. Пример компоновки web-страницы с правым позиционированием элементов навигации

Как видно из рисунка, логотип в этом случае помещен на один уровень с заголовком документа, а рекламный баннер позиционирован относительно центра страницы. При таком подходе рекомендуется выдерживать графическое оформление заголовка, логотипа и поля для размещения рекламы в едином цветовом и художественном стиле - тогда несимметричность положения данных объектов будет не столь очевидна и не станет "резать глаз" сторонникам строгой, "табличной" эстетики дизайна.

Элементы навигации можно разместить не только вблизи правой и левой границ страницы, но и в верхней части документа. Такой вариант компоновки наиболее подходит при создании домашних страничек: в этом случае все объекты страницы гармонично "вписываются" в заданную ширину невидимой таблицы, при этом подготовка самой таблицы значительно упрощается. Единственным недостатком подобного подхода является необходимость продублировать элементы навигации в нижней части документа, поскольку при вертикальной прокрутке страницы они исчезают за верхней границей экрана, и, чтобы добраться до них, пользователю придется "отматывать" экран назад, что, согласитесь, весьма неудобно. Пример

оформления страницы с верхним расположением элементов навигации показан на рис. 3.5.



Рис. 3.5. Пример компоновки web-страницы с верхним позиционированием элементов навигации

Безусловно, все, что было сказано в данном разделе, является не панацеей, а руководством к действию. Мы пытаемся изложить лишь общие принципы, которые применяются при компоновке структуры сайта, окончательное же решение всегда остается за web-мастером. Примером дизайнерского решения, не попадающим ни в одну из указанных выше категорий, может служить так называемая смешанная компоновка, примерная схема которой приведена на рис. 3.6.



Рис. 3.6. Смешанная компоновка

Как видно из рисунка, в данном примере часть управляющих элементов встроена непосредственно в заголовок страницы - речь идет о кнопках переключения между русской и английской версиями сайта, а также о кнопках навигации: это может быть ссылка на адрес электронной почты создателей ресурса, продублированная в нижней части документа, и ссылка на один из тематических разделов, например, страницу новостей. Основной блок элементов навигации позиционирован относительно левой границы документа, однако меню выбора кодировки кириллицы расположено непосредственно под рекламным баннером в верхней части страницы. Текстовое поле разделено на две несимметричные колонки, причем в правой размещены краткие анонсы составляющих ресурс тематических рубрик, включая ссылки на эти разделы.

Очевидно, что вариантов "смешанной компоновки" web-страницы может быть великое множество: конкретные решения зависят от количества составляющих ресурс разделов, объема подготовленного для размещения на сайте текста и, наконец, от фантазии самого дизайнера. Важно лишь, чтобы внешний вид сайта не вызывал нареканий у посетителей.

## Вопросы для закрепления.

1. Что такое динамическая компоновка сайта?
2. Что такое статическая компоновка сайта?
3. Назовите основные элементы web- страницы.
4. Назовите классификацию web-сайтов по признакам их компоновки.

Домашние задание: Подготовить рефераты на тему: «Типы веб-дизайна: текстовый, динамический, конструктивный, полиграфический, смешанный». «Правила профессионального веб-дизайна».

## Тема 8

### Тема: Графические изображения. Особенности веб-графики

**Цели:** *Ознакомить учащихся с основными понятиями графических форматов; возможностями оптимизации графических изображений;*

План урока.

1. **Организационный момент.**
2. **Изложение нового материала.**
  - 1) Виды графических форматов их особенности и отличия
  - 2) Оптимизация графических объектов
3. **Закрепление изученного.**
4. **Итог урока.**

Ход урока.

#### **Организационный момент.**

Отметить присутствующих. Сообщить тему урока. Настроить учащихся на восприятие материала.

#### **Изложение нового материала.**

### Графика для web

Оформление **Web-страниц** подразумевает необходимость размещения максимально возможного объема информации на сравнительно небольшом пространстве. Вы должны исходить из того, что до сих пор встречаются пользователи, которые работают в Интернете с разрешением экрана 800x600. Область, которая отображается в окне браузера при таком разрешении экрана, составляет примерно 790x430 пикселей. Чтобы представить величину этой области, вспомните, что значки на рабочем столе Windows имеют размер 32x32 пиксела.

Подготавливая графику для Web-страницы, необходимо помнить о небольших размерах области, с которой вы должны работать. Изображение не должно превышать 700 пикселей по ширине и 400 пикселей по высоте. Но файл даже такой картинке может оказаться слишком большим для передачи по каналам связи сети. Следует также помнить и о том, что часть пользователей могут видеть на своих мониторах только 256 цветов. Поэтому 24-битная графика, способная обеспечить 16,7 миллиона оттенков цвета, не сможет быть оценена некоторой частью зрителей. При этом пользователи не только не смогут увидеть на своих мониторах всей заложенной вами цветовой гаммы, но, кроме того, изображение может оказаться худшего качества, чем в случае

использования 8-битной графики, отображающей 256 цветов. К тому же файл окажется в три раза больше по размеру, так как, его размер пропорционален глубине цвета, и, соответственно, будет требовать большего времени для загрузки на экран.

### **Цветовая модель**

Цвета в природе редко являются простыми. Большинство цветов получаются смешением каких-либо других. Например, сочетание красного и синего даёт пурпурный цвет, синего и зелёного – голубой. Таким образом, путём смешения из небольшого количества простых цветов, можно получить множество (и при чём довольно большое) сложных (составных). Поэтому для описания цвета вводится понятие цветовой модели – как способа представления большого количества цветов посредством разложения его на простые составляющие.

Возникает естественный вопрос – не проще ли было взять и представить в цветовой модели не основные, а все возможные цвета? Оказывается, нет. Дать описание каждого цвета в отдельности очень сложно, особенно сейчас, когда на экране монитора мы имеем возможность видеть не сотни, не тысячи, а 4 миллиарда цветов и цветовых оттенков! Таким образом цветовые модели – это способ для описания цветов особенно в компьютерных технологиях и полиграфии. Однако не любой цвет можно представить в виде комбинации основных. Это является основной проблемой цветковых моделей. Кроме того, излучаемый и поглощаемый цвет описывается по-разному.

Перед тем как перейти к рассмотрению цветковых моделей в отдельности, рассмотрим сначала понятие цветового охвата или диапазона (рис. 17), который даст нам представление о том, насколько та или иная цветовая модель хорошо представляет цвета.

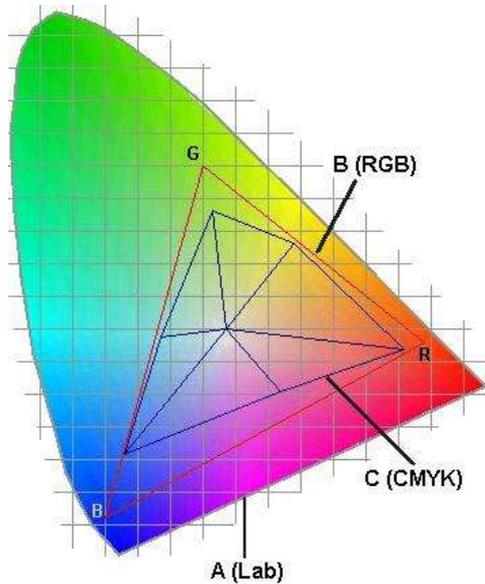


Рис.17. Цветовой охват

Цветовым охватом называется диапазон цветов, который может быть воспроизведён, зафиксирован или описан каким-либо образом. Определённым цветовым охватом обладают: электронно-лучевая трубка монитора или телевизора, цветовые модели, полиграфические краски и, конечно же, глаз человека. На рис. 17 схематически показано сравнение цветовых охватов человеческого глаза (А), монитора (В) и печатающей машины (С).

Цветовой охват монитора соответствует модели RGB, печатающей машины – CMYK.

Итак, цвет в компьютерных технологиях, в типографии, во многих других отраслях производства, связанных с обработкой изображения, представляется в виде комбинации небольшого количества трёх составных. Такое представление называется цветовой моделью. Различные виды моделей имеют различные цветовые охваты. В этом и заключается их основные преимущества или недостатки. Отражённый и поглощаемый цвет описывается по-разному. Несмотря на довольно большое количество цветковых моделей, остановимся лишь на тех, которые наиболее часто используются в графических пакетах.

Эта модель описывает излучаемые цвета. Она основана на трёх основных (базовых) цветах: красный (Red), зелёный (Green) и синий (Blue). Остальные цвета получаются сочетанием базовых. Цвета такого типа называются аддитивными.



Рис.18. Аддитивная цветовая модель RGB

Из рис.18 видно, что сочетание зелёного и красного дают жёлтый цвет, сочетание зелёного и синего – голубой, а сочетание всех трёх цветов – белый.

#### Варианты моделей RGB

Цветовая модель sRGB. Существуют различные варианты моделей RGB. Среди них модель стандартного цветового пространства для Интернета – sRGB (так называемое standard RGB – стандартное RGB). По инициативе двух фирм – Microsoft и HP – оно стандартизировано и соответствует цветовому пространству типичного монитора VGA низшего класса.

Другие варианты RGB – схемы с большим цветовым охватом (рис.19):

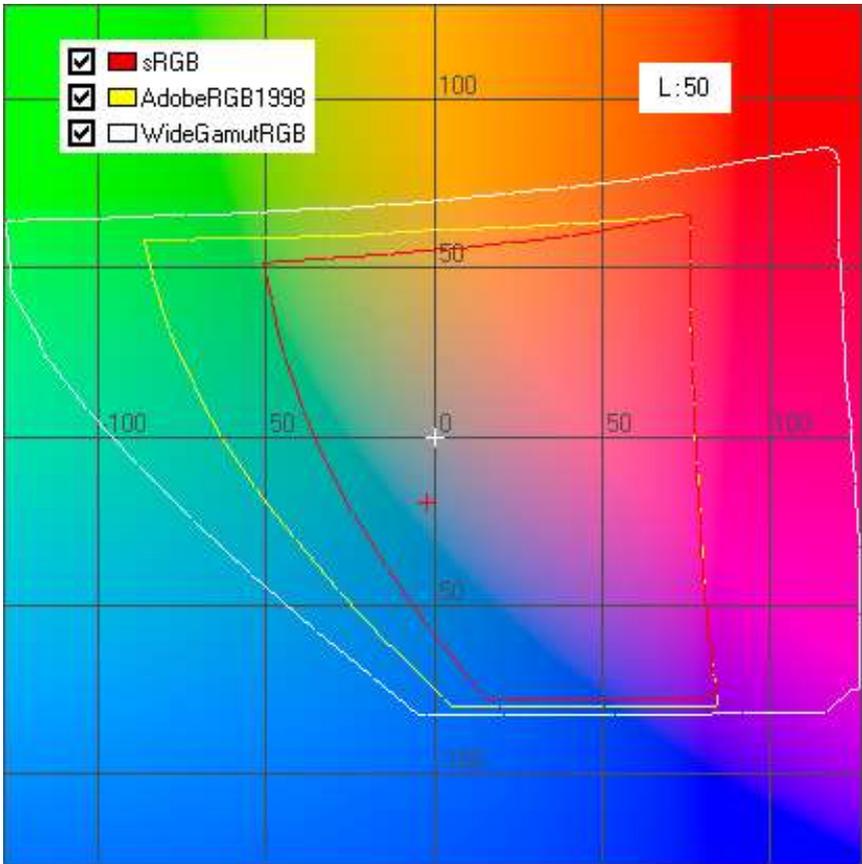


Рис. 19. Ширина цветового охвата различных моделей RGB

при 50% яркости

**Adobe RGB (1998)** – основано на одном из стандартов, предложенных для телевидения высокой четкости (High Definition TV, HDTV);

**Wide-Gamut RGB** (RGB с расширенным диапазоном) – основано на чистых значениях для красного, зеленого и синего цветов, обладает очень широким охватом, который может быть представлен лишь в 48-разрядных файлах изображений;

#### Кодирование цвета в модели RGB

В программах для ПК канал изображения кодируется одним байтом. Чтобы понять, что такое канал изображения, обратимся к примеру. В RGB три

канала: красный, синий и зелёный, т.е. RGB – трёхканальная цветовая модель. Каждый канал может принимать значения от 0 до 255 в десятичной или, что ближе к реальности, от 0 до FF в шестнадцатиричной системах счисления. Это объясняется тем, что байт, которым кодируется канал, состоит из восьми битов, а бит может принимать 2 значения, итого  $2^8 = 256$ . В RGB, например, красный цвет может принимать 256 градаций: от чисто красного (FF) до чёрного (00). Таким образом несложно подсчитать, что в модели RGB содержится всего 2563 или 16 777 216 цветов.

Теперь рассмотрим модель RGB в пространстве. На рис. 20 изображено

пространственное представление модели RGB. В трёх углах куба расположены чистые цвета: красный, зелёный и синий. В других трёх углах их полные сочетания: жёлтый, голубой и пурпурный. Между чёрным и белым цветами проведена диагональ, изображающая градацию серого.

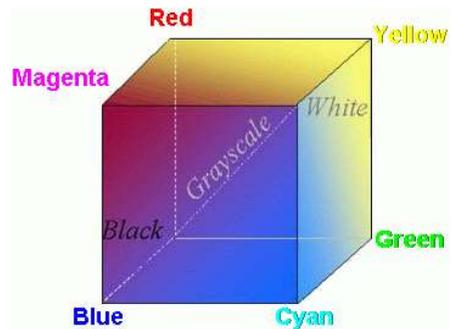


Рис.20. Модель RGB

Расскажем подробнее о кодировании, теперь уже применительно к RGB. Мы уже знаем, что в RGB три канала, и каждый кодируется 8-ю битами. Максимальное, т.е. FF (или 255) значение даёт чистый цвет. Мы знаем также, что белый цвет получается путём сочетания всех цветов, точнее, их предельных градаций. Теперь мы можем записать код белого цвета: FF(красный) FF(зелёный) FF(синий). Код чёрного, соответственно: 000000. Код жёлтого: FFFF00, пурпурного: FF00FF, голубого: 00FFFF.

Примеры кодов цветов показаны на рис. 21.

	black = "000000"		green = "008000"
	silver = "C0C0C0"		lime = "00FF00"
	gray = "808080"		olive = "808000"
	white = "FFFFFF"		yellow = "FFFF00"

	maroon = "800000"		navy = "000080"
	red = "FF0000"		blue = "0000FF"
	purple = "800080"		teal = "008080"
	fuchsia = "FF00FF"		aqua = "00FFFF"

Рис. 21. Примеры цветов по схеме RGB

Для задания цвета в программах используют двумерное представление трехмерного цветового пространства в виде селектора цветов – color picker (рис.22).

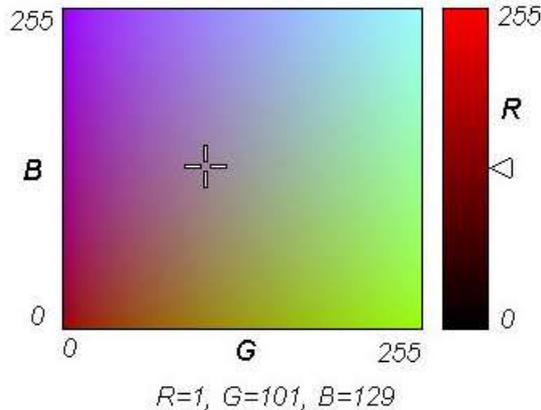


Рис. 22. Двумерный способ задания цвета в RGB

Использование корректной цветовой гаммы важно при разработке любого графического проекта. Для графики Web-страниц необходимо применять цвета модели RGB. Так как для Web-страниц устройством вывода графики является экран монитора, то здесь, как правило, соблюдается условие WYSIWIG - What You See Is What You Get (Что видишь, то и получаешь).

На вопрос, «Каков должен быть размер графического файла?», предназначенного для Web, трудно ответить однозначно. Представим ситуацию, что пользователь, обращается к вашему сайту на скорости 64 Кбит/сек

(провайдер ограничил скорость за неуплату, или мобильная передача данных за городом, где нет 3G/4G, работает только в режиме EDGE). При использовании такой скорости передача блока данных объемом 8 килобайт занимает примерно 1 секунду, так что при объеме графики в 240 килобайт для ее передачи по сети потребуется приблизительно 30 секунд. После таких расчетов возникает следующий вопрос: как долго посетитель, обратившийся к вашему сайту, готов ждать появления на экране загружаемого изображения? Если вы сможете ответить на этот вопрос, то сразу же определите и максимальный размер файла с графическим изображением. Считается, что файл должен иметь такой размер, при котором время его загрузки по сети не превышает одной минуты, что выражается в ограничении размера величиной примерно 60 килобайт.

Размер графического файла, в частности, полученного в результате сканирования изображения, можно определить по следующей формуле:

$$F = (A/25,4 \times B/25,4) \times D^2 \times C/8,$$

где

F - размер файла без сжатия в байтах;

A - ширина исходного изображения в мм;

B - длина исходного изображения в мм;

D - графическое разрешение изображения в dpi;

C - глубина цвета в битах.

Таким образом, если вы хотите отсканировать для своей Web-страницы фотографию размером 10x15 см с разрешением 150 dpi и глубиной цвета 24 бита, то результирующий файл будет иметь размер 1569378 байт или 1,49 мегабайта.

Любая компьютерная информация может храниться только в определенном формате. Каждый вид информации имеет собственные форматы. Для текста используются одни форматы, для электронных таблиц - другие, для графики - третьи. Формат графической информации обычно определяется программой, в которой она создана. Для Web-графики стандартными считаются форматы JPEG, GIF и PNG.

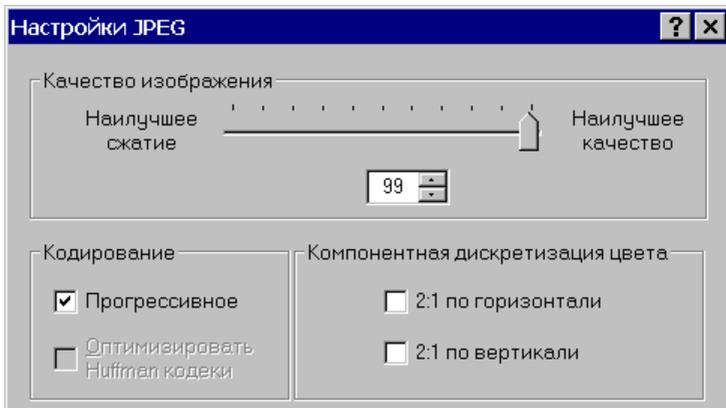
Формат **JPEG** (*джей нег* – Joint Photographic Experts Group – Объединенная группа экспертов фотографии) - самый популярный формат для хранения фотографических изображений. JPEG является стандартом в Интернете и предназначен для сохранения растровых (точечных) изображений путем сжатия, уменьшающим размер файла на десятые доли процента – и до

99%, за счет потери избыточной информации, почти не влияющей на отображение документа.



Но практически диапазон сжатия значительно уже: от 5 до 15 раз. Хотя алгоритм сжатия изображения этого формата и ухудшает качество, он может быть легко настроен на минимальные, практически незаметные для человеческого глаза, потери. Распаковка JPEG-файла происходит автоматически, во время его открытия. Между качеством изображения и степенью уплотнения существует обратная зависимость: чем более высокую степень сжатия для результирующего изображения вы зададите, тем хуже будет его качество.

Существует разновидность формата JPEG, именуемая «прогрессивный JPEG» (p-JPEG). Прогрессивный JPEG отличается от обычного тем, что при выводе на экран изображение появляется почти сразу, но плохого качества, а по мере загрузки качество улучшается.



В формате JPEG можно хранить только полноцветные изображения с глубиной цвета 24 пиксела. Алгоритм JPEG в наибольшей степени пригоден для сжатия фотографий и картин, содержащих реалистичные сцены с плавными переходами яркости и цвета. Наибольшее распространение JPEG получил в цифровой фотографии и для хранения и передачи изображений с использованием сети Интернет.

Формат JPEG в режиме сжатия с потерями малоприспособлен для сжатия чертежей, текстовой и знаковой графики, где резкий контраст между соседними пикселями приводит к появлению заметных артефактов. Такие изображения целесообразно сохранять в форматах без потерь, таких как TIFF, GIF, PNG или использовать режим сжатия Lossless JPEG.

JPEG (как и другие форматы сжатия с потерями) не подходит для сжатия изображений при многоэтапной обработке, так как искажения в изображения будут вноситься каждый раз при сохранении промежуточных результатов обработки.

JPEG не должен использоваться и в тех случаях, когда недопустимы даже минимальные потери, например, при сжатии астрономических или медицинских изображений. В таких случаях может быть рекомендован предусмотренный стандартом JPEG режим сжатия Lossless JPEG (который, однако, не поддерживается большинством популярных кодеков).

Формат **GIF** (Graphics Interchange Format - Формат для обмена графической информацией) был разработан специалистами компьютерной сети CompuServe в 1987 году с целью ускорения пересылки файлов по телефонным линиям. Требовалось создать такой формат, который бы содержал достаточно

хорошо упакованные данные. Алгоритм распаковки должен был быть при этом встроено во все программы, которые будут работать с этим форматом: утилиты просмотра, графические редакторы и т.п. Именно такой подход и реализован в формате GIF.

Хотя с тех времен пропускная способность сетей увеличилась во много раз, требования к компактности файлов, передаваемых по сетевым каналам, отнюдь не отошли на второй план. Web-страница, которая со всем своим текстом и графикой занимает много места, т.е. считается «большой» - и не каждый посетитель дожидется, пока она целиком загрузится в его браузер. Профессионализм Web-дизайнера как раз и заключается в том, чтобы отыскать должный баланс между художественностью, информативностью страницы и ее объемом. Вот почему формат GIF, главным достоинством которого является малый размер файлов, до сих пор сохраняет свое значение основного графического формата World Wide Web. Из распространенных графических форматов GIF однозначно уступает по степени сжатия только формату JPEG.

Конечно, GIF - не единственный формат, позволяющий сжимать графические данные; способности к сжатию есть у формата TIFF и многих других. Однако GIF - один из немногих форматов, использующих алгоритм сжатия (LZW), почти не уступающий по эффективности программам-архиваторам. Иными словами, GIF-файл не нужно архивировать, так как это не дает ощутимый выигрыш в объеме.

Как и у программ-архиваторов, степень сжатия графической информации в формате GIF сильно зависит от уровня ее повторяемости и предсказуемости, а иногда еще и от ориентации картинка. Поскольку GIF сканирует изображение по строкам, то, к примеру, плавный переход цветов (градиент), направленный сверху вниз (т.е. по горизонтали цвет точек одинаковый), сожмется куда лучше, чем тех же размеров градиент, ориентированный слева направо, а этот последний - лучше, чем градиент по диагонали.

В отличие от формата JPEG, который позволяет достичь серьезного выигрыша в размере при минимально заметных для глаза потерях в качестве, формат GIF пользуется обычным сжатием без потерь. С другой стороны, преимуществом JPEG является возможность хранить полноцветные изображения с 16 миллионами цветов, тогда как GIF ограничен лишь 256-цветной палитрой.

В 1989 году появилась новая версия формата GIF, в которой используется «постепенный» показ изображения по мере его загрузки из сети. Эта технология заключается в том, что при показе специально записанных GIF-

файлов строки изображения выводятся на экран не подряд, а в определенном порядке: сначала каждая 8-я, затем 4-я и т.д., то есть полностью изображение показывается в четыре прохода. Такая система показа позволяет понять, что изображено на картинке еще до того, как она полностью загрузится, что очень важно на медленных линиях связи. Нечто подобное используется в телевидении для передачи одного полного кадра за два прохода луча - так называемая «чересстрочная развертка». Поэтому такие изображения были названы чересстрочными (interlaced). Приняв из сети очередной проход чересстрочного GIF-файла, браузер не только рисует каждую его строку на своем месте, но и заполняет копиями предыдущей строки все промежутки между строками, иначе после первого прохода изображение было бы почти невидимым. Поэтому сначала чересстрочный GIF состоит из широких горизонтальных полос, которые постепенно сужаются по мере проявления изображения.

Один из цветов в палитре может быть объявлен «прозрачным». В этом случае в программах, которые поддерживают прозрачность GIF (например, большинство современных браузеров) сквозь пиксели, окрашенные «прозрачным» цветом, будет виден фон. «Полупрозрачность» пикселей (технология *альфа-канала*) не поддерживается.

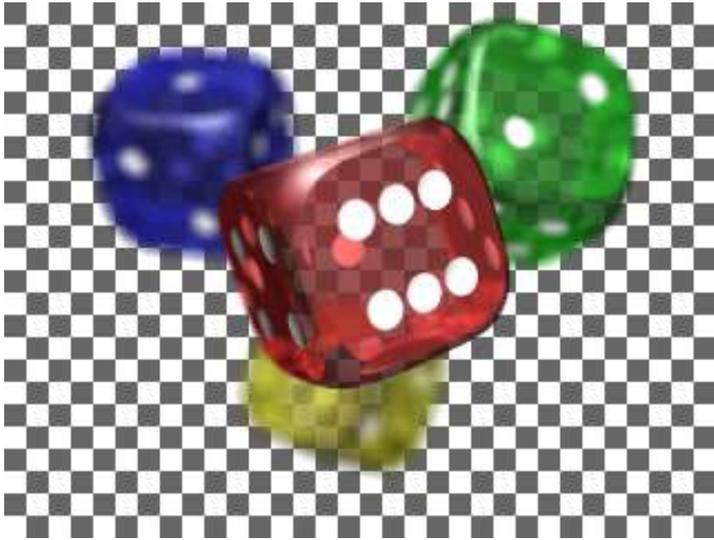
Также эта версия формата позволяет получать «анимированные gif». Они представляют собой последовательность из нескольких статичных кадров, а также информацию о том, сколько времени каждый кадр должен быть показан на экране. Анимацию можно сделать цикличной (англ. loop), тогда вслед за последним кадром начнется воспроизведение первого кадра и т.д.

Формат **PNG** (Portable Network Graphics - Переносимая сетевая графика) поддерживается поздними версиями браузеров, но широкого распространения пока не получил. Этот формат, основанный на вариации алгоритма сжатия без потерь качества, в отличие от GIF сжимает растровые изображения и по горизонтали, и по вертикали, что обеспечивает более высокую степень сжатия; кроме того, он поддерживает цветные графические изображения с глубиной цвета до 48 бит включительно.

PNG реализует гораздо более эффективный алгоритм чересстрочности. Достаточно сказать, что первый проход, дающий общее представление об изображении, занимает в PNG не одну восьмую исходного файла, как в GIF, а всего лишь одну шестьдесят четвертую – и, тем не менее, распознаваемость картинки при этом заметно лучше.

PNG позволяет хранить полную информацию о степени прозрачности в каждой точке изображения в виде так называемого *альфа-канала*. Каждый

пиксел PNG-файла, вне зависимости от его цвета и местоположения, может иметь любую градацию прозрачности: от нулевой - полная непрозрачность, до абсолютной невидимости.



Чтобы сгладить различия в яркости дисплеев, существующие между разными типами компьютеров и, соответственно, между создаваемыми на них изображениями, PNG позволяет хранить в файле исходный коэффициент яркости того дисплея, на котором изображение было создано. Это дает возможность Программам просмотра перед выводом на экран регулировать яркость изображения, приспособляя ее к особенностям данной видеосистемы – так называемая *гамма-коррекция*.

Существует одна особенность GIF, которая в PNG не реализована: поддержка нескольких изображений в одном файле, и в частности – анимации; PNG изначально был предназначен лишь для хранения одного изображения в одном файле.

Не все веб-браузеры одинаково отображают содержимое png-файла. Проблемными местами могут быть:

- частичная прозрачность (альфа-канал)
- поддержка прозрачности в палитре
- гамма-коррекция
- цветовая коррекция

### Возможности оптимизации

Чтобы любое изображение можно было использовать на Web-странице, его необходимо предварительно оптимизировать, уменьшив до минимума размер файла и, следовательно, время загрузки по сети так, чтобы качество изображения при этом заметно не ухудшилось. Оптимизация это, по сути, компромисс между размером файла и качеством изображения.

В формате JPEG изображение сохраняется с глубиной цвета 24 бит и используется сжатие с потерей информации. У него не так уж много способов оптимизации, точнее сказать, два:

- использование прогрессивной развертки;
- оптимальный коэффициент сжатия.

Использование прогрессивной развертки не сказывается на размере изображения, но заметно при загрузке. В случае записи изображения в стандартном формате вывод на экран осуществляется горизонтальными полосами, и до загрузки всего изображения невозможно определить его смысл. Напротив, при записи файла в прогрессивном формате изображение появляется сразу целиком, но в грубой форме, и затем постепенно улучшается. Это дает посетителям сайта возможность сразу же оценить изображение и решить, стоит ли ожидать его окончательной загрузки, что позволяет сэкономить немного времени при просмотре страниц. Общее время загрузки страницы при этом не уменьшается, возникает лишь иллюзия его уменьшения. Прогрессивная развертка JPEG поддерживается всеми браузерами, но не каждый графический пакет может записывать в этом формате. Adobe Photoshop умеют это делать.

Оптимальный коэффициент сжатия. Это - другая возможность оптимизации изображения в формате JPEG. Для осуществления такой оптимизации нужен графический пакет или утилита, которая позволяет регулировать степень сжатия изображения. Adobe Photoshop позволяет выполнить такую регулировку. Другие программы, например, Photo- Paint и PaintShopPro, также позволяют плавно регулировать степень сжатия в интервале значений от 1 до 255.

Как определить оптимальную степень сжатия? Для этого нужно сохранить изображение при нескольких значениях коэффициента сжатия, а затем внимательно рассмотреть его. Критерием потери качества будут так называемые «дрожащие контуры», наиболее заметные в местах контрастных переходов, и появление пятен в областях с плавными переходами. Тот коэффициент сжатия, при котором искажения еще не видны, но с увеличением коэффициента на одну ступеньку становятся заметными, и будет оптимальным.

Размер файла при таком коэффициенте сжатия будет минимально возможным при сохранении качества.

Формат GIF сильно отличается от JPEG и использует совершенно другие методы оптимизации. Он предназначен для хранения изображений с количеством цветов не более 256 и использует сжатие без потерь по методу LZW, который применяется в программах-архиваторах. Методы оптимизации GIF можно разделить на следующие типы, исключая оптимизацию файлов с анимацией:

- уменьшение количества цветов;
- оптимизация палитры изображения;
- стилизация изображения;
- изменение размеров изображения;
- фрагментарная оптимизация;
- оптимизация «прозрачных» изображений;
- использование чересстрочной развертки.

Теперь разберемся по порядку с этими методами и приемами оптимизации.

**1) Уменьшение количества цветов.** Для большинства изображений, не являющихся фотографическими, количество необходимых для воспроизведения цветов часто меньше 256. Лишние цвета можно убрать из изображения, тем самым уменьшив его размер. Это возможно потому, что формат GIF поддерживает размер палитры меньше 256 цветов. Теоретически мы можем задать любое число цветов палитры от 2 до 256, например, 45 цветов или 99. Практически же количество цветов в изображении выбирается из ряда: 2, 4, 8, 16, 32, 64, 128, 256. Это связано с тем, что для представления цвета используется всегда целое число бит, а приведенный выше ряд является рядом максимального количества цветов при использовании от 1 до 8 бит на пиксел. В случае, если вы используете в рисунке, допустим, только 97 цветов, для сохранения одного пиксела все равно будет использовано 7 бит, как и при использовании 128 цветов. Поэтому при оптимизации количества цветов нужно ориентироваться на приведенный выше ряд. Это позволит получить более качественную картинку. Уменьшив количество цветов с 256 до 32, можно иногда добиться уменьшения размера файла в 2 раза без заметной потери качества.

**2) Оптимизация палитры изображения.** Преобразование полноцветного изображения в индексное, т.е. уменьшение количества цветов до 256

является важной операцией, и здесь следует сказать несколько слов о том, как это делается, современными графическими пакетами.

Палитра конечного изображения может быть либо фиксированной, либо оптимизированной. В первом случае графический редактор просматривает каждую точку изображения и подбирает ближайшую ей по цвету из палитры. Этот способ дает самые худшие результаты с точки зрения правильности воспроизведения цветов. Попробуйте, например, преобразовать фотографию красной розы в индексное изображение, используя палитру, содержащую оттенки зеленого. Это, конечно, крайний случай, но даст вам возможность оценить качество преобразования с использованием фиксированной палитры. Тем не менее, данный способ (фикс. палитра) применяется в основном для того, чтобы изображения приемлемо выглядели на мониторах с малым количеством цветов = 256. Обычно в этом случае пользуются так называемой **безопасной палитрой**, состоящей из набора (216) часто используемых цветов и их оттенков. Безопасная палитра гарантирует, что изображения, использующие эту палитру, будут одинаково показаны всеми браузерами.

Во втором случае, когда используется оптимизированная палитра, графический редактор вначале анализирует рисунок и составляет список всех используемых в изображении цветов. Далее, на основании частоты появления цветов, составляется палитра, которая называется оптимизированной. После этого рисунок снова просматривается, и цвет пиксела изменяется на ближайший из палитры. Этот способ дает гораздо лучшие результаты, и именно его следует применять, если у вас нет причин для использования фиксированной палитры.

При оптимизации конкретного изображения задача состоит в том, чтобы выбрать оптимальное количество цветов в палитре. Здесь уместно сказать о способе увеличения количества используемых цветов при их недостатке – так называемом смешивании (дизеринг - dithering). При этом области, которые в оригинале были залиты однородным цветом, после преобразования заполняются смесью пикселей разных цветов, разбросанных по случайному закону.

Данный метод используется, когда необходимо, чтобы изображение с ограниченным количеством цветов (256 или менее) содержало большее количество оттенков. Обычно он применяется к изображениям с плавным переходом цветов, чтобы увеличить диапазон используемых оттенков, но в жизни ничего не дается даром, и поэтому после применения данного метода изображение может покрыться точками.

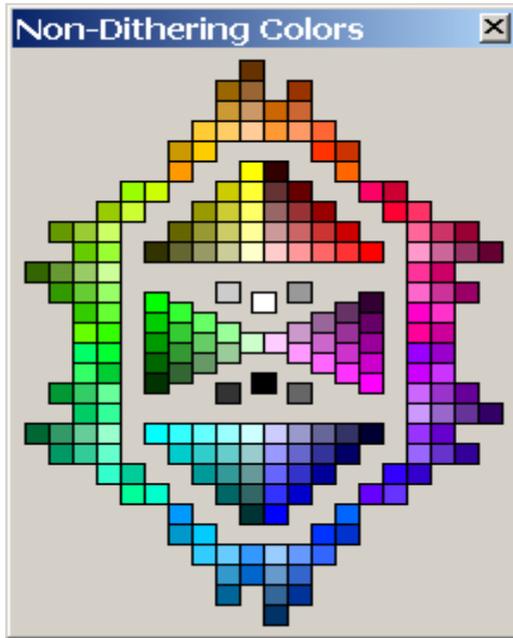
Как правило, каждый отсутствующий в редуцированной палитре цвет передается пикселями двух самых близких к нему цветов новой палитры. В

результате изображение приобретает характерную зернистую, шероховатую фактуру. Очень часто смешивание является единственным способом хоть сколько-нибудь адекватно передать исходные цвета с помощью палитры, в которой этих цветов уже нет. Например, у нас есть 16 стандартных чистых цветов, а нам нужен отсутствующий оранжевый цвет. В таком случае можно составить его из красных и желтых точек, разместив их в шахматном порядке. Издалека будет казаться, что это сплошной оранжевый цвет. Это классический пример смешивания. Его использование при преобразовании изображений может дать очень хорошие результаты. Но с точки зрения оптимизации размеров файла происходит совсем обратное. Смешивание может увеличить размер файла, причем довольно существенно - он может стать больше, чем до оптимизации. А все дело в способе хранения изображения форматом GIF. Изображение перед сохранением на диске подвергается сжатию методом LZW, а основная особенность этого метода заключается в том, что сжатию лучше всего поддаются области, заполненные однородным цветом, и хуже всего – области, состоящие из набора разноцветных точек. Смешивание же как раз основано на том, что получает недостающие цвета путем «перемешивания» точек разных цветов.

Все существующие ныне Web-браузеры вынуждены при выводе графических файлов на экран изменять их палитру. Во-первых, им приходится приводить к общему знаменателю палитру всех файлов, включенных в одну страницу. Поскольку некоторые экраны не могут отобразить больше 256 цветов одновременно, приходится идти на компромиссы. Во-вторых, в некоторых случаях браузеру приходится также приспособливать палитру файла к фиксированной системной палитре. Чтобы эти неизбежные преобразования не приводили к совсем уж неприемлемым результатам, все современные браузеры пользуются при замещении палитры смешиванием. Это значит, что даже если ваше изображение состоит из идеально гладких цветовых плоскостей, но его палитра почему-либо не может быть отображена напрямую, то в браузере это изображение неизбежно станет шероховатым.

Однако для очень многих изображений, составляющих нередко основу дизайна страницы, **смешивание (дизеринг) нежелательно**. Что же делать? Выход, хоть и не самый лучший, есть. Уже упоминался **набор из 216 цветов**, которые с гарантией никогда не подвергаются смешиванию ни в одном из браузеров, ни в Windows, ни в Macintosh. Если ваше изображение будет содержать только цвета из этого набора или, по крайней мере, к этому набору будут относиться цвета, занимающие наибольшую площадь, в браузере оно бу-

дет выглядеть намного чище и привлекательнее. Эту палитру часто называют **безопасной (browser-safe palette)**.



Воспользоваться безопасной палитрой не поздно, даже когда графика уже готова. Однако, если вы хотите иметь больший контроль над цветовой стороной своего дизайна, те графические элементы, для которых хотелось бы избежать смешивания, лучше с самого начала рисовать с использованием безопасной палитры.

Безопасная палитра поддерживается Adobe Photoshop (см. далее).

**3) Стилизация изображения.** Для уменьшения количества используемых цветов в изображениях, содержащих фотографические сюжеты, возможно применение стилизации фотографии. Для этого можно воспользоваться такими приемами, как тонирование изображения и применение художественных фильтров, имитирующих рисунки. Фотографии, обработанные таким способом, будут сжиматься лучше и требовать палитру, содержащую менее 256 цветов.

**4) Изменение размеров изображения.** Этот метод оптимизации основан на изменении размеров рисунка. Его нельзя применять к обычным рисункам, но он очень хорошо подходит для рисунков, состоящих из элементов типа линий и однородных областей. Допустим, нам понадобился рисунок,

представляющий собой черный квадрат размером 100x100 пикселей. Для уменьшения размера файла достаточно сделать так называемый однопиксельный, т.е. размером 1x1 пиксел, рисунок GIF черного цвета и, как обычно, вставить его в HTML-файл, но размеры картинки указать те, что нужны нам - 100x100. Браузер сам растянет наш пиксел до нужных размеров, и, таким образом, будет достигнута значительная экономия в объеме файла.

пример:

Подобным же образом можно использовать одно-пиксельные рисунки других цветов, но наибольшую пользу может принести прозрачное однопиксельное изображение GIF. С его помощью удобно задавать «пустышки» на страницах, абзацные отступы, фиксировать минимальную ширину или высоту таблицы и делать другие полезные вещи.

**5) Фрагментарная оптимизация.** Фрагментарная оптимизация применяется в случае, если у вас есть некоторое большое изображение, содержащее сильно отличающиеся по количеству цветов области. В этом случае изображение можно разрезать на фрагменты и поместить их в таблицу, тогда в браузере оно будет выглядеть как единое целое. Теперь каждый фрагмент исходного рисунка оптимизируется отдельно. Для фрагментов, содержащих мало цветов, вполне может хватить палитры в 8 или 16 цветов, а для богатых цветами фрагментов можно использовать полную палитру в 256 цветов или сохранять их в формате JPEG. Этот способ позволяет уменьшить объем, занимаемый исходным файлом, примерно раза в 2-3, в зависимости от конкретного изображения. Кроме того, изображение будет появляться на экране более равномерно, так как все фрагменты будут загружаться параллельно. Однако, если количество фрагментов превысит 10, то возможного выигрыша не получится, так как каждый фрагмент должен будет хранить свою копию палитры и служебной информации.

**6) Оптимизация «прозрачных» изображений.** Формат GIF позволяет сохранять так называемые «прозрачные» изображения. На самом деле, GIF не поддерживает прозрачность в изображениях - альфа-канал - он лишь позволяет назначить одному любому цвету в палитре атрибут прозрачности. Если браузер рисует на экране такое изображение, то, встречая «прозрачный» пиксел, он просто игнорирует его и не показывает на экране. Само по себе это хорошо, но в случае использования метода сглаживания краев изображения (anti-aliasing) и последующего сохранения в формате GIF, возникает проблема появления ореола вокруг прозрачного изображения в случае, если GIF показывается на

фоне, отличном от того, на котором происходило сглаживание и который стал прозрачным. Единственным способом уменьшения проявления этого эффекта является назначение прозрачности цвету, близкому к цвету фона, на котором будет использоваться GIF, и проведение сглаживания на этом же фоне.

### 7) Чересстрочный GIF89a

Для создания чересстрочной графики понадобится графический редактор, поддерживающий запись в файл формата GIF89a. Для PhotoShop это называется "File Export", "GIF89a Export". Должен быть отмечен пункт "Interlace".

### Какой формат использовать?

Вопрос о том, в каком формате хранить нужное вам изображение, не так прост, как кажется. Очень часто можно прочесть, **что фотографические изображения, поскольку они имеют более 256 цветов, следует хранить в формате JPEG, а рисунки в формате GIF.** Это правило действительно верно в большинстве случаев. Однако могут быть и исключения. В ряде случаев, **если вам требуется особое качество изображения, следует предпочесть GIF.** Дело в том, что главной помехой в использовании JPEG является то, что изображения, сохраненные в этом формате, в отличие от GIF имеют визуально заметные искажения. Они тем заметнее, чем больше величина сжатия и меньше геометрические размеры деталей изображения. **Считается, что если исходный размер файла сжимается до 10 раз то искажения будут не очень заметны.** В то же время, если в достаточно «интеллектуальной» программе уменьшить количество цветов фотографии до 256 этих 256 цветов в большинстве случаев хватит на то, чтобы изображение не проиграло в визуальном восприятии.

Поэтому в том случае, когда для вас существенно, чтобы картинка воспроизводилась с минимальными искажениями, сделайте два файла - в GIF и JPEG-форматах и посмотрите, какой файл при устраивающем вас качестве займет меньше места на диске.

Во всяком случае, имейте в виду следующие обстоятельства:

- Чем больше на рисунке мелких и контрастных деталей, тем сильнее будут заметны искажения, вносимые при сжатии в формате JPEG.
- Чем крупнее геометрические размеры изображения в пикселах и чем ниже контрастность снимка, тем искажения JPEG будут меньше заметны.
- Чем больше на рисунке контрастных и мелких деталей, тем более крупный размер файла получится при сохранении в формате GIF.

- Чем большую площадь занимают однородно окрашенные участки и чем большую горизонтальную протяженность имеют детали рисунка, тем меньший размер займет на диске файл GIF.

- Файлы, записанные в чересстрочном (interlaced) формате GIF и в прогрессивном формате JPEG, то есть так, чтобы изображение «проявлялось» постепенно, будут, при прочих равных условиях, иметь больший размер.

Но есть две области, где GIF - вне конкуренции. Первая - это создание анимированных, то есть движущихся изображений. На самом деле анимированный GIF -это несколько графических изображений, записанных в один файл. Последовательный показ этих изображений создает иллюзию движения. В файле GIF хранится информация о том, сколько раз в секунду надо менять изображение, в какой последовательности показывать картинки и т.д. Для создания анимированного изображения вы в любом графическом редакторе рисуете картинки, из которых будет строиться анимация, а затем загружаете их в специальную программу подготовки анимации, где указываете, как эти картинки должны чередоваться. Таких программ существует большое количество.

Другое незаменимое свойство GIF – это возможность получения прозрачного фона изображения, то есть такого фона, сквозь который видно все, что под ним находится. Создавать GIF с прозрачным фоном может любой профессиональный графический редактор, например, Adobe Photoshop.

### Советы начинающим

- **Большие рисунки** помещайте в формате JPEG, маленькие в GIF.
- Оптимизируйте палитру рисунков. Многие пользователи работают с разрешением 256 цветов. Лучше всего, если палитра **всех** рисунков укладывается в 216 цветов. Это также снижает размер файла.
- Если графический файл получается **довольно большим по размеру**, лучше его просто порезать на куски и выводить как таблицу.
- Если **графическое изображение содержит текст**, ни в коем случае не сохраняйте его как JPG, сохраняйте как GIF.
- В стандартной ситуации **цвет фона страницы** задается тегом

```
<BODY BGCOLOR="#??????">
```

где вместо ?????? подставляется цифровой код цвета в RGB составляющих. Однако, если этот цвет **не входит в универсальную 216-цветовую палитру** (WEB-палитру), на системах с 256 цветами он будет представлен в виде смеси двух или более цветов (размыт). С другой стороны, цвета 216-цветовой палитры далеко не всегда точно соответствуют искомому цвету. Выход из этой ситуации существует. Найдите среди цветов WEB-палитры два наиболее близких к искомому. Создайте из них маленький GIF'чик. Точки выбранных цветов можно расположить в шахматном порядке, или строчками, высотой в 1 пиксел. Используйте этот GIF файл, в качестве фона, тегом

```
<BODY BACKGROUND="backgrnd.gif">.
```

Практически на любом мониторе ваш фон (особенно, если два составляющих цвета не представляют собой контрастное сочетание) будет смотреться, как "сплошной" цвет. По сути, вы сделали дизеринг за браузер, но гарантировали ее корректность и точность.

- Компрессия в **JPEG файлах**. При больших значениях компрессии JPEG имеет тенденцию затемнять фоновый белый цвет картинки, что делает проблематичным ее применение на белом фоне.
- **Нейтральный фон**. Многие помещают в качестве фона логотипы и тому подобную мишуру. Это вызывает усталость глаз, затрудняет чтение (если есть, что читать), раздражает пользователя.
- **Не забывайте про alt="" в картинках**. Не у всех быстрые каналы. Позаботьтесь о тех у кого отключен показ рисунков.
- **Не помещайте** на главную страницу **фоновые звуки** и другие нестандартные м.м. объекты. Это раздражает, когда в качестве приветствия получаешь сообщение об отсутствующем plugin'e.
- **Не злоупотребляйте анимированными GIF'ами** и прочим движением, т.к. это сильно отвлекает пользователя от остального содержимого узла, а при долгом просмотре (например: чтение материала) начинает сильно раздражать.
- **Лучше проставлять размеры у картинок**, которые влияют на раскладку (layout) страницы для сохранения ее первоначального вида при отключенных картинках. Но если Вы используете иллюстрации к

статье, то наоборот, нужно убрать параметры размеров, чтобы большие пустые пятна не мешали читать текст.

- Если Вы используете `BACKGROUND="fone.gif"`, то **неплохо так же задать и `BGCOLOR="#RRGGBB"`**, где `#RRGGBB` - цвет, соответствующий основному цвету изображения `fone.gif`. При загрузке такая страница производит более благоприятное впечатление, так как фоновое изображение подгружается в последнюю очередь.
- **Первое впечатление о вашем узле** складывается от просмотра его заглавной страницы, а если еще учесть, что более 75% пользователей уйдут с нее через 10 секунд, не найдя того, что им нужно, то целесообразно, если активное содержание страницы (информация о том, что конкретно пользователь может найти на этом сайте) будет целиком умещаться в окне браузера при просмотре в разрешении 800x600 (основное количество пользователей пользуются именно этим разрешением либо выше).
- **Создайте прозрачный GIF размером 1x1** (размер обычно 43 байта) и используйте ее для точной подгонки расположения элементов вашего узла или создания разделительных пространств, **выставляя атрибуты `<WIDTH>` и `<HEIGHT>`**.
- Если Вы помещаете страницы узла в таблицы с фиксированной шириной (например 600 пикселей), **не забывайте помещать всю таблицу внутрь тэга `<CENTER>...</CENTER>`**. Этим вы сильно улучшите внешний вид узла при просмотре в больших разрешениях (например 1600x1200).

### Photoshop и web-дизайн

Аннотация: В данной лекции мы поговорим об особенностях графики для Сети, о том, как подготовить изображения для последующего их использования в Интернет. Adobe Photoshop – одна из лучших программ для создания и обработки иллюстраций для Сети.

Программа имеет большие возможности, ориентированные на Web-дизайнеров. Это, например:

- фильтры экспорта в основные графические форматы (GIF, JPEG, PNG) для World Wide Web;

- специальная (безопасная) палитра цветов, обеспечивающая их одинаковую передачу на компьютерах разных платформ, то есть с разными операционными системами, браузерами и техническими (аппаратными) возможностями;
- возможность оптимизации изображений для их размещения в Интернет;
- карты ссылок (image map);
- *ролловеры*; (см. примечание)
- другие возможности.

### **Примечание**

На английском языке интерактивные элементы дизайна web страниц, сочетающие использование графики и скриптов, называются *rollovers* (roll over = перекапываться, переворачиваться). Ролловер представляет собой несколько (до четырех) изображений, связанных с определенными событиями. При возникновении вышеперечисленных событий браузер заменяет одно из этих изображений другим. С каждым событием ассоциируется одно изображение. Обработкой этих событий, их регистрацией и загрузкой изображений занимается сценарий JavaScript. Простейший пример ролловера - кнопка, меняющая свой цвет в зависимости от того, нажата она или отжата.

### **Цветовое кодирование для Интернет. Шестнадцатеричные веб-цвета**

Мы уже знаем, что монитор для отображения цветов на экране компьютера использует цветовую модель RGB. Для Интернет отображаемые браузером цвета кодируются с помощью HTML тегов в шестнадцатеричной системе счисления. В системе координат RGB черный цвет представляется последовательностью 000, 000, 000, а белый - 255, 255, 255. Шестнадцатеричные цвета представляют собой наборы из трех двузначных шестнадцатеричных чисел. Каждое отдельное число соответствует значениям R, G и B, общее количество символов в наборе равно шести. Например, 000000 соответствует черному. Другое крайнее значение - FFFFFFFF, т. е. максимальная интенсивность всех трех цветов соответствует белому цвету.

Итак, в веб-дизайне цвет задается в шестнадцатеричной системе счисления шестью цифрами, каждая от 0 до F. Первые две цифры - красный цвет, следующие две цифры - зеленый цвет, а последние две цифры - синий цвет.

### **Пример 8.1. Перевод RGB цветов в шестнадцатеричную систему счисления (HEX)**

Один и тот же цвет может показываться в разных браузерах по-своему. Это связано со способом отображения графики, когда браузер работает со своей собственной палитрой и не может показать цвет, который у него в палитре отсутствует. В этом случае цвет заменяется сочетанием пикселей других, близких к нему, цветов, имитирующих заданный. Для того, чтобы представляемые цвета были корректно отображены у всех пользователей ПК разработана специальная *Безопасная сетевая палитра цветов* (Websafe Palette), содержащая 216 цветов.

Известно, что **оптимальной палитрой** для GIF файлов, используемых в ВЕБ является т.н. "**non-dither palette**" или "**WEB-palette**", состоящая из 216 цветов. Часто она представляется в виде "цветового куба" со сторонами, состоящими из 6-ти градаций составляющих цветов R, G, B. Эти градации распределяются с шагом в 20%. RGB составляющие принимают одно из 6 значений: 00, 51, 102, 153, 204, 255 -- это и есть 20-процентный сдвиг в рамках 256 градаций.

Использование изображений в этой палитре цветов гарантирует одинаковый показ цветов изображения на любых ПК. Иначе говоря, изображения для показа в Интернет, в идеале (теоретически), должны соответствовать этой палитре. Тогда цвета будут восприниматься другими пользователями точно так же, как видите их вы.

## Новый термин

*Веб-цветами* называются такие цвета, когда для каждой составляющей RGB устанавливается одно из шести значений HEX. Общее количество цветов из всех возможных сочетаний дает палитру веб-цветов. Эта палитра идентична (синоним) палитре безопасных (индексированных) цветов. Пример веб-цвета - #35CC68.

Предположим, что на изображении имеется оттенок зеленого цвета, определяемый в цветовой модели RGB числами 0, 255, 133. Переведем это число в шестнадцатеричную систему счисления, используемую для кодирования цветов в Интернет.

Для пересчета из цветовой модели RGB (DEC) в цветовую систему Index Color (HEX) удобно использовать калькулятор Windows в инженерном режиме. В результате вычислений получим следующие числа:

RGB 0 DEC = 00 HEX,

RGB 255 DEC = FF HEX,

RGB133 DEC = 85 HEX,

где DEC и HEX - обозначение десятичной и шестнадцатеричной систем счисления чисел.

Итак, RGB 0, 255, 133 DEC = HEX 00, FF, 85 = #00FF85.

## Преобразование изображения в режим Indexed Color

Практически для того, чтобы перевести ваше графическое изображение в программе Adobe Photoshop в палитру безопасных цветов (в режим **Indexed Color**), нужно выполнить следующую последовательность действий:

Откройте любое изображение в программе Adobe Photoshop и командой **Image (Изображение) | Mode (Режим)** убедитесь, что изображение находится в режиме RGB.

Выполните команду **Image (Изображение) | Mode (Режим) | Indexed Color (Индексированные цвета)** - [рис. 8.1](#).

### Примечание

При преобразовании многослойного изображения в режим **Indexed Color** все слои будут объединены командой **Flatten Image**.

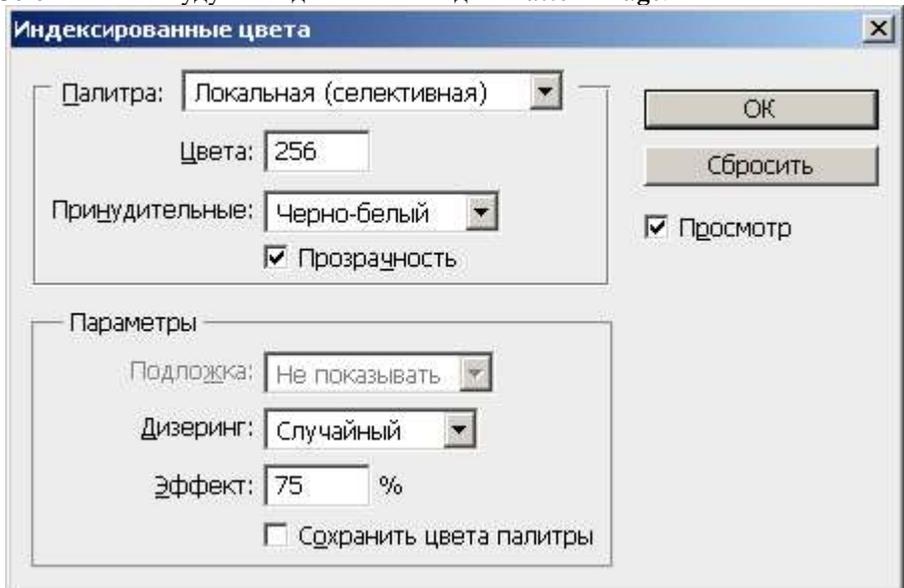


Рис. 8.1. Окно Indexed Color

Далее выберите один из следующих пунктов в списке **Palette** (Палитра):

<b>Exact (Точная)</b>	если изображение содержит 256 или меньше цветов. При этом ни один цвет оригинала не будет удален;
<b>System Mac OS (Системная Mac OS)</b>	если собираетесь экспортировать файл в приложение, которое воспринимает только палитру Макинтош, установленную по умолчанию;
<b>System Windows (Системная Windows)</b>	если собираетесь экспортировать изображение в приложения на платформе Windows;
<b>Web (Сетевая)</b>	если намереваетесь загружать файл в Сеть. Эта опция ограничивает набор цветов изображения тем множеством, которое доступно в большинстве Web-браузеров;
<b>Uniform (Равномерная)</b>	чтобы создать таблицу цветов, равномерно распределенных по спектру. При этом не учитываются преобладающие цвета изображения;
<b>Local (Локальная)</b>	выбирается так, чтобы создать таблицу цветов на основе только тех цветов, которые входят в изображение или в зависимости от того, какие цвета желательно сохранить при преобразовании изображения в режим <b>Indexed Color</b> ;
<b>Master (Специальная)</b>	использует специальную таблицу цветов, созданную заранее на основе группы изображений;
<b>Custom (По выбору пользователя)</b>	чтобы создать собственную палитру. При желании отредактируйте таблицу цветов и щелкните на кнопке <b>OK</b> ;
<b>Previous (Предыдущая)</b>	чтобы использовать палитру, применявшуюся в предыдущем преобразовании, если оно выполнялось.

Из этих вариантов мы остановимся на палитре **Web (Сетевая)**. Тем не менее, когда из миллионов цветов делаешь сотни, то, как видно из иллюстрации ([рис. 8.2](#)) потери качества не избежать. Качество исходного изображения при его индексировании заметно ухудшилось.



**Рис. 8.2.** Исходное изображение (слева) и изображение, после преобразования его цветов в Web Palette (справа)

Несколько улучшить изображение в безопасной палитре цветов можно, если в списке **Dither (Дизеринг)** выбрать подходящий вариант дизеринга, то есть попробовать все способы размытия изображения: **Diffusion (Диффузия)**, **Pattern (Текстура)**, **Noise (Шум)** и выбрать наилучший.

### **Примечание**

Актуальность перевода изображений в безопасную палитру цветов была высока на заре Интернет. Сегодня подобная задача не является первостепенной, так как сегодня вы практически не найдете в Интернет пользователя, который работает на мониторе, показывающем не более 256 цветов.

### **Сжатые форматы графических файлов (GIF, JPEG, PNG)**

Популярный в Сети формат **GIF (Graphics Interchange Format - Формат обмена графическими данными)** - это формат для изображений, содержащих 256 или менее цветов. Он использует алгоритм сжатия LZW (Lempel-Ziv-Welch) с хорошей степенью сжатия. Формат поддерживает прозрачность и чересстрочное отображение. Формат GIF следует использовать для изображений, создаваемых в графических редакторах программным путем (диаграммы, рисунки, текст и т.д.). Формат пригоден для создания анимационных эффектов, так как GIF позволяет хранить в одном файле несколько изображений, а браузеры могут их поочередно демонстрировать. Для каждого изображения можно указать время его демонстрации. Если каждое из изображений, хранящееся в файле GIF, представляет фазу анимации, то браузер покажет маленький "фильм". Эта особенность формата GIF нашла широкое применение в Web-дизайне. Особенно часто анимированные изображения используются в баннерах (от английского banner, плакат). GIF использует 256 цветов и эффективно сжимает сплошные цветные области, при этом сохраняя детали изображения без потерь. Может содержать прозрачные области. Поэтому формат широко используется для создания анимированных рисунков, может содержать текст, логотипы, иллюстрации с четкими краями, изображения с прозрачными участками.

Формат **JPEG (Joint Photographic Experts Group - Объединенная экспертная группа по фотографии)** - это масштабируемый упакованный формат, обеспечивающий высокую степень сжатия с очень небольшим снижением качества изображения. Формат JPEG может быть наилучшим вариантом для точной передачи изображений с плавным переходом цветов, то есть для изображений фотографического типа. Это любое изображение с количеством цветов более 256 (фото со сканера, цифровой фотокамеры, компакт диска и т.д.). Исходя из того, что JPEG объединяет в себе хорошую степень сжатия и незначительную потерю качества, он является идеальным форматом для среды Web-сайта. JPEG использует 16,7 миллионов цветов, поддерживает чересстрочную развертку. Область применения формата в Сети - любое изображение с количеством цветов более 256 - фотографии; рисунки, содержащие прозрачные участки; рисунки с большим количеством цветов и четкими краями изображений.

Оба формата (как GIF, так и JPEG) поддерживают чересстрочную развертку, т.е. загрузка таких форматов осуществляется через несколько строк и на экране по мере загрузки границы изображения становятся более четкими. Преимущество

формата JPEG заключается в том, что он сжимает 24-битное изображение так же эффективно, как формат GIF сжимает 8-битное. Недостатки JPEG состоят в следующем: JPEG - не самый лучший вариант для изображений, содержащих однотонные области или текст, так как подобный метод сжатия вносит заметные искажения в изображения такого типа.

Невозможность создания полноцветных изображений с прозрачными участками в рамках форматов GIF и JPEG привела к появлению формата PNG. Этот формат сохраняет всю цветовую информацию и один альфа-канал изображения. В формате **PNG-8** максимально возможное количество цветов оптимизированного изображения - 256, этот формат похож на GIF. В формате **PNG-24** у изображения могут быть миллионы цветов, этот формат, в свою очередь, похож на JPEG. В форматах PNG используется метод сжатия без потерь данных. Оба формата PNG могут сохранять полупрозрачные области (например, смягченные растушеванные границы), используя метод под названием альфа-прозрачность. В этом случае у точки может быть 256 уровней непрозрачности, начиная с полной прозрачности и заканчивая полной непрозрачностью.

Формат PNG занимает промежуточное положение между GIF и JPEG. Однако в формате PNG-8 нельзя создать анимацию (в формате GIF можно), а файлы в формате PNG-24 больше по размеру аналогичных JPEG-файлов. Формат PNG-8 аналогичен GIF, однако поддерживается не всеми программами. Использует улучшенный формат сжатия данных. Формат PNG-24 аналогичен JPEG, поддерживает 16 миллионов цветов. Подобно формату JPEG, сохраняет яркость и оттенки цветов в фотографиях. Подобно GIF сохраняет детали изображения, как, например, в простых рисунках, логотипах, или иллюстрациях.

После сравнения основных графических форматов для веб-страниц, можно сделать вывод о том, что идеального (универсального) графического формата не существует, поскольку каждый призван решать строго определенный круг задач. Как правило, на веб-странице многоцветные изображения, такие, как фотографии, сохраняются в формате JPEG. Формат PNG-24 также годится для фотографий, однако, в отличие от JPEG, этот формат занимает больше места. PNG-24 рекомендуется для работы с полноцветными изображениями содержащих многоуровневую прозрачность. Иллюстрации с малым количеством цветов, четкими линиями или мелкими деталями, например, содержащие текст, должны быть оптимизированы в формате GIF или PNG-8.

## Оптимизация изображения для Сети в программе Adobe Photoshop

Предположим, что в программе Adobe Photoshop вы командой **Файл | Импортировать** получили цифровое изображение со сканера с разрешением 300 dpi и сохранили его с родным для этой программы расширением \*.PSD. Полученное изображение непригодно для использования в Интернет, поскольку данный графический файл по меркам Сети имеет слишком большой размер за счет избыточного качества изображения. В этом практическом примере мы займемся его оптимизацией, а затем экспортируем в один из графических форматов, поддерживаемых браузерами. Таким образом, мы подготовим сканированное изображение для его применения в Интернет.

Для открытого в программе Adobe Photoshop оригинального изображения выполните команду **File (Файл) | Save For Web (Сохранить для Веб)** и в окне **Save For Web** перейдите на вкладку **Original (Оригинал)**. В нашем случае сканированное оригинальное изображение girl.psd имеет размер 5,49 Мб.

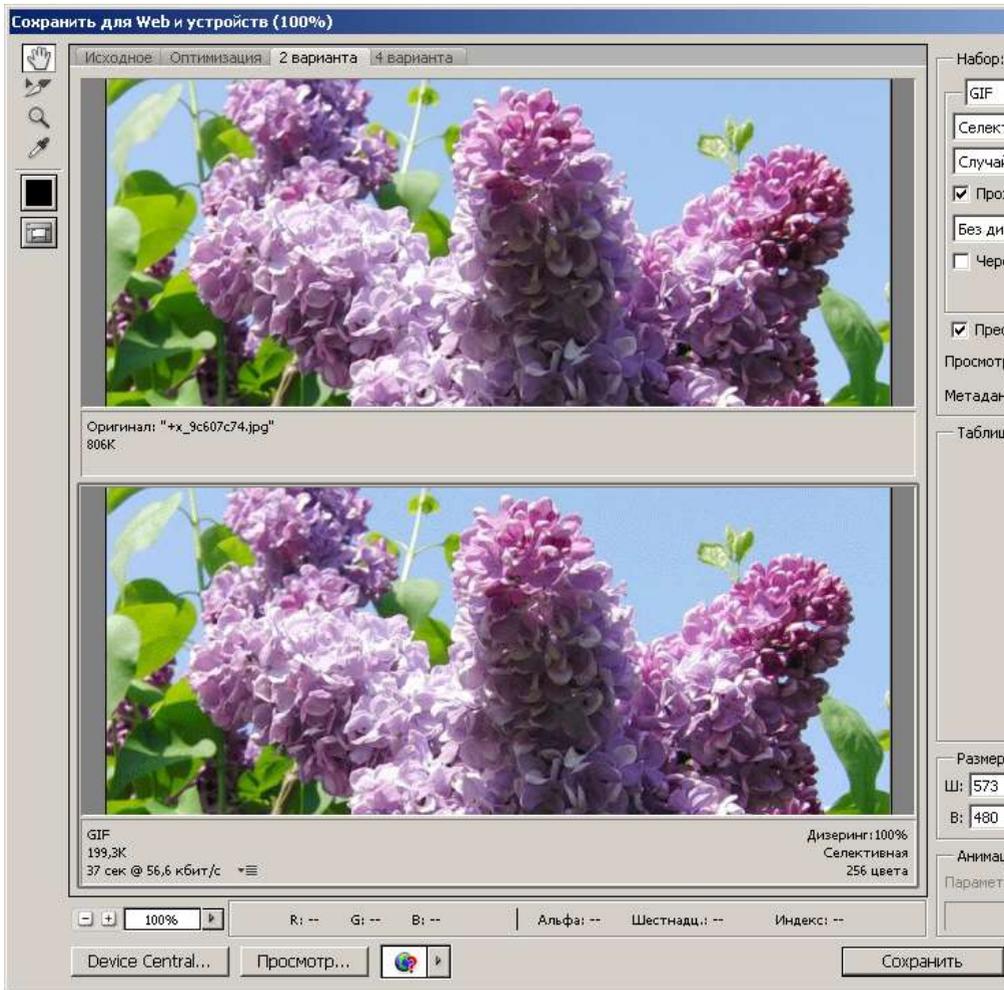
Теперь перейдите на вкладку **Optimized (Оптимизация)**. Перед вами результат оптимизации изображения в соответствии с параметрами, установленными в данном окне по умолчанию. Изменение этих параметров приведет к немедленному обновлению изображения на вкладке **Optimized** в окне документа. Поскольку у нас фотография, то в списке **Optimized file format (Оптимизированный формат файла)** имеет смысл выбрать формат **JPEG**.

Перемещайте ползунок **Quality** справа (высокое качество) налево (низкое качество) до тех пор, пока не заметите появления нежелательных артефактов (искажений) изображения. Мы установили ползунок в положение 30%, что соответствует среднему качеству изображения.

### Совет

Если незначительные артефакты остались, то их можно попробовать удалить размыванием изображения с помощью ползунка **Blur (Размытие)**. Следите за тем, чтобы изображение не слишком потеряло в резкости.

Перейдите на вкладку **2-Up** в окне документа ([рис. 8.3](#)).



**Рис. 8.3.** Окно Save For Web (Сохранить для Веб), вкладка 2-Up

Перед вами одновременно оригинал (вверху) и оптимизированное изображение (внизу). Имея перед глазами результат одновременно двух вариантов оптимизации, вы можете выбрать наиболее подходящий. При сравнении вариантов можно воспользоваться функцией предварительного просмотра изображения в браузере. Для этого следует нажать в нижней правой части окна кнопку **Preview in Default Browser (Просмотр в браузере)** - она открывает список браузеров, установленных на вашем компьютере. При нажатии этой кнопки Adobe Photoshop автоматически создаст временный

оптимизированный файл изображения и необходимый для его демонстрации HTML-код, который и загрузит в браузер.

Взгляните на строку состояния в окне документа. Там отображается текущий размер оптимизированного нами файла (199 Кбайт) и оценка времени его загрузки (37 сек) при скорости линии (56,6 Кбит/сек).

Итак, в результате оптимизации мы получили весьма существенный выигрыш в размере файла, сохранив приемлемое качество изображения.

### Деление изображения на ломтики

Операция разрезания изображения на ломтики состоит в том, что изображение делится на отдельные зоны. На Web-страницах такие изображения, поделенные на ломтики, часто используются для организации меню. Щелчок в разных участках изображения приводит в действие различные гипертекстовые ссылки. Ломтики, создаваемые при помощи инструмента **Slice (Раскройка)** можно создавать, выделять, редактировать и просматривать как в программе ImageReady, так и в программе Photoshop.

### Разрезание изображений на фрагменты

В Photoshop можно делить изображение на фрагменты (ломтики) и назначать каждому такому ломтику свою гиперссылку, то есть можно превращать участки изображения в кнопки. Для практического знакомства с ломтиками проделайте следующее:

- откройте изображение, которое вы собираетесь разделить на ломтики;
- включите режим отображения кусочков нажатием кнопки **Slice Tool (Раскройка)** на палитре инструментов. Даже если вы не трогали этим инструментом изображение, то все равно появляется специальная метка о том, что теперь изображение разрезано на один "кусоч" Об этом говорит окаймляющая его рамка с порядковым номером фрагмента 01 и специальной пиктограммой фрагмента;
- нажмите на левую кнопку мыши и инструментом **Slice (Раскройка)** обведите, как рамкой, левые часы, затем отпустите кнопку мыши. Вы определили фрагмент, который хотели бы иметь в виде отдельного изображения (и файла) - [рис. 8.4](#);



Рис. 8.4. Процесс разрезания изображения на ломтики

### Примечание

Фрагменты, созданные пользователем (пользовательский - 01) и автоматические (02), имеют разные свойства. Программа не может изменить расположение и размер тех фрагментов, которые создал пользователь. За размером и размещением автоматических фрагментов следит сама программа. Если вы измените размер пользовательского фрагмента, то все соседние автоматические фрагменты сами изменятся таким образом, чтобы не возникало "дырок" (перекрывания областей). Автоматические фрагменты имеют в левом верхнем пиктограммы серого цвета, а пользовательские - других цветов, то есть автоматические фрагменты отображаются более приглушенными цветами, чем пользовательские. Границы между автоматическими фрагментами отображаются линией с маркерами, а между пользовательскими - сплошной линией.

Давайте продолжим пример и углубим наши сведения с работой над ломтиками на практике:

- выберите на палитре инструментов инструмент **Slice Select Tool (Выделение фрагмента)**. Он используется для уточнения размеров и положения фрагментов (ломтиков);
- щелкните курсором мыши в пределах созданного вами фрагмента 01. Этот фрагмент будет выделен. Изображение внутри выделенного фрагмента, будь он автоматическим или пользовательским, всегда показывается в оригинальной цветовой гамме. Его рамка приобрела восемь маркеров-манипуляторов, которые используются для масштабирования;
- перетащите правый боковой манипулятор рамки немного вправо. Пользовательский ломтик при этом увеличится по горизонтали, а автоматический ломтик справа станет уже. Верните пользовательскому

ломтику исходный размер. Автоматические ломтики скорректирует сама программа;

- щелкните курсором мыши в пределах правого автоматического ломтика. Изображение внутри ломтика станет ярким, ломтик будет выделен. Обратите внимание, что рамка автоматического ломтика не имеет манипуляторов, поскольку его размер устанавливается программой;
- автоматический ломтик можно преобразовать в пользовательский. Выделите автоматический ломтик 02 и из контекстного меню, открывающегося щелчком курсором с помощью правой кнопки мыши выберите команду **Promote to User Slice (Преобразовать в пользовательский)**. Фрагмент приобретет манипуляторы масштабирования (маркеры), и пиктограммы в его левом верхнем углу перекрасились в синий цвет, а это говорит о том, что автоматический ломтик стал пользовательским.

## Организация гиперссылок

Когда изображение разрезано на требуемое количество фрагментов, их надо ассоциировать с соответствующими документами. Для этого предназначена специальная палитра **Slice Options (Параметры фрагмента)**, вызываемая двойным щелчком кнопки мыши на пользовательском ломтике - [рис. 8.5](#). Содержимое палитры **Slice** относится только к текущему выделенному ломтику и изменяется при выделении другого ломтика.

Параметры фрагмента

Тип фрагмента:

Имя:

URL:

Цель:

Текст сообщения:

Альтернативный текст:

Размеры

X:  Ш:

Y:  В:

Фон:  Фоновый цвет:

OK

Сбросить

**Рис. 8.5.** Вид палитры Slice Options (Параметры фрагмента)

Список **Type (Тип)** задает тип фрагмента. В большинстве случаев это тип **Image (Изображение)**, но возможен и альтернативный вариант **No Image (Без изображения)**, который предполагает отсутствие изображения на месте выбранного фрагмента. Такое "пустое" место (в HTML-коде это пустая ячейка таблицы) может быть заполнено произвольным текстом на фоне со сплошной заливкой. Цвет заливки определяется списком **Фон** в левом нижнем углу палитры. Для типа **Image (Изображение)** цвет, выбранный в списке **Фон**, используется для заполнения прозрачных областей фрагмента.

В поле **Name (Имя)** вводится имя выделенного фрагмента. Поскольку придумывание индивидуальных имен фрагментам в большинстве случаев не имеет смысла, то можно оставить то имя, которое фрагменту назначила сама программа.

Для каждого фрагмента может быть определена единственная гипертекстовая ссылка, которая вводится в поле **URL** палитры **Slice Options (Параметры фрагмента)**.

Список **Target (Цель)**, расположенный ниже, позволяет определить фрейм, в котором появится ресурс, задаваемый ссылкой. Иначе говоря, запись в поле **Target (Цель)** сообщает браузеру, в какой HTML-фрейм загрузить содержимое ссылки и какие HTML-фреймы сохранить.

В нашем случае для выделенного ломтика с именем (номером) 989.02 в поле **URL** мы ввели адрес гиперссылки <http://www.adobe.com>. Это будет имя документа, который откроется в браузере по щелчку на этом фрагменте.

В поле **Message (Текст сообщения)** вводится текстовое сообщение, которое будет отображаться в строке состояния браузера, когда курсор мыши окажется на выбранном фрагменте. Одновременно с этим на экране ПК будет показан текст "Сайт фирмы Adobe", введенный в поле **Alt Tag (Альтернативный текст)**.

Для того, чтобы увидеть результат наших действий в браузере, выполните команду **OK**, затем команду **File (Файл) | Save for web (Сохранить для Web)**, далее откройте сохраненный документ (у нас это файл 989.html) в браузере - [рис. 8.6](#).



Рис. 8.6. Предпросмотр ломтика 2.01 в Internet Explorer

Как видим, программа сама сгенерировала HTML-код и превратила выделенное нами изображение в гиперкнопку.

### Оптимизация ломтиков

*Оптимизация* - процесс выбора такого формата файла, его размера и цветовых параметров, которые позволяют в максимально возможной степени сохранить качество изображения, но при этом обеспечить высокую скорость загрузки его по сети.

Продолжим нашу работу с изображением часов и разрежем данное изображение на четыре ломтика (рис. 8.7), а затем зададим единые параметры оптимизации для всех фрагментов.



Рис. 8.7. Изображение разрезано на четыре фрагмента

Выполните команду **Файл | Сохранить для Web** и перейдите на вкладку **Optimize (Оптимизация)** в окне документа. В палитре **Optimize (Оптимизация)** выберите формат GIF, в списке ниже - палитру **Selective (Выбор)**, режим сглаживания **Diffusion (Диффузия)** и степень размытия (**Dither**) равной 88%, количество цветов равным 128 - [рис. 8.8](#).

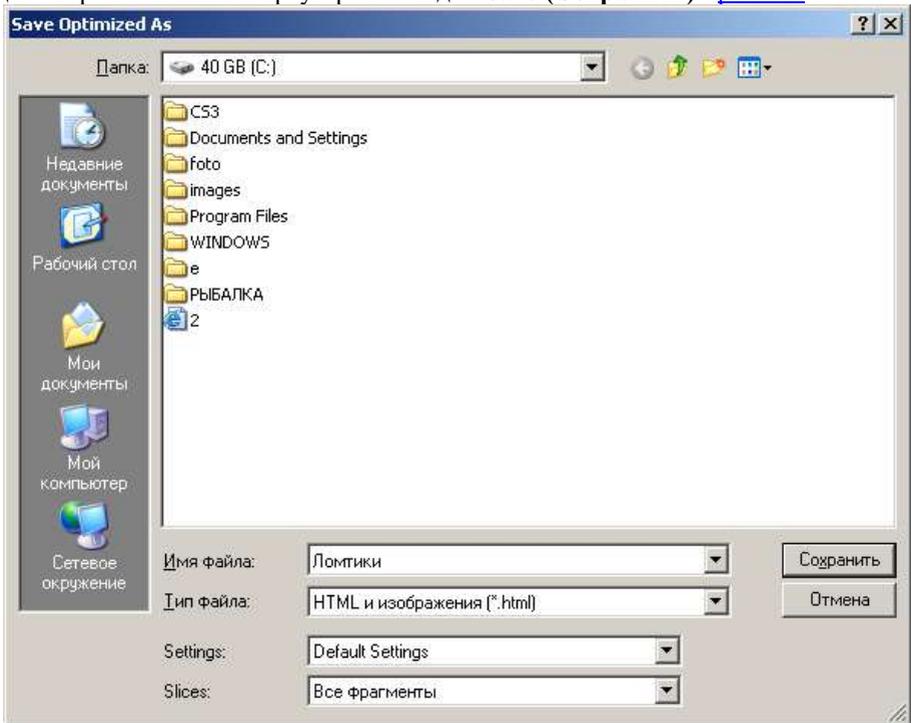


Рис. 8.8. Настройки палитры Optimize

Удерживая нажатой клавишу **Shift**, выделите мышкой все фрагменты изображения (01-04).

Спрячьте границы фрагментов нажатием клавиши **<Q>** на клавиатуре и посмотрите на результат оптимизации. Функция **Preview in Default Browser** позволяет протестировать сборку изображения и гиперссылки на нем еще до экспорта вашей работы в HTML-код. При предварительном просмотре ссылок изображений в браузере границ фрагментов не должно быть видно. Ниже изображения вы увидите HTML-код, сгенерированный программой Adobe Photoshop.

Когда изображение разрезано и ссылки назначены, можно сохранить его фрагменты в отдельные оптимизированные файлы и получить HTML-код для сборки ломтиков в браузере командой **Save (Сохранить)** - [рис. 8.9](#).



**Рис. 8.9.** Окно сохранения результатов нашей работы

В поле **Тип файла** выберите вариант **HTML and Images**, этот вариант включает сохранение кода, необходимого для сборки фрагментов и функционирования ссылок, в HTML-документ и сохранение изображений.

Список **Settings (Установки)** оставьте с установками по умолчанию.

В списке **Slices (Ломтики)** вы можете выбрать для сохранения все ломтики или только выделенные (**Selected Slices**). При этом необходимо предварительно выделить с помощью инструмента **Slice Select** те фрагменты, которые вы хотите сохранить.

Нажмите на кнопку **Сохранить (Save)** - программа сохранит HTML-документ в выбранную папку под именем, которое было введено в поле **File name (Имя файла)** диалогового окна **Save Optimized As**.

### Карта ссылок

Изображения на Web-страницах, созданные описанным выше образом, часто используются как карта ссылок (**image map**). Щелчки курсором мыши в

окне браузера на различных участках такого изображения приводят к загрузке разных адресов страниц (URL). Однако, для создания карт ссылок обычно используют не Photoshop, а Image Ready. Когда карта ссылок определена, функция **Preview in (Предварительный просмотр)** позволяет протестировать карту ссылок еще до ее экспорта HTML-кода и самого изображения на локальный диск. После того, как карта ссылок оттестирована и найдены параметры оптимизации изображения, можно выполнить команду **File (Файл) | Save Optimized As (Сохранить оптимизированное как)**. После выбора параметров сохранения карты ссылок и нажатия кнопки **Сохранить (Save)**, файлы графических фрагментов и HTML-код будут сохранены на диск.

Применение карт-изображений имеет как свои плюсы, так и минусы. Плюсы в том, что карты изображений - удобный инструмент для создания сложных навигационных веб-меню или, например, географических карт. Использование карты в качестве навигационных меню на каждой странице веб-узла сократит время загрузки документов и сэкономит место на сервере, к тому же вы ограничитесь изготовлением всего лишь одного графического изображения. Карта-изображение украсит вашу страницу и придаст вашему сайту оригинальность.

Минусы при создании карт-ссылок таковы. Изображение для карты, имеющее большой размер, увеличивает время загрузки такой страницы. Если параллельно с картами изображений не предусмотрено текстовое меню, то посетители, которые по каким-либо причинам не могут загрузить графику или отключили ее в своем браузере, останутся ни с чем.

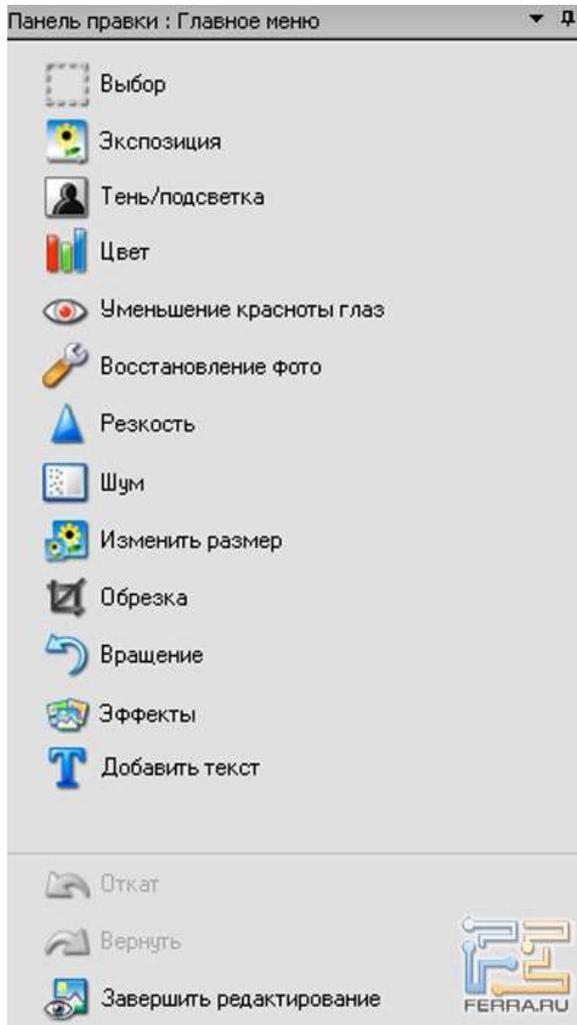
### Уменьшение размеров фотографии без потери деталей

Каждый, кто когда-нибудь сохранял фотографии для Интернета, замечал такую неприятную мелочь: исходная фотография абсолютно резкая, а как только уменьшаем число пикселей раз в десять, вся резкость сразу пропадает.

Но есть способы и алгоритмы, которые позволяют получить нормальную детализацию уменьшенных снимков.

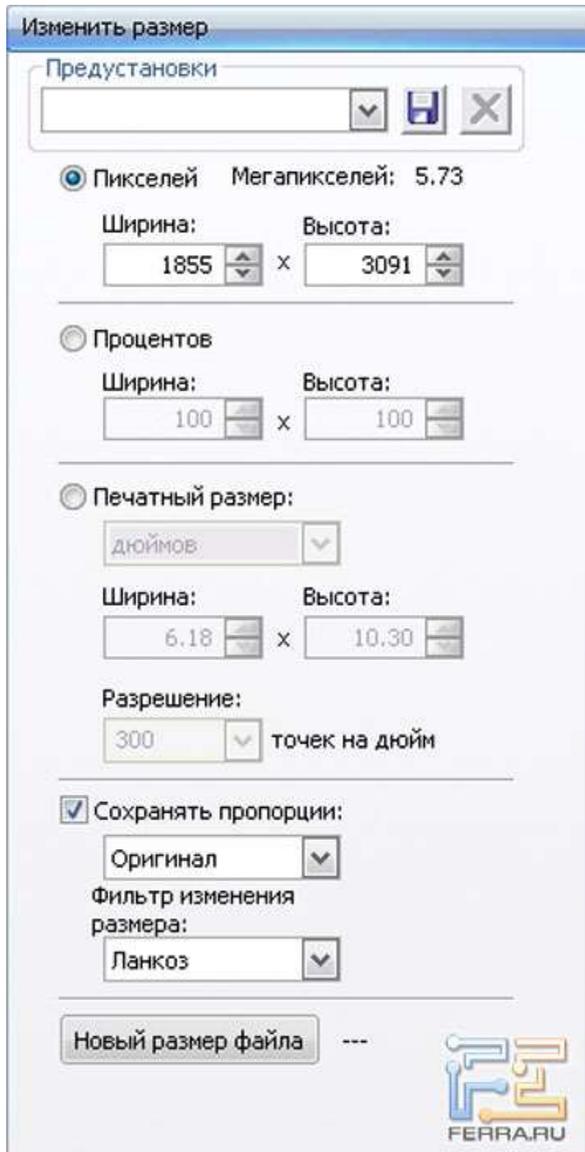
Самый простой и достаточно качественный способ уменьшить размер картинки – воспользоваться распространённой программой для просмотра изображений ACDSec.

Рассмотрим, как можно это сделать в ACDSec Pro 2.5. Запускаем ACDSec и отрываем в программе нужную фотографию. После этого выбираем в меню Изменить-> Режим редактирования).



### Панель правки

Перед нами появляется панель для редактирования изображения. Нажимаем на **Resize** (Изменить размер). Теперь мы меняем размер изображения: можно указать новый размер в пикселях, в процентах от оригинала и в сантиметрах или дюймах.



### Изменение размера

При ресайзе фотографий обязательно убедитесь в том, что выставлен масштаб просмотра 100%, – это позволит контролировать конечный результат. В панели фильтра изменения размера должен быть выставлен алгоритм Lanczos

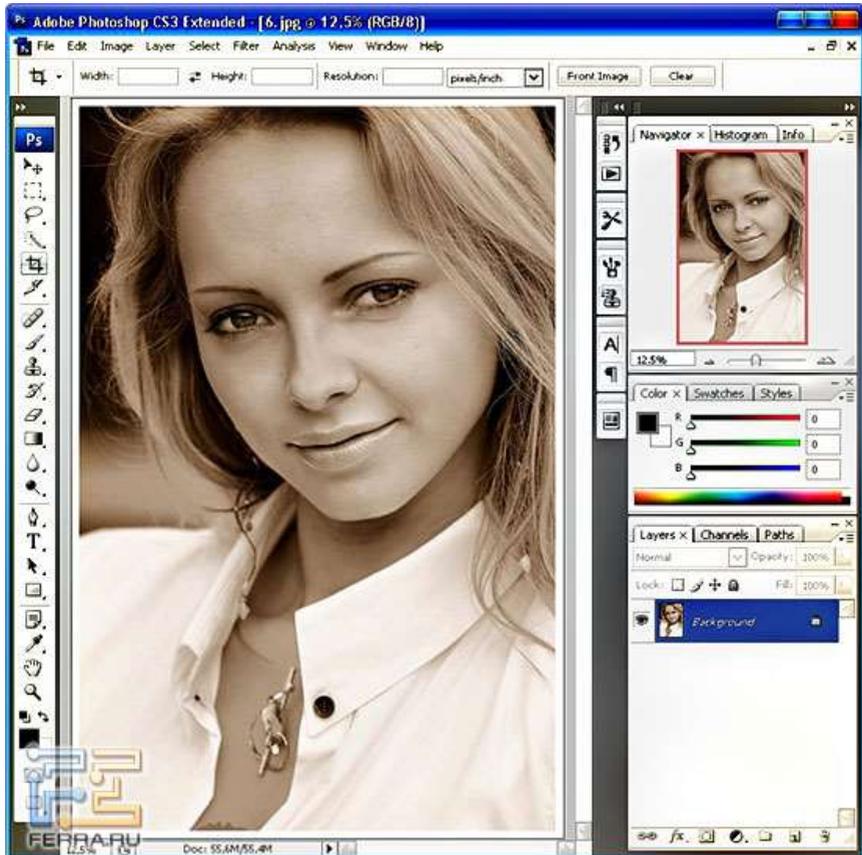
– он показывает наилучшие результаты для большинства фотографий. Конечно, никто не запрещает экспериментировать – можно указать другой алгоритм ресайза, благо в ACDSee их немало.

### Ступенчатый ресайз в Photoshop

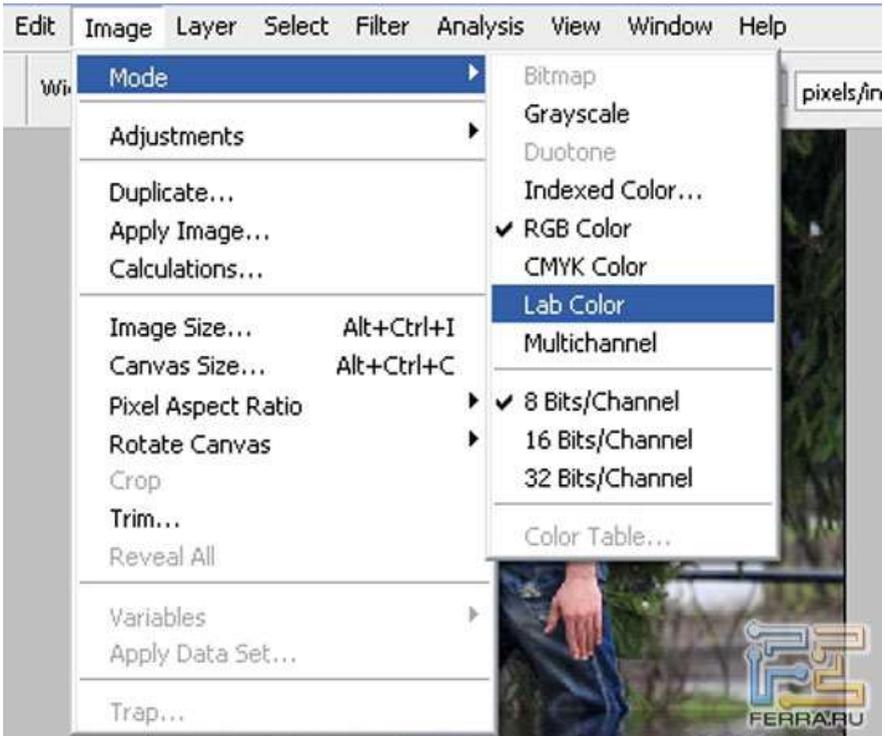
Есть и более сложные методы ресайза фотографий, которые лучше сохраняют детали при изменении размеров изображения. Наиболее распространён последовательный, ступенчатый метод – каждый раз повышать резкость фотографии при уменьшении числа пикселей в четыре раза. Рассмотрим этот алгоритм.

**Шаг №1:** Загрузка изображения и подготовка его к редактированию

1) Запускаем Photoshop и открываем фотографию.



Открываем вкладку Image -> Mode и отмечаем галочкой Lab Color вместо стандартного RGB.



#### Выбор пространства Lab

Благодаря такому преобразованию изображения при повышении резкости не будут возникать цветовые ореолы на контрастных границах деталей.

Дело в том, что при представлении фотографии в режиме RGB за каждый пиксель изображения отвечают три числа: одно обозначает яркость красного, другое – яркость синего, третье – яркость зелёного цвета. То есть у нас есть как бы не одно изображение, а целых три. И увеличиваем резкость мы для каждого канала в отдельности – так уж устроен фильтр Unsharp Mask. При повышении резкости в RGB мы влияем не только на резкость, но и на цвет – ведь при увеличении чёткости фактически изменяется красный, синий и зелёный каналы в отдельности, и от этого возникают цветные ореолы.

А вот Lab Color – другое дело. В цветовом пространстве Lab изображение состоит из трёх каналов – два несут в себе информацию о цвете фотографии, а третий содержит информацию о яркости. И поэтому при повышении резкости только в канале яркости Lightness цветные ореолы не появятся.

2) Делаем активным канал Lightness во вкладке Channels. Остальные каналы необходимо сделать видимыми для просмотра результата повышения резкости.

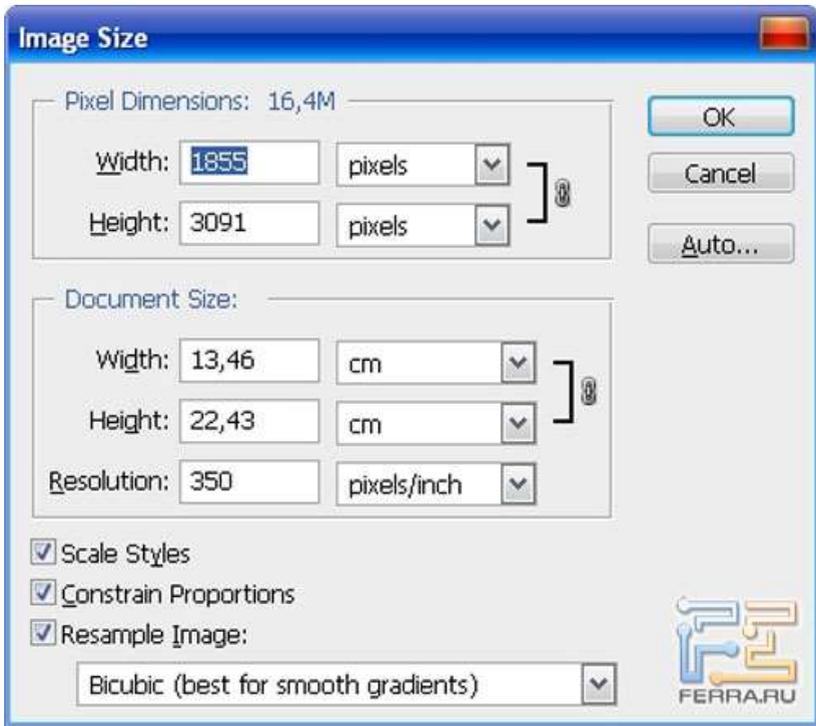


Активируем яркостный канал

3) Перед повышением резкости изменим размер изображения. Для этого открываем окно Image -> Image Size. В поле Width указываем учетверённое значение той ширины изображения (в пикселях), которую хотим получить в конечном счёте.



Изменение размеров изображения



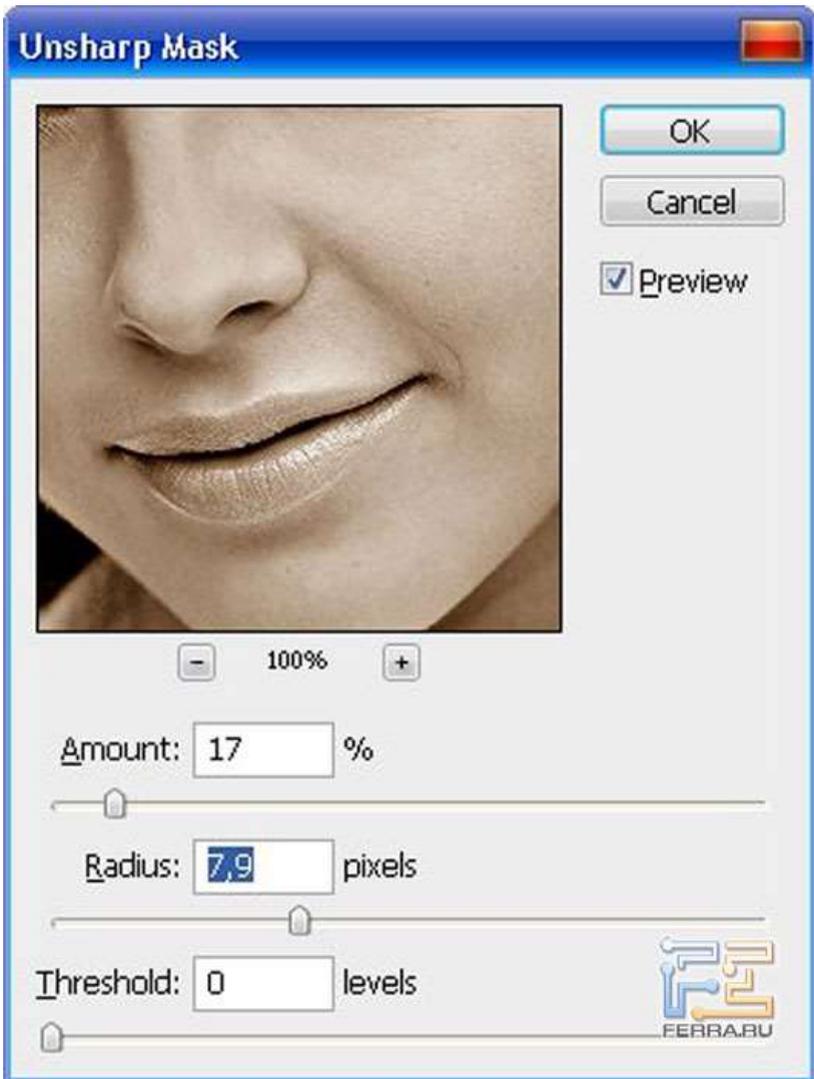
Изменение размеров изображения

## Шаг №2: Повышение резкости и уменьшение изображения

Для эффективного повышения резкости необходимо разобраться с принципом действия фильтра Unsharp Mask. Конечно же, можно было бы сразу записать конкретные настройки для этого фильтра, но это будет не всегда эффективно, а в некоторых случаях даже губительно для фотографии. Просто есть огромное количество факторов, влияющих на выбор тех или иных значений установок фильтра, – для каждой фотографии нужны индивидуальные настройки Unsharp Mask.

Итак, что же значит «резкое изображение»? Резкость – это чёткие границы между деталями. То есть если граница контрастная и чётко выражена, то картинка выглядит резкой. Именно на таком оптическом обмане и основано действие фильтра для повышения резкости Unsharp Mask – этот фильтр увеличивает локальный контраст на границе деталей, что выглядит как повышение резкости. И именно из-за увеличения локального контраста при

злоупотреблении фильтром у предметов возникают неприятные белые окантовки.



Фильтр Unsharp Mask

Рассмотрим настройки фильтра Unsharp Mask. У фильтра есть три ползунка: Amount, Radius и Threshold. Amount влияет на силу воздействия фильтра, Radius – на ширину ореола, Threshold задаёт порог для действия

фильтра Unsharp Mask. Порогу Threshold стоит уделить особое внимание. При нулевом значении для Threshold фильтр Unsharp Mask может подчёркивать шумы изображения и различные нежелательные детали изображения.

Теперь наконец приступим к обработке фотографии. Первым делом мы дублируем слой при помощи комбинации Ctrl+J. Далее делаем активным канал Lightness во вкладке Channels.

### Первое повышение резкости

Приступаем к первому повышению резкости. Для этого выбираем Filter -> Sharpen -> Unsharp Mask. Значение Radius должно быть примерно от 0.8. При выставлении параметра Radius обращаем внимание на сюжетно важные детали (для портрета это будут глаза, для пейзажа – листва и камни) – фильтр должен подчёркивать нужные нам элементы. Если же Unsharp Mask не повышает резкость сюжетно важных деталей, это свидетельствует о необходимости увеличить Radius – такие действия даже позволяют слегка исправить промашку автофокуса, так как в некоторой мере при помощи Radius мы влияем на глубину резкости.

Второй важный параметр – Threshold. Его необходимо подобрать так, чтобы Unsharp Mask не подчёркивал шумы. Обычные значения при повышении резкости для Threshold – 6-8.

И третьим мы выставим Amount, влияющий на силу воздействия фильтра Unsharp Mask. Необходимые значения для Amount зависят от конкретных изображений и колеблются в очень широких пределах – то 70 до 180.

Далее уменьшаем фотографию при помощи Image -> Image size. В поле Width задаём 50 процентов и нажимаем ОК.

### Второе повышение резкости

Выбираем Filter -> Sharpen -> Unsharp Mask. Значение Radius должно быть в два раза меньше по сравнению с первичным повышением резкости. При выставлении параметра Radius снова обращаем внимание на сюжетно важные детали – фильтр их должен подчёркивать. Threshold оставляем неизменным, Amount немного уменьшаем. И опять фотографию делаем в 2 раза меньше при помощи Image -> Image size (в поле Width пишем 50 процентов и нажимаем ОК).

### Финальное повышение резкости

И теперь – последнее повышение резкости: Filter -> Sharpen -> Unsharp Mask, значение Radius – 0.2 или 0.3, Threshold делаем нулевым, Amount увеличиваем до необходимого значения.

Вот и всё. Остаётся только перевести изображение из Lab в RGB (Image -> Mode -> RGB Color) и сохранить уменьшенную фотографию. В результате применения такого ступенчатого ресайза улучшается контроль над шумами и появляется даже некоторый контроль над глубиной резкости по сравнению с однократным уменьшением фотографии до нужных размеров.

### Вопросы для закрепления.

1. Какие виды графических форматов Вы знаете?
2. Какой объем графического файла допустим на web – странице? ».
3. Назовите способы оптимизации графических файлов
4. Чем отличается формат Gif от Jpg?

Домашнее задание: Подготовить графические файлы для сайта и оптимизировать их.

Подготовить рефераты на тему: «Форматы, применяемые для представления графики в сети Интернет», «Методы интеграции изображений в документ», «Приемы позволяющие осуществить фрагментацию изображений», «Принципы создания GIF-анимации», «Программы для оптимизации изображений».

**Тема 9**

**Тема: Этапы создания web-сайта**

**Цели:** *Научиться создавать простые сайты, последовательно проходя все этапы жизненного цикла сайта.*

План урока.

1. Организационный момент.
2. Изложение нового материала.
  - 1) Создание макета сайта в PS
  - 2) Определение и настройка сайта
3. Закрепление изученного.
4. Итог урока.

Ход урока.

**Организационный момент.**

Отметить присутствующих. Сообщить тему урока. Настроить учащихся на восприятие материала.

**Изложение нового материала.**

**Этапы создания сайта**

# Способы создания сайта



## Определение проекта

Выяснение потребностей

подготовка ТЗ

Анализ ЦА и конкурентов

Уточнение целей и планирование

## Разработка структуры

Определение тематики сайта

Создание карты сайта

Разработка навигации по сайту

Создание макета сайта

## Веб дизайн и выверка

Дизайн главной страницы сайта

Утверждение; правки

Дизан внутренних страниц сайта

Дизан интерфейсов и пр. элементов

## Производство и отладка

HTML верстка страниц сайта

DHTML, Flash, JavaScript ...

Подключение и настройка CMS

Тестирование и отладка

## Запуск и поддержка

Наполнение сайта из баз данных

Синхронизация с базами данных

домен, хостинг

Сдача сайта заказчику

Поддержка и продвижение сайта



**1) Определение задач и технического задания (ТЗ) на разработку сайта.**

**2) Разработка дизайн-проекта и его утверждение**

**3) Слияние системы управления сайтом и дизайна.** На этом этапе происходит обрабатывание графического макета web-программистами. После этого сайт обретает функциональные возможности. Весь процесс предполагает проведение нескольких операций:

- верстка;
- система управления;
- интеграция системы управления и результатов верстки.

**Верстка.** Происходит перевод разработок дизайнеров в HTML-код. Профессиональная адаптивная верстка позволит вашему сайту бесперебойно работать во всех web-браузерах и при любых разрешениях.

**Система управления (CMS).** Понятная и функциональная система управления поможет руководить сайтом без посторонней помощи, а также позволит решать задачи статистики, целевого воздействия на сегменты аудитории и видоизменять контент.

**Интеграция системы управления и результатов верстки.** После окончания работ программиста и верстальщика, мы имеем так называемую «рыбу» сайта. Пустой веб-ресурс полностью готовый функционально. Прогнозировать длительность этого этапа невозможно ввиду существенных различий концепций и объема сайтов. Сроки могут составлять от недели до двух месяцев. Еще на этапе верстки и программирования сайта необходимо уделить внимание на разработку функционала для *внутренней оптимизации*, поскольку это очень важный аспект для продвижения сайта в Интернете.

### **Внутренняя оптимизация**

Процесс оптимизации сайта для поисковых систем состоит из следующих пунктов:

1. Наполнение сайта текстами.
2. Установка мета-тегов.
3. Улучшение навигации.
4. Увеличение скорости загрузки сайта.
5. Улучшение юзабилити сайта.

Рассмотрим каждый из пунктов.

**Текстовое наполнение сайта.** Этому пункту уделяется особое внимание, поскольку на исправление допущенных ошибок в будущем может уйти много времени и средств. Главную роль играет объем – поисковая оптимизация сайта считается качественной в том случае, когда численность символов колеблется в пределах от 2 до 8 тысяч. Если количество знаков будет выходить за указанные пределы, то робот может воспринять контент в качестве спама.

Качественный текст соответствует следующим требованиям:

- большая уникальность (проверяется системой Антиплагиат);
- соблюдение правил грамматики;
- соответствие тематики;
- структура;
- наличие ключевых слов

**Заполнение мета-тегов.** Они нужны не для посетителя, а для оптимизации сайта под поисковые системы. Робот сравнивает их с текстом на предмет соответствия, что способствует быстрому продвижению. По словам ленивых мастеров, этот инструмент уже не играет роли. Но их мнение меняется в противоположную сторону после того, когда страницы перестают индексироваться или подниматься в выдаче. Сегодня теги выполняют другие функции, но их значение остается на прежнем уровне. Мета-теги бывают следующими: **title, description, keywords**.

1. Тег <**Title**>. Выводится в заголовке окна, и для каждой страницы уникален. Согласно принципам внутренней оптимизации сайта, в title прописывается только одно ключевое слово, которое стоит в самом начале. Технология не приемлет наличия большого количества ключевых слов, а поисковая система учитывает только первые 62 символа 'тайтла'. Поэтому он должен быть содержательным и передавать смысл текста.

2. Тег <**Description**>. Представляет собой краткое описание содержимого. Оптимизация сайта для поисковых систем невозможна без учёта человеческого фактора: перед кликом по ссылке клиент должен быть осведомлен в том, что он нуждается в этой информации. Description содержит в себе основные ключевые слова. На ведущие позиции выходят те сайты, к статьям которых прилагаются описания. Длина данного тега должна составлять 160 – 200 символов.

3. Тег <Keywords> В этом блоке перечисляются все ключевые слова без повторов, но желательно максимально естественно скомпоновать их. При поисковой оптимизации сайта кейворды имеют не такой вес, как раньше, но забывать о них нельзя.

**Навигация сайта** Предусмотрена для удобного перемещения целевой аудитории по интернет-ресурсу. В этих целях устанавливаются внутренние ссылки, которые также положительно влияют на индексацию страниц. Периодически нужно оценивать эффективность юзабилити путём определения показателя отказов. Чем больше юзеров просмотрели менее двух страниц, тем хуже выполнена оптимизация сайта под поисковые системы. Чтобы процент отказов был как можно ниже, стимулируйте посетителя задержаться на сайте

**Повышение скорости загрузки страниц.** Медленная скорость загрузки сайта негативно сказывается на позициях в выдаче. Если сайт имеет большой вес, то его создатель не сделал код оптимальным. Люди сразу закрывают такие порталы, и время на внутреннюю оптимизацию сайта время потрачено зря. Чтобы избежать печального сценария, руководствуйтесь правилами написания кода и не забывайте о существовании CMS. В результате будете наблюдать продвижение по выдаче и низкие проценты отказов.

**Улучшение юзабилити сайта.** На этой стадии оптимизации сайта для поисковых систем больше всего учитывается человеческий фактор. Эксперты предпринимают действия, заставляющие людей вернуться на сайт для получения новой информации. Чтобы получить хорошее юзабилити, нужно руководствоваться такими принципами:

- мало вводных слов и предложений, полная информативность;
- поставив заголовки, вы сможете не только оптимизировать сайт под поисковые системы, но и привлечь аудиторию, поскольку такой формат более привлекателен, чем «чистый» текст;
- первые слова должны ответить на большую часть вопросов;
- использование маркированных и нумерованных списков;
- наличие параграфов, расстояний между строчками и свободного пространства;
- использованные ссылки отмечаются другим цветом; помещайте медиафайлы (фотографии, видео)

#### **4) Размещение сайта в Интернет**

Приступив к созданию сайта, рекомендуется заранее побеспокоиться о том, чтобы у него было свое имя и адрес, то есть о таких вещах, как домен и хостинг.

**Домен** – имя вашего веб-ресурса. Желательно, чтобы в этом имени находило свое отражение название компании или продукта. Учтите, что необходимо обязательно проверять доменные имена на «занятость».

**Хостинг** – непосредственное место размещения вашего веб-ресурса. То пространство в сети, которое предоставляется вашему сайту.

#### **5) Наполнение сайта.**

#### **6) Поддержка сайта.**

### **Создаем простой сайт на HTML**

Для начала создадим простейший сайт на HTML. Состоит он всего из одной страницы.

Перед Вами текст простейшей HTML-страницы:

```

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Туристическое агентство ТурБеро</title>
<meta name="description" content="Описание страниц" />
<meta name="keywords" content="Киевские сады" />
<link rel="stylesheet" type="text/css" href="style.css" media="screen" />
</head>
<body>
<div id="page">
<header>
<a href="/> <title="> <class="logo"></a>
<div class="phone">+7 (123) 45-67-89</div>

<nav>
<ul id="top_menu">
<li><a href="index.html">О нас</a></li>
<li><a href="services.html">Наши услуги</a></li>
<li><a href="contacts.html">Контакты</a></li>
</ul>
</nav>

<div class="header_img">

</div>
</header>

<section id="content">
<h1>Приветствуем Вас на сайте ТурБеро!</h1>

<p>Здравствуйте! Мы рады приветствовать вас на сайте туристического агентства «ТурБеро»! Мы предлагаем<
<p>Только у нас вы сможете полететь в Индию на слоне и окунуться в глубокое синее море, держась за и<
<p>Если вам нечего заняться, вы можете решить заняться приключениями на свои головы, то приглашаем<
</section>

<footer>
<a href="http://www.internet-technologies.ru/" title="Сайтостроение от А до Я">Сайтостроение от А до Я</>
</footer>
</div>
</body>
</html>

```

Сохраним страницу на компьютере в файле с именем **index.html**. Можно открыть страничку в любом браузере и посмотреть результат. Выглядеть она должна так:

**ТурБюро**  
туристическое агентство

+7 (123) 45-67-89

[О компании](#)   [Наши услуги](#)   [Контакты](#)



**Приветствуем Вас на сайте ТурБюро!**

Здравствуйте! Мы рады приветствовать вас на сайте туристического агентства «ТурБюро»! Мы предлагаем нашим клиентам только самые интересные и захватывающие приключенческие туры по всему миру!

Только у нас вы сможете полететь в Индию на слоне и окунуться в глубокое синее море, держась за плавник акулы. Но не волнуйтесь - каждый клиент нашего агентства застрахован по полной программе и жаждет приключений снова и снова!

Если вам нечего заняться, вы поистине решили заняться приключениями на свою голову, то приглашаем вас в офисы нашей компании по всей стране для выбора самых вкусных предложений.



Сайт создан на Ада.В  
Все права защищены

Ваш первый сайт – готов!

Кстати, `index.html` – не случайно выбранное нами имя. Конечно, не обязательно, но считается хорошим тоном главную (или ту, которая должна отображаться первой) страницу назвать именно так. По умолчанию, любой веб-сервер пытается выдать браузеру страницу `index.html` при обращении к нему (необязательно, т.к. на любом веб-сервере это настраивается).

## Этапы создания сайта

Для того, чтобы создать сайт самому, необходимо выполнить 3 этапа работ:

- Создание шаблона сайта. Именно здесь необходимо определить, как сайт будет выглядеть.
- Верстка сайта. Наполняем сайт контентом и заставляем все это работать.
- Внедрение PHP. Делаем сайт динамичным.

### Создание шаблона сайта

Как правило, этот этап работ выполняется в достаточно мощном графическом редакторе, например, Adobe Photoshop.

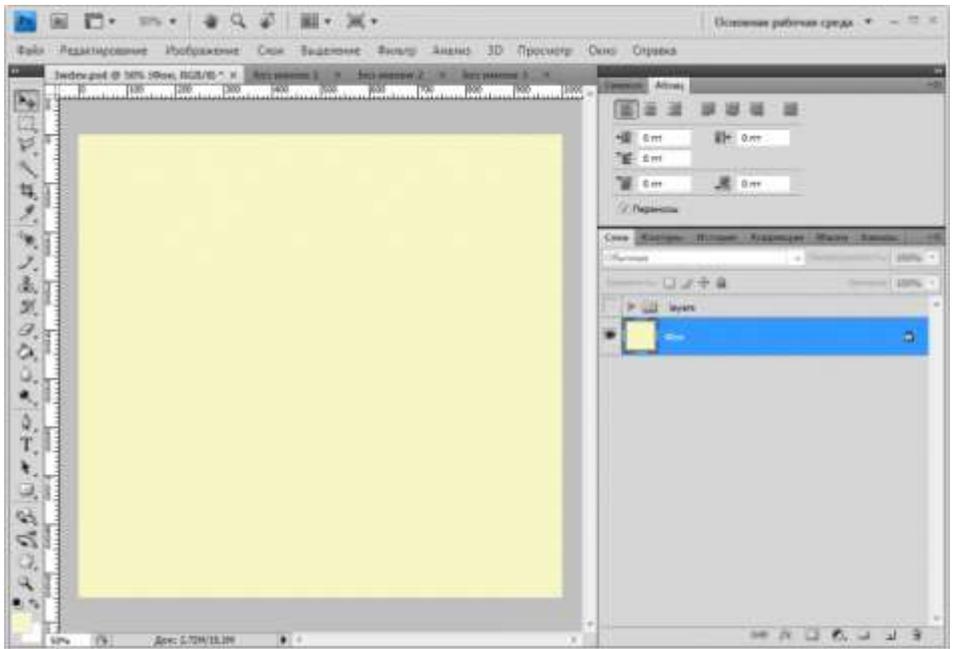
Создаем в Adobe Photoshop новый документ. Задаем имя – *MySite*.

Выбираем разрешение 1000 x 1000, это гарантирует корректное отображение у любого пользователя, размер по вертикали потом сможем увеличить. Выбираем разрешение 72 пикселя на дюйм и RGB-цвет. Эти установки делаем обязательно, именно они отвечают за корректное отображение веб-страницы.

Устанавливаем цвет фона *F7F7C5* в шестнадцатеричном виде или выбираем с помощью палитры цветов.

Выбрав пункт меню «Просмотр» – «Направляющие» активируем отображение линеек и направляющих.

В пункте меню «Просмотр» - «Привязка к» не забываем убедиться, что включена привязка к направляющим и границам документа.

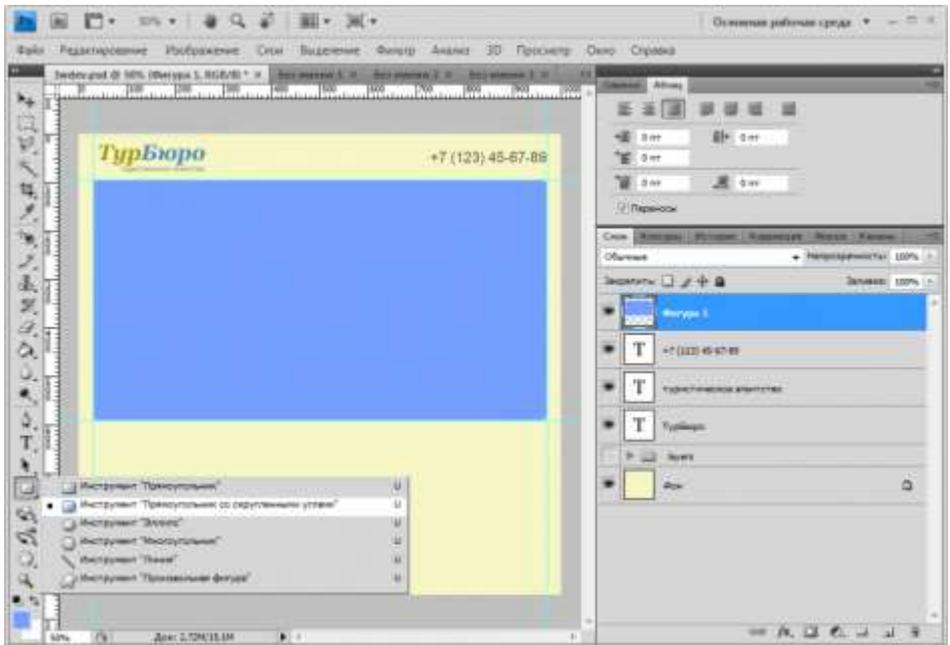


Используя инструмент «Текст», пишем текстовое название будущего сайта, слоган под ним, а также контактный телефон справа вверху макета.



Слева от логотипа, как и справа от контактного телефона, проводим направляющие, которые позволят обозначить рамки по ширине сайта.

Затем с помощью инструмента «Формы» выбираем прямоугольник с закруглёнными краями (радиус – 8 точек), и с его помощью обозначаем место под изображение в шапке сайта:



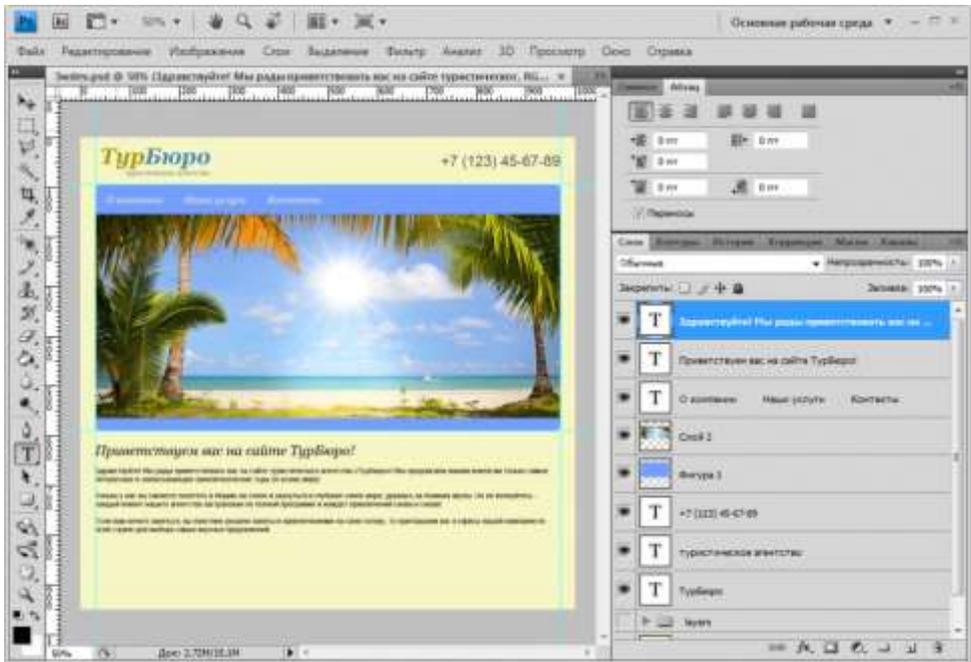
Далее вставляем изображение в шапку сайта. Подходящее изображение вы можете найти на любом из сервисов, предлагающих загрузку изображений.



В помощь инструмента «Текст», и шрифта Georgia, входящего в стандартный набор MS Windows, создаем навигационное меню и заголовок главной страницы сайта.

Также, используя инструмент «Текст» и шрифт «Arial», добавляем текст главной страницы, здесь рекомендуем использовать блочный текст для последующей работы с ним.

Для заголовка и шрифтов в тексте используем черный шрифт. Для навигационного меню – белый.



Перемещая правую границу основного текстового блока, вставляем изображение в тексте страницы (справа от текста).

Используя инструмент «Формы» - «Прямая», проводим заключительную линию под текстом страницы.

Под линией с помощью инструмента «Текст» (шрифт Arial) размещаем копирайт в подвале страницы.

Нарезаем нужные нам фрагменты изображений для вёрстки сайта с помощью инструмента «Раскройка» (мы выделили основную картинку в шапке и картинку в тексте страницы).



В результате проделанных работ получаем эскиз сайта (PSD-файл), который был представлен выше.



Для сохранения результатов работы в виде изображений для вёрстки сайта переходим в меню «Файл» выбираем «Сохранить для Веб». Далее настраиваем качество изображений и сохраняем, в результате получаем множество графических фрагментов нашего шаблона. В папке, куда был сохранен сам шаблон, появилась папка images с изображениями. Отбираем нужные и переименовываем.

Макет создан, необходимые фрагменты получены, можно переходить к верстке. Конечно, пока мы воспользовались лишь самыми примитивными возможностями Adobe Photoshop.

Для быстрого старта зачастую выгоднее воспользоваться готовыми шаблонами, чем рисовать их самим.

### Верстка сайта

Создаем в текстовом редакторе файл index.html.

**ВАЖНО!** Первой строкой файла должна быть строка вида

```
<!DOCTYPE html>
```

Именно она сообщает браузеру, как именно обрабатывать содержимое страницы. Далее следует набор тегов:

```
<html>
<head> «Голова» документа</head>
<body> «Тело» документа</body>
</html>
```

Внутри `<head>...</head>` располагаются теги, которые не отображаются в основном окне. Как правило, они начинаются словом `meta`, и называются мета-тегами, а вот тег `<title>...</title>` отображается в заголовке окна браузера и используется поисковыми системами.

Следом располагается пара `<body>...</body>` в них заключено содержимое страницы. Именно эта часть, тело страницы, отображается в окне браузера.

Существует несколько способов организации контента. Наиболее популярные из них – организация с помощью блоков (`<div>...</div>`) и в виде таблиц (`<table>...</table>`).

Формат отображения элементов может быть задан как напрямую, с использованием соответствующих тегов, так и с помощью таблиц стилей CSS. Второй метод более универсален, так как позволяет использовать стили компонентов повторно во всем документе. Таблица стилей задается либо внутри тега `<head>`, либо в отдельном файле `style.css`, ссылка на который также располагается внутри `<head>`.

На примере нашего сайта, структура его элементов выглядит следующим образом:



Это все касается серверной части, а для программирования на клиентской стороне чаще всего используется JavaScript.

Рассмотрим вариант динамического сайта – папка `php` с основным файлом `index.php`. Он реализует три странички нашего сайта с использованием языка PHP. PHP – Си-образный язык для создания динамических веб-страниц. Главное отличие динамической страницы от статической заключается в том, что она формируется на сервере, а уже готовый результат передается пользователю.

Мы не будем углубляться в программирование на PHP и для наглядности ограничились простыми вставками фрагментов страницы. Суть следующая - мы выносим шапку и подвал сайта в отдельные файлы: `header.php` и `footer.php`, затем в страницах с текстовым содержанием, с помощью PHP, вставляем в макет сайта. Делается это с помощью следующего кода:

```
<?php include('templates/header.php'); // Вставка шапки
сайта ?>
...
<?php include('templates/footer.php'); // Вставка подвала
сайта ?>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Туристическое агентство ТурБюро</title>
    <meta name="description" content="Описание страницы" />
    <meta name="keywords" content="Ключевые слова" />
    <link rel="stylesheet" type="text/css" href="style.css" media="screen" />
  </head>
  <body>
    <div id="page">
      <?php include('templates/header.php'); ?>
      <section id="content">
        <h1>Приветствуем Вас на сайте ТурБюро!</h1>
        
        <p>Здравствуйте! Мы рады приветствовать вас на сайте туристического аген
        <p>Только у нас вы сможете полететь в Индию на слоне и окунуться в глуби
        <p>Если вам ничего заняться, вы поистине решили заняться приключениями и
      </section>
      <?php include('templates/footer.php'); ?>
    </div>
  </body>
</html>
```

```

<header>
  <a href="/" title="" class="logo">
  </a>
  <div class="phone">+7 (123) 45-67-89</div>

  <nav>
    <ul id="top_menu">
      <li><a href="index.php">О компании</a></li>
      <li><a href="services.php">Наши услуги</a></li>
      <li><a href="contacts.php">Контакты</a></li>
    </ul>
  </nav>

  <div class="header_img">
  
  </div>
</header>

```

Попробуйте запустить в браузере `php/index.php`. Не получилось? Конечно, нет. Ведь браузер не знает, что ему делать с командами, из которых состоит файл.

Для того, чтобы файл (скрипт PHP) выполнялся, он должен быть обработан интерпретатором языка. Такой интерпретатор обязательно есть на веб-серверах, который и обеспечивает работу PHP кода.

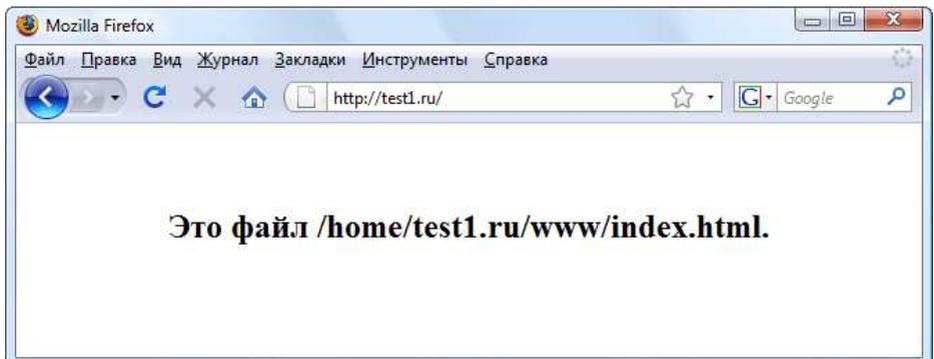
Как же посмотреть, что изменилось? Для отладки веб-приложений и реализации полноценного веб-сервера на машинах под управлением Windows в сети бесплатно распространяется пакет **Denwer**. Денвер (от сокр. Д.н.в.р или ДНВР – джентльменский набор web-разработчика) – набор дистрибутивов и программная оболочка, предназначенные для создания и отладки сайтов на локальном ПК (без необходимости подключения к сети Интернет) под управлением ОС Windows. Является локальным сервером WAMP – акроним от «Windows, Apache, MySQL и PHP. Он включает в себя веб-сервер Apache, интерпретаторы языков PHP и Perl, базу данных MySQL и средства работы с электронной почтой.



LAMP	под Linux
WAMP	под Windows
MAMP	под Mac OS

Установка WAMP не требует никаких усилий. Запускаем установочный файл, выполняем все его требования. Выбираем букву виртуального диска для быстрого доступа, создаем ярлыки. Вот и все. Denwer к работе готов!

Запускается он по ярлыку Denwer (возможно у вас будет называть иначе). После старта, в появившемся в системе виртуальном диске (обычно Z), в папку `home/test1.ru/www/` скопируйте содержимое нашей папки `php`, с которым мы работаем, кроме файла `index.html`. Наберите в адресной строке браузера `test1.ru` (Проверка "Интернет"-домена второго уровня `test1.ru`).



Теперь пройдитесь по остальным ссылкам таблицы тестирования Denwer.

URL	Описание
<a href="https://subdomain.localhost/ssl.php">https://subdomain.localhost/ssl.php</a>	Проверка SSL
<a href="http://subdomain.localhost/">http://subdomain.localhost/</a>	Проверка "не-Интернет" доменов второго уровня, а также SSI
<a href="http://test1.ru/">http://test1.ru/</a>	Проверка "Интернет"-доменов второго уровня: test1.ru (вначале отключите прокси-сервер!)
<a href="http://subdomain.test1.ru/">http://subdomain.test1.ru/</a>	Проверка "Интернет"-доменов третьего уровня
<a href="http://localhost/Tests/phpnotice/index.php">http://localhost/Tests/phpnotice/index.php</a>	Проверка перехвата PHP Notice в Денвере
<a href="http://localhost/Tests/PHP5/index.php5">http://localhost/Tests/PHP5/index.php5</a>	PHP5 information
<a href="http://localhost/Tools/phpMyAdmin">http://localhost/Tools/phpMyAdmin</a>	Проверка MySQL и phpMyAdmin
<a href="http://custom-host:8648">http://custom-host:8648</a>	Проверка хоста с другим IP-адресом и портом (127.0.0.2:8648) <i>В Windows XP SP2 имеется ошибка, из-за которой данный хост может не работать. Официальную "заплатку" от Microsoft качайте <a href="#">здесь</a>.</i>
<a href="http://localhost/Tests/sendmail/index.php">http://localhost/Tests/sendmail/index.php</a>	Проверка отправочной заголовки для sendmail

Работает? Отлично!

Итак мы получили динамический трехстраничный сайт и проверили его!

Но до сих пор мы создавали сайт своими руками от начала и до конца. В настоящее время существуют средства, облегчающие этот процесс – системы управления контентом (CMS – content management system). К наиболее популярным относятся Joomla, Wordpress, Drupal. Для них разработаны сотни шаблонов и дополнений, позволяющих получить профессионально выглядящий сайт с любой функциональностью.

Скорость разработки сайтов, удобство, широкие возможности – всего лишь малая часть преимуществ, которые CMS предоставляют вебмастеру. Особенно приятно, что сами системы и многие дополнения к ним относятся к свободному программному обеспечению, что снимает вопрос о том, как создать свой сайт бесплатно.

Как вариант вы можете использовать онлайн-конструкторы сайтов вроде Wix, A5 и других.

### Публикация сайта в Интернет

Мы готовы опубликовать свой сайт в Интернет. Что для этого нужно?

Во-первых, доменное имя. Доменное имя или домен – символическое имя, под которым Ваш сайт будет известен в сети. Получить его можно либо, купив у регистратора, либо бесплатно, но в виде поддомена от одного из бесплатных хостингов. Более подробно читайте в статье как подобрать домен.

Во-вторых, хостинг. Хостинг – услуга размещения сайта на серверах, имеющих выход в Интернет. В настоящее время компаний, предоставляющих

эту услугу огромное количество. Остается только выбрать подходящую. К критериям выбора относятся:

- Цена – есть как платные, так и бесплатные хостинги.
- Объем предоставляемого трафика и дискового пространства
- Поддержка тех или иных средств (PHP, MySQL, возможность развертывания CMS, наличие дополнительных скриптов и т.д.)

Чаще всего, компании, предоставляющие хостинг, являются и регистраторами доменов, и наоборот. Так что приобрести эти 2 услуги труда не составляет. Оплата для большинства компаний возможна через электронные платежные системы типа Webmoney, Яндекс.Деньги и другие.

### **Размещение сайта на сервере**

Предположим, домен и хостинг мы приобрели. Что дальше? Нам необходимо разместить все файлы нашего сайта на сервере у хостера. Как правило, нам предоставляется несколько вариантов для этого.

Обычно это загрузка содержимого по HTTP-протоколу, с использованием панели управления хостера, или по FTP с помощью FTP-клиента.

**Заливаем сайт на хостинг по FTP.** Второй вариант намного быстрее и проще. Одним из лучших FTP-клиентов является встроенный в Total Commander и бесплатный FileZilla.

Как только устанавливается связь с сервером FTP (хостер обязательно сообщит адрес, логин и пароль входа), пространство сервера станет доступным в виде логического устройства (как и обычные диски вашего компьютера) на одной из двух панелей. Запускаем процесс копирования, ждем его окончания.

Все, наш сайт опубликован в Интернете. Можете набрать в адресной строке браузера его имя и полюбоваться результатом.

### **Конструктор сайтов**

На сегодня наиболее популярным сервисом подобного рода является конструктор **Wix**.

Стоит отметить, что данные сервисы обладает целым рядом явных преимуществ по сравнению со многими другими подобными решениями. По большому счету весь процесс создания сайта с помощью конструктора сводится всего к нескольким простым действиям:

1. Шаг первый – регистрируемся на сайте сервиса. Регистрация абсолютно бесплатна.
2. Шаг второй – входим в свой только что созданный аккаунт и уже в нем выбираем категорию и подкатеорию для создаваемого сайта.

3. Шаг третий – выбираем один из нескольких сотен вариантов графических шаблонов, который позволит придать сайту красочный вид. Стоит отметить, что при желании можно создать свой собственный шаблон, используя технологию HTML5.

После выполнения последнего шага, владельцу только что созданного сайта, остается только наполнить его всеми необходимыми материалами (текстами, картинками, контактами и т.д.). Как видите, выполнив несколько простейших действий, вы создадите свой сайт быстро и бесплатно!

**Тема 10****Тема: SAAS и CMS**

**Цели:** *Научиться создавать проекты в конструкторе сайтов Wix.*

**SAAS и CMS – их различия, плюсы и минусы**

Что такое, по своей сути конструктор сайтов? Если следовать общепризнанной терминологии, то это система, которая позволяет создавать сайты без знания языков программирования и разметки. К конструкторам можно причислить и WordPress. Однако среди конструкторов различают CMS (Content Management System) и Saas-системы (Software as a service). Отличие состоит в том, что Saas предоставляет услуги пакетом: вместе с созданным по готовым шаблонам сайтом вы получаете хостинг и техническую поддержку своего ресурса. Однако, при этом у вас нет возможности вносить изменения в код и менять шаблон изнутри, как это можно делать в CMS. Т.е. по сравнению с CMS, Saas-системы дают меньше пространства для самостоятельной деятельности.

При этом Saas-системы зачастую взимают ежемесячную оплату за обслуживание в зависимости от необходимого функционала. Базовые функции могут предоставляться бесплатно. Но бесплатные версии не дают возможности создавать домены второго уровня. Кроме того, на бесплатном сайте размещается реклама от поставщиков ПО, которую нельзя убрать. В платной версии эти минусы легко устраняются.

SaaS – одна из форм облачных вычислений, модель обслуживания, при которой подписчикам предоставляется готовое прикладное программное обеспечение, полностью обслуживаемое провайдером. Поставщик в этой модели самостоятельно управляет приложением, предоставляя заказчикам доступ к функциям с клиентских устройств.



CMS (система управления контентом) – информационная система или компьютерная программа, используемая для обеспечения и организации процесса по совместному созданию, редактированию и управлению содержимого сайта.

### Преимущества и недостатки SaaS решений

#### Плюсы

1. Высокая скорость развертывания, обусловленная отсутствием затрат времени на развертывание системы.
2. Низкая стоимость поддержки путем сокращения затрат на техническую поддержку.
3. Сокращение финансовых расходов на приобретение лицензии в краткосрочной перспективе.
4. Со стороны разработчика – эффективная борьба с пиратством, т.к. конечная программа не попадает в руки пользователя, а полностью находится на сервере разработчика.

#### Минусы

1. Низкая возможность настройки под Ваши специфические нужды.
2. Наличие постоянных платежей, которые могут вылиться в БОльшую общую стоимость в долгосрочной перспективе.
3. Передача Ваших коммерческих данных стороннему провайдеру, что не всегда безопасно. Стоит отметить, что современные технологии

шифрования позволяют надежно обмениваться коммерческой информацией в сети.

Из вышеперечисленного следует, что пользователи SaaS – это в основном малый и средний бизнес, которому невыгодно вкладывать большие суммы в разработку и поддержку собственных программных продуктов, а проще взять в аренду уже готовый сервис, но в то же время пожертвовать возможностью тонкой модернизации под нужды бизнеса.

### Примеры SaaS

- конструкторы сайтов: Wix, Tilda, uCoz, Shopify. Некоторые имеют базовые бесплатные тарифные планы с ограниченным функционалом и предлагающие дополнительные возможности пропорционально росту тарифа;
- прикладное АО: Adobe Creative Cloud, Microsoft Office 365;
- корпоративная почта на Gmail, Яндекс и др.

### Преимущества и недостатки CMS

Основная задача CMS – предоставление удобного доступа к управлению контентом на сайте для людей не знакомых с программированием, т.к. современные CMS позволяют выполнять большинство повседневных задач в графическом интерфейсе без надобности вносить правки в код системы.

Плюсы:

1. Обширный набор базового функционала.
2. Наличие готовых тем и плагинов для покрытия недостающего.
3. Большое количество специалистов способных дописать отсутствующий.
4. Легкость в освоении управлением сайта.
5. Большинство CMS бесплатны.
6. Высокая скорость разработки в сравнении с решениями без использования CMS.
7. Популярность, которая вытекает в огромные сообщества людей использующих Вашу CMS и если у Вас возникнут трудности, то Вы вероятнее всего сможете найти готовые решения проблем.

Недостатки:

1. Безопасность сайта ниже в сравнении с решениями базирующихся на фреймворках или облачных SaaS решений. Связано это с тем, что исходный код открыт для всех и любой желающий может изучить и найти уязвимости, особенно это касается сторонних плагинов. Но к счастью разработчики популярных CMS довольно оперативно реагируют на подобного рода критические уязвимости и своевременно выпускают заплатки, чего не скажешь о разработчиках сторонних плагинов, некоторые и вовсе могут быть заброшены и не иметь важных обновлений. Поэтому владельцам CMS сайтов необходимо вовремя устанавливать критически важные обновления и заплатки и по возможности включать автоматические обновления, конечно, если CMS позволяет.
2. Большая нагрузка на сервер. Т.к. CMS предоставляет много функций из коробки, некоторые из них могут быть лишними и создавать дополнительную нагрузку на сервер. Дабы её снизить придется покопаться в настройках CMS.
3. Шаблонность базового функционала и дизайна. Для решения этой проблемы существует масса качественных тем и плагинов, которые хоть и не бесплатны, но дешевле разработки подобных оных с нуля. Если же дизайн и функционал не особо сложен, то скорее всего вы обойдетесь хорошими бесплатными темами и плагинами.

### Наиболее популярные бесплатные CMS

- **WordPress**, используется на ~30% всех сайтов в мире и занимает ~60% от рыночной доли среди всех CMS;
- **Joomla**, используется на 3.2% всех сайтов в мире и занимает 6.7% от рыночной доли среди всех CMS;
- **Drupal**, используется на 2.3% всех сайтов в мире и занимает 4.7% от рыночной доли среди всех CMS.

### Конструктор сайта Wix

Конструктор сайтов Wix относится к SaaS-системам. И является одним из лидеров в этой сфере за рубежом. Свою популярность он завоевал благодаря качественному дизайну, адаптивным шаблонам, богатому функционалом при относительно невысоких расценках.



Разработчики Wix ушли гораздо дальше простого создания одностраничников, и сегодня предлагают широкий выбор возможностей: от создания блога до полноценного коммерческого сайта. Рассмотрим подробнее возможности конструктора сайтов Wix.

### **Возможности платформы по созданию сайтов Wix**

Конструктор Wix предоставляет следующие возможности своим пользователям:

- Понятный интуитивный интерфейс;
- Адаптивность под мобильные устройства;
- Бесплатный хостинг;
- Выбор доменного имени;
- App Market – магазин приложений;
- Доступ к фотобанку;
- Видео в качестве фона;
- 3-D эффекты;
- Постоянная тех. поддержка;
- Обновление шаблонов, функционала и многое другое.

App Market позволяет выбирать и устанавливать на сайт множество полезных виджетов: кнопки соц. сетей, чаты, кнопки оплат, комментарии, обратный звонок, Яндекс.Карты, календари и т.д.

Кроме того, Wix дает возможность встроить html-код для расширения функций сайта. Вы можете загружать аудио и видео файлы с возможностью их продажи. Есть в конструкторе Wix и функции необходимые для создания интернет-магазина, инструменты для управления финансами.

Также вы можете пользоваться новыми функциями системы: добавить соавтора сайта, использовать слайд-шоу, анимацию или популярную в среде веб-дизайнеров параллакс-прокрутку, загрузить приложение для фотографов

Wix Pro Gallery, позволяющее обрабатывать фото перед публикацией или загружать свои собственные шрифты.

### **Как создать сайт с помощью Wix?**

1) Чтобы приступить к созданию сайта необходимо зарегистрироваться в системе Wix. Форма очень простая.

2) Далее вам нужно выбрать направленность вашего сайта. Выберем для примера «Блог».

3) Теперь мы можем выбрать любой бесплатный шаблон WIX. Наведя курсор мышки на подходящий шаблон вы увидите следующую информацию: «Цена» - «Редактировать» - «Смотреть», а также «О Шаблоне». Чтобы ознакомиться с шаблоном переходим по ссылке «Смотреть шаблон» и просматриваем его внешний вид и основные функции. Также можем ознакомиться с краткой информацией о нем, кликнув на ссылку «О шаблоне».

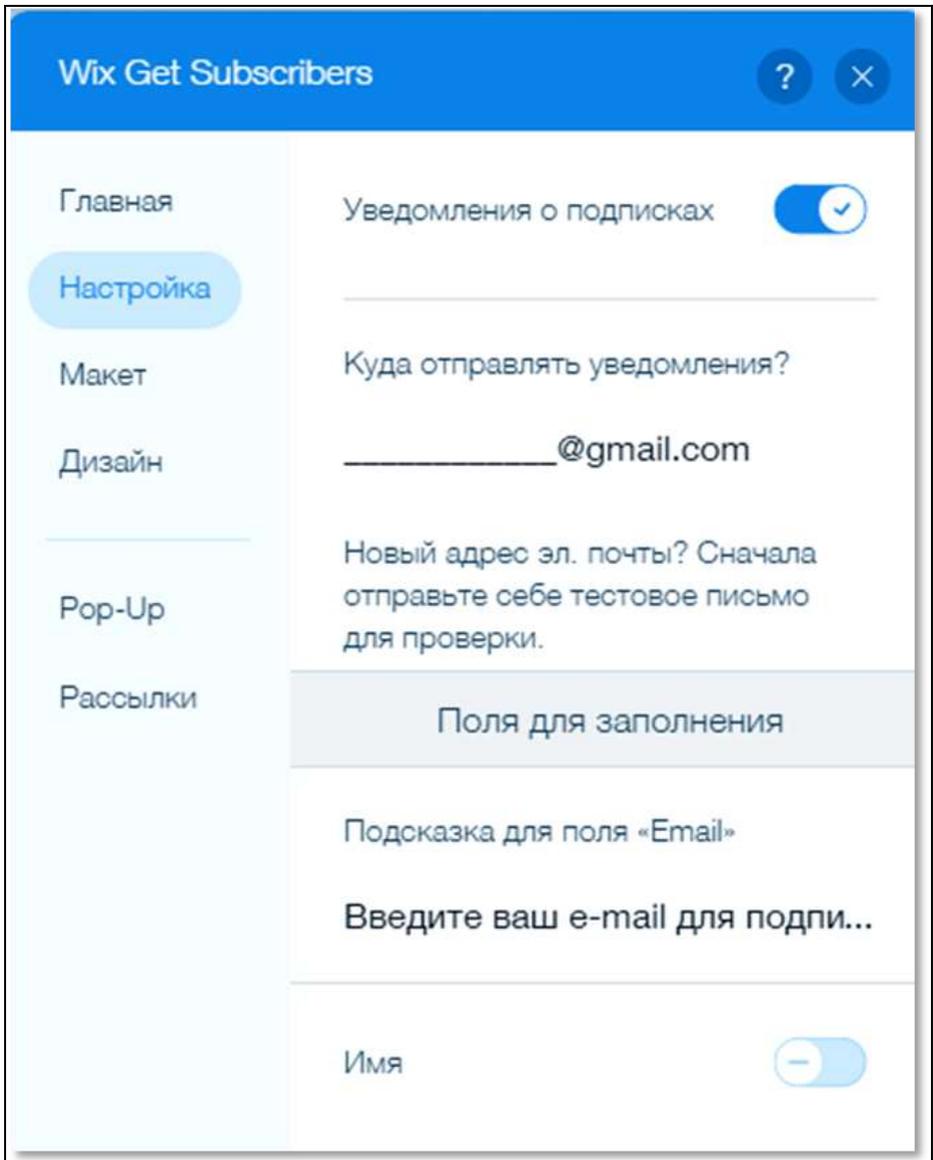
4) Если нас все устроило, кликаем по кнопке «Редактировать», чтобы перейти в редактор Wix. Теперь переходя от блока к блоку, мы можем менять все компоненты блога, по своему вкусу. Кликаем на заголовок сайта и меняем надпись. Мы можем поменять ее размер, шрифт, способ выделения, добавить эффекты.

5) Далее мы можем перейти непосредственно к редактированию постов в нашем Блоге. Для этого нужно кликнуть на один из них и нажать «Редактировать посты».

Там же в редакторе вы можете добавлять новые посты, прикреплять к ним фото, видео, добавлять элементы верстки html. Изображения вы можете загружать свои, либо из библиотеки Wix, а также и соц.сетей (из своих страничек в Facebook, Инстаграмм, Picassa, Google Диск, Flickr, Dropbox) и фотостока BigStock (платно).

6) Теперь перейдем к настройкам виджетов. К примеру, у нас есть строка – подписка на новости сайта. Чтобы ее настроить, кликаем на нее и выбираем «Настройки».

В настройках указываем e-mail, куда нам будут приходиться данные о подписке, а также текст подсказки, который указан в строке, текст кнопки («Go!») и остальные необходимые данные.



7) Теперь разберемся с панелью инструментов. У нас есть шесть кнопок слева, панель инструментов справа и меню сайта сверху:

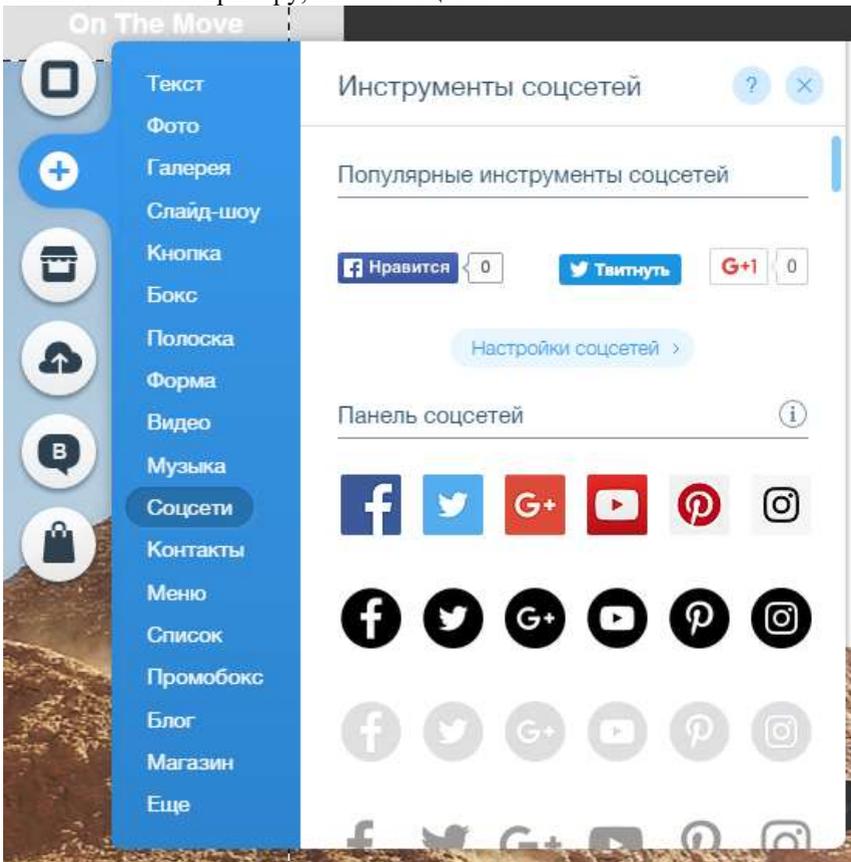
Начнем с шести кнопок слева:

- Фон

- Добавить
- App Market
- Мои загрузки
- Блог
- Создайте интернет-магазин

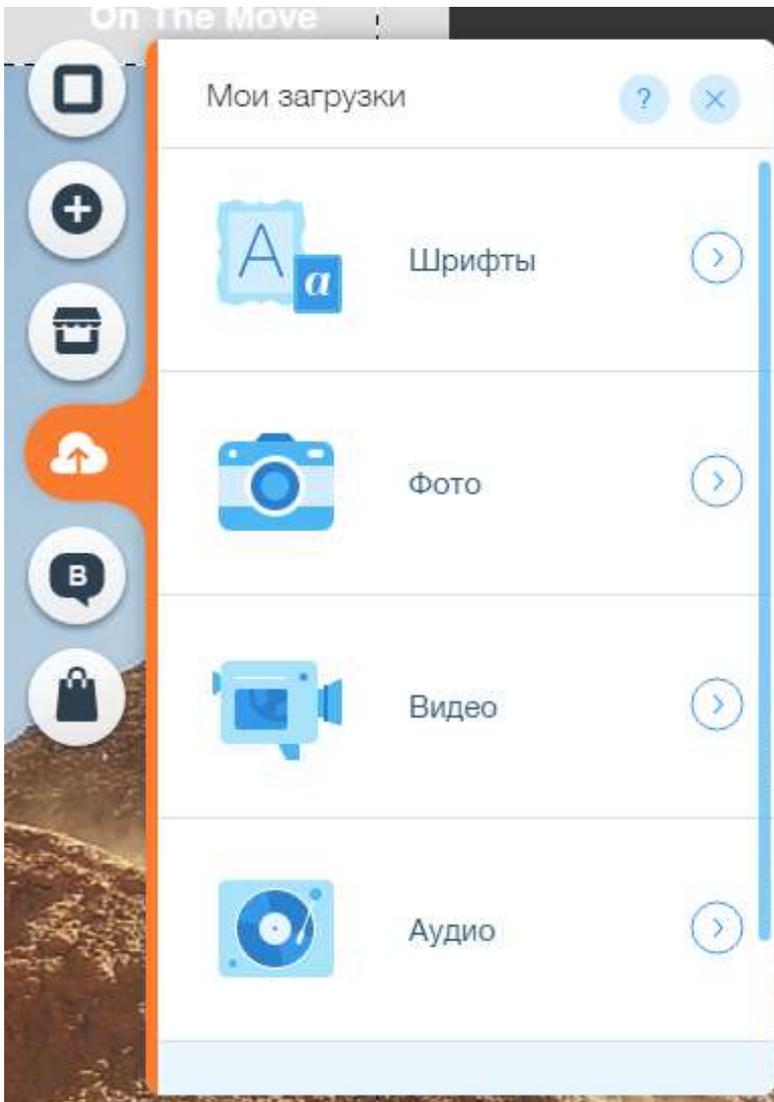
Фон – позволяет добавлять и менять фон страниц, как главной, так и страниц с постами.

Добавить – с помощью этой кнопки можно добавлять дополнительные элементы на сайт. К примеру, кнопки соц. сетей.



App Market – магазин приложений. Вы можете выбирать дополнительные виджеты, которые не нашли перейдя по кнопке «Добавить».

Мои загрузки – загружайте свои фото, видео, музыку и шрифты.



С помощью кнопки «Блог» вы можете перейти в панель управления своим блогом и вносить изменения в отдельные его блоки, а также зайти в центр помощи. Там можно найти ответы по вопросам управления и настройки сайта.

И последняя кнопка «Мой магазин» позволяет на любой странице сайта создать свой интернет-магазин с перечнем товаров.

Теперь рассмотрим правую панель. Ее функциональное значение – управлять расположением и размерами элементов на странице. Так, кликая на каждый блок сайта, вы с помощью правой панели можете изменять его ширину и высоту, угол поворота, выравнивать или перемещать на задний и передний план, копировать элемент, дублировать его или удалить.

Верхнее меню – самое развернутое. Разберем сначала его правую часть.



Лупа – уменьшает масштаб страницы для наглядности.

Стрелками вперед-назад можно быстро отменить или вернуть выполненное действие.

Иконка с телефоном – переход к мобильной версии и обратно. Напомню, что все шаблоны сайтов на Wix адаптивные.

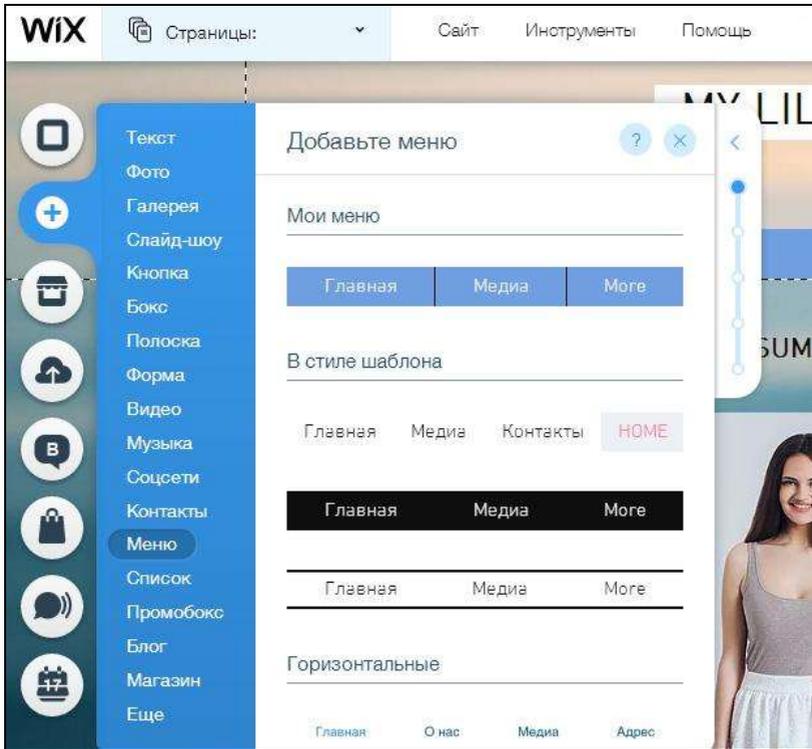
Кнопка с дискетой – одним щелчком сохраняет внесенные на сайте изменения.

Предпросмотр – возможность посмотреть на сделанную работу перед сохранением.

При нажатии кнопки «Опубликовать» – публикуем свой сайт в сети. Однако, если у вас бесплатный набор функций, то доменное имя вашего сайта будет длинным, следующего вида: `https://название_вашей_почты.wixsite.com/mysite-1`. Если же вы хотите нормальное доменное имя второго уровня, то потребуется подключить данную функцию, выбрав подходящий тарифный план.

Теперь рассмотрим левую часть меню.

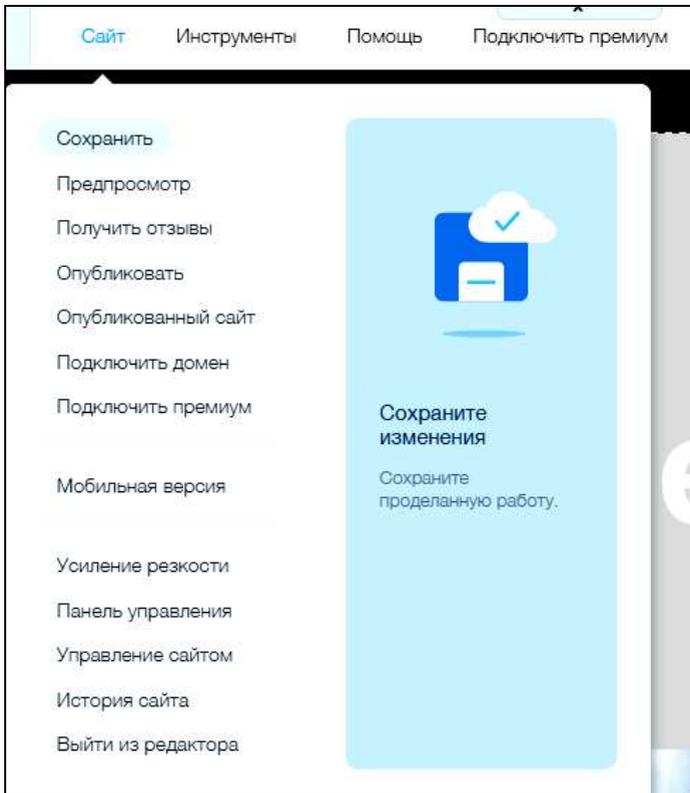
Крайняя левая кнопка «Страницы» позволяет увидеть полную структуру сайта – все его страницы. Возле каждой страницы есть раскрывающееся меню с ее настройками.



В разделе меню «Сайт» есть множество пунктов, которые уже встречались в других разделах панели управления: Сохранить – Предпросмотр – Получить отзывы – Опубликовать – Опубликованный сайт – Подключить домен – Подключить премиум – Мобильная версия.

Новые пункты здесь:

1. Усиление резкости – улучшает четкость фотографий на сайте
2. Панель управление – управление расширенными функциями сайта, такими как Seo-мастер, выставление счетов и прием оплаты, авторассылка, соавторство и другие дополнительные возможности
3. Управление сайтом – включает все основные настройки сайта, в том числе и возможность подключения Веб-аналитики
4. История сайта – позволяет вернуться к любой ранее сохраненной версии вашего блога
5. Выйти из редактора



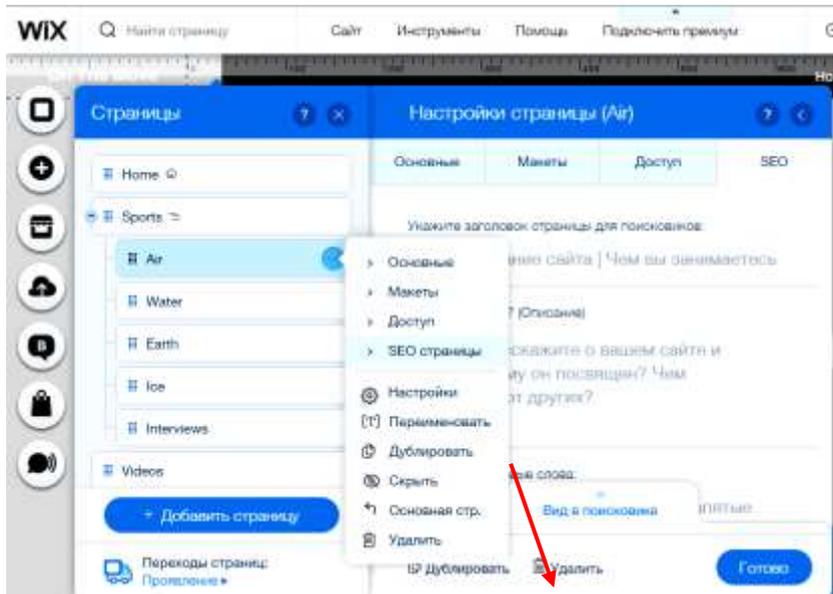
Следующий пункт меню «Инструменты». В нем можно убрать правую панель инструментов, включить/отключить линейку, разметку страницы, а также выравнивание элементов на сайте.

В разделе «Помощь» вы можете найти множество информации по работе со своим сайтом и системой Wix.

Тут вы можете прочесть о том, как подключить домен, набор горячих клавиш в редакторе, условия использования и конфиденциальности, информацию о поисковой оптимизации блога.

Отдельно остановимся на [SEO-оптимизации сайта](#) на Wix. Здесь есть возможность добавлять для каждой страницы сайта мета-теги (title, description, keywords), ч.п.у (человеко-понятный URL).

Для этого в верхней панели выбираем крайний правый пункт «Страницы», возле выбранной страницы ждем кнопку с тремя точками и выбираем «SEO страницы».



Также в редакторе картинок можно добавлять alt-текст изображений. Все эти функции, позволяют осуществлять продвижение сайта на Wix. И, конечно же, не забываем писать [seo-оптимизированные тексты](#) для своего блога. Важным фактором продвижения будет и внятное доменное имя.

## Тема 11

### Тема: CMS WordPress и SEO-оптимизация

#### WordPress

После переноса сайта с локального сервера (Denwer) на реальный хостинг, для SEO-оптимизации необходимо сделать следующее.

1. **Оптимизация контента.** Как уже не раз говорилось, оригинальный и правильно оптимизированный контент – это главный фактор успеха Вашего сайта.

2. **Настройка сервисов обновлений.** В сети существуют специальные службы, следящие за появлением каких либо новых материалов. Называются они сервисами обновлений (Update Services).

Поисковые роботы, в свою очередь, постоянно сканируют эти службы, и узнают таким образом, что и где появилось новенького, и куда можно побыстрее заглянуть.

То есть, если Вы опубликуете новый пост, то благодаря сервисам обновлений, он может быть проиндексирован быстрее. И в дальнейшем, при доработке поста, активность на сайте, не останется без внимания.

В движке WordPress, встроена услуга оповещения этих служб, при добавлении новых статей, и при каких либо изменениях в статьях.

Чтобы использовать эту услугу, нужно зайти в «Консоль – Настройки – Написание», прокрутить страницу вниз до блока «Сервисы обновлений», и ввести в поле для ввода, адреса необходимых служб.



Где их взять? Прямо над полем для ввода адресов, есть ссылка «Службы слежения за обновлениями». Перейдя по ней Вы попадёте в Codex WordPress, на страницу посвящённую этим службам.

Внизу страницы находится список адресов. Прежде чем его скопировать, желательно проверить, все ли они являются действующими.

Для этого каждый адрес нужно вставить в адресную строку браузера, и если откроется сайт службы, или XML код, или сообщение, что служба принимает только XML запросы, то это значит, что адрес рабочий.

Если же выйдет какая либо ошибка, или просто пустой экран, то данный адрес можно считать устаревшим.

Все рабочие адреса нужно вставить в поле для ввода, блока «Сервисы обновлений», каждый с новой строки, и нажать «Сохранить изменения».

К этому списку можно добавить ещё один адрес:  
<http://blogs.yandex.ru/add.xml>

3. Создать карту сайта. Обычно на блоге бывает две карты сайта, одна – та которую создаёте Вы для посетителей, и вторая, в формате XML, для поисковых машин, которую создаёт специальный плагин.

Обе карты имеют большое значение. Первая улучшает поведенческий фактор, так как помогает посетителям ориентироваться на Вашем ресурсе. А вторая, облегчает ту же ориентацию поисковым роботам.

Чтобы создать карту сайта для пользователей, нужно добавить новую страницу (не запись), написать заголовок «Карта сайта» (он появиться в горизонтальном меню), и поместить на эту страницу список с заголовками всех постов, которые имеются на блоге на блоге.

К каждому заголовку нужно прикрепить ссылку на этот пост, чтоб посетитель мог легко перейти на нужную ему страницу.

XML карта сайта создаётся плагином Google xml sitemaps. Как его настроить Вы узнаете прочитав статью [Создание xml карты сайта](#).

4. Создать подписку на рассылку ваших новых статей на почту для ваших пользователей. Существует несколько способов создания подписки. Можно воспользоваться услугами сервиса FeedBurner.

Помимо почтовой рассылки, у него есть услуга оповещения тех же служб слежения за обновлениями, услуга автоматического добавления анонсов или заголовков Ваших статей со ссылкой, практически во все социальные сети, и много чего ещё.

## Языки

... Все, что еще может быть Греческий к нам, но теперь это легче получить FeedBurner говорить на вашем языке. Выберите нужный язык веб-сайт FeedBurner использовать из списка справа.

Выберите один:

- ♦ Английский
- ♦ 日本語
- ♦ Español
- ♦ Português
- ♦ **русский**

5. Устанавливаем на сайт код [Google Analytics](#), и [Яндекс. Метрика](#). А так же регистрируемся в [Google веб-мастер](#), и [Яндекс веб-мастер](#).

6. [Социальные закладки](#). Социальные закладки – это специальные ресурсы в которых любой владелец сайта может разместить анонс на свой ресурс, и добавлять, впоследствии, анонсы новых статей.

Сервисы соц. закладок проводят рейтинги сайтов, и многие предоставляют услуги по аналитике. Эти ресурсы доступны абсолютно всем.

То есть любой пользователь может зайти в соц. закладки, выбрать интересующую его тему, увидеть там Ваш сайт, и зайти на него.

7. Начинайте [общаться на форумах и в соц. сетях](#), и чем больше тем лучше.

## Тема 12

### Тема: Способы размещения сайтов в сети Интернет

**Цели:** *Ознакомить учащихся с основными понятиями и способами размещения и продвижения сайта в сети Интернет.*

План урока.

1. Организационный момент.
2. Изложение нового материала.
  - 1) Основные понятия
  - 2) Домены различных уровней
  - 3) WEB-сервера, назначение, принципы организации
  - 4) Сравнение сайтов регистраторов
3. Закрепление изученного.
4. Итог урока.

Ход урока.

#### Организационный момент.

Отметить присутствующих. Сообщить тему урока. Настроить учащихся на восприятие материала.

#### Изложение нового материала.

##### Базовые понятия и термины

**Web-сервер** – компьютер и программное обеспечение на нем, распределяющее информацию по запросам в Интернете.

**Домен** – часть электронного адреса, состоящего обычно из нескольких полей. Является средством идентификации области ресурсов в Интернете. Домены бывают первого «ru», «com», «net», «org», второго «firma.ru» и третьего «forum.fir-ma.ru» уровня.

**DNS (Domain Name System)** – система адресации доменов, обеспечивающая преобразование имени компьютера в числовой IP-адрес сети Интернет.

**MySQL** – SQL-сервер баз данных. Обеспечивает размещение, хранение и выборку данных через функции доступа для приложений web-сайта. Обладает высокой производительностью, но не обладает большинством присущих для дорогих серверов баз данных возможностей.

**PHP, PERL** – языки программирования, широко используемые для создания интерактивных динамических приложений web-сайтов. Выполняют

большой спектр задач, таких как обработка текстов, работа с базами данных, работа с сетевыми протоколами и т. д.

**Скрипт** – файл сценария (коммуникационная программа), написанный на PERL, PHP C++, ASP и т. п.

**Трафик** – суммарный объем передаваемых входящих и исходящих данных между сервером (почтовым сервером, web-сайтом, FTP-сервером) и пользователем.

**Web-сервер** - это выделенный компьютер с установленным программным обеспечением. Как известно, хорошие серверы стоят дорого, порядка нескольких тысяч, а порой и десятков тысяч долларов. Кроме всего прочего, сервер нуждается в поддержке системным администратором. Такие специалисты являются, как правило, высокооплачиваемыми.

Некоторые компании, у которых есть свой IP-адрес в Интернете, размещают сервер на обычном персональном компьютере, к примеру, используя для этого программу Microsoft Internet Information Server (IIS). В этом случае вы будете платить провайдеру за поддержку **доменного имени** и за трафик. Виртуальный сервер – это услуга, которую сегодня предоставляют многие хостинг-провайдеры. **Сервер называется виртуальным** в том случае, если управление его контентом (содержимым) осуществляется удаленно, а как он реализован физически провайдером, который его предоставляет, не имеет значения. Фактически хостинг-провайдеры сдают в аренду часть дискового пространства серверов своим клиентам. В этом случае провайдер обязуется поддерживать круглосуточную и бесперебойную работу Вашего виртуального сервера. Клиент лишь должен заниматься обновлением своего сайта и вовремя платить арендную плату.

Чтобы практически разместить виртуальный сервер в Интернете и он стал рабочим, минимально необходимо проделать следующее.

1. Первым делом придумайте доменное имя для компании (это имя впоследствии должно стать известным и в Интернете). **Доменное имя**, как правило, имеет такой формат записи: **www.имя\_фирмы.имя\_зоны**. От выбора доменного имени зависит будущее представительства в Сети. Если настоящее (уже существующее) имя компании хорошо известно на рынке, то лучше использовать именно его. Если же по каким-то причинам это имя уже кто-то занял, регистрируйте сокращенное название компании. Замечательно, если имя сайта будет совпадать со словами, известными каждому. Например, **www.anekdot.ru**, **www.rabota.ru**, **www.service\_law.ru** и т. п. Если круг интересов компании относится исключительно к России, то домен следует

регистрировать в зоне ru. Если же к Европе или странам зарубежья, тогда в зоне com, net, org или появившихся за последнее время новых зонах info и biz. Регистрация доменного имени стоит \$15-30 в год. Некоторые крупные компании регистрируют сразу несколько доменных имен, например «firma.ru» и «firma.com». А многие компании одновременно с регистрацией домена обеспечивают и хостинг для Вашего сайта.

2. После выбора домена надо **выбрать хостинг-провайдера**, у которого вы будете физически размещать свой виртуальный сервер и регистрировать доменное имя. В настоящее время таких компаний очень много, и они предоставляют далеко не одинаковые возможности своим клиентам.

Существует как платный хостинг, так и бесплатный. Следует отметить, что бесплатный хостинг целесообразно использовать исключительно если:

- Вы собираетесь сделать свою персональную страничку в Интернете с краткой информацией о себе, своих увлечениях и т. д.
- Ваш сайт не носит коммерческий характер.

### **Домены второго и третьего уровня**

Конечно, если сайт вашей фирмы все же является зарегистрированным у провайдера бесплатного хостинга, то ничего криминального в этом тоже нет. Многие сотни фирм имеют именно такие сайты. Однако сам факт бесплатного размещения вашего сайта может негативно повлиять на имидж вашей компании, так как большинство солидных фирм являются обладателями **доменов второго уровня**. Если же сайт расположен на бесплатной хостинг-площадке, то его адрес в Интернете будет иметь примерно следующий вид: **http://www.xxx.yyy.ru**, где xxx – это название вашей компании, а yyy – это имя хостинг-провайдера, который и предоставляет бесплатный хостинг. Представьте себе, что хостинг-компания «УУУ» обанкротилась или закрылась. Соответственно, ваш сайт тоже закрывается, так как название представительства вашей фирмы (которое называется в этом случае **доменом третьего уровня**) привязано к имени хостинг-провайдера (домену второго уровня). А домен всегда зависит от домена предыдущего уровня. Бесплатным бывает только сыр в мышеловке! Бесплатный хостинг окупается за счет размещения рекламы на сайтах, которую вставляют на них провайдеры в обмен на бесплатное предоставление своих услуг. Постоянная посторонняя реклама на коммерческом сайте производит негативное впечатление на посетителей, в то время как на платном хостинге посторонняя реклама отсутствует, если, конечно, вы не опубликуете ее сами. Кроме того, компании, которые предоставляют

бесплатный хостинг, как правило, не несут ответственности за различные сбои на сервере и потерю информации.

Рассмотрим теперь ситуацию, когда адрес вашего сайта имеет доменное имя второго уровня, например **http://www.firma.ru**. В этом случае благополучие сайта не зависит от благополучия компании, предоставляющей платный хостинг, так как домен второго уровня фирмы зарегистрирован в DNS Интернета. Провайдер занимается лишь его поддержкой. В случае закрытия этой хостинг-компании адрес вашей фирмы в Сети останется прежним, нужно будет лишь отдать его на обслуживание другому хостинг-провайдеру. Это еще одна причина, по которой коммерческим проектам рекомендуется регистрировать домены второго уровня и использовать платный хостинг. Ведь в случае какой-нибудь неполадки на сервере провайдера вы всегда сможете отстоять свои права или даже потребовать компенсации, если хостинг вас не устраивает.

#### **Хостинг-площадка**

Чтобы выбрать хостинг-площадку, сделайте сравнительный анализ нескольких провайдеров, которых вы можете без труда найти в любой поисковой системе. Прочитайте, что пишет хостинг-провайдер сам про себя, то есть его описание на странице поиска.

**3.** Определитесь, каким по объему будет ваш сайт, вернее, сколько информации по объему он будет в себе нести. Для маленьких сайтов (краткая история фирмы с контактными данными и перечень услуг, оказываемых ею) вполне хватит даже 1 Мбайта. Если же вы собираетесь размещать фотографии, тогда выбирайте объем 10-20 Мбайт. В том случае, если же сайт представляет собой крупную корпорацию и будет содержать большое количество информации, подразделы, фотографии, прайс-листы и архивы, тогда целесообразнее выбрать сразу 100 Мбайт. Для реализации крупных проектов, например, интернет-магазина с большим количеством товаров, поисковой системы с базой данных и т. п., вам может потребоваться 500 Мбайт, а то и больше. Кстати, в этом случае виртуального сервера может оказаться недостаточно, так как, с точки зрения разработчика большого проекта, рядовые провайдеры предоставляют достаточно малый набор услуг. Будем полагать, что с размером сайта мы определились.

**4.** Теперь нужно выбрать услуги, которые будут необходимы для функционирования вашего интернет-проекта.

Большинство хостинг-компаний предоставляют следующие программные возможности:

- запуск своих скриптов на языках PHP, PERL;
- содержание базы данных MySQL;
- гостевые книги;
- др. сервис.

### **Трафик**

Вопрос предоставляемого хостинг-компанией трафика может решаться по-разному (он может быть ограниченным определенным значением либо вообще неограниченным). Первым делом лучше обратить внимание именно на ограничения трафика (или на отсутствие такового). Ведь если трафик, который будут создавать посетители вашего сайта, не придется оплачивать дополнительно, это выгодно и удобно!

### **Возможность использования PHP, PERL, CGI, SSI, MySQL, ASP, C++**

Кроме трафика, надо обратить внимание на возможность использования PHP, PERL, CGI, SSI, MySQL, ASP, C++ и других программных средств. Впрочем, если проект предполагается сделать достаточно примитивным, простым и не интерактивным, тогда они попросту не нужны. Напротив, если вы собираетесь устраивать различные голосования, организовать конференции или чат, то размещение скриптов и сценариев просто необходимо. Если же вы уже имеете хостинг у хостинг-компания, а по предоставленному (ранее выбранному вами) плану вас не устраивает перечень возможностей для функционирования сайта или требуются др. возможности, - смените план. Переход на другой план у той же хостинг-компания осуществляется без каких либо проблем. Все сведется к дополнительной оплате услуг на выбранный Вами тарифный план и продолжительность оплаты по нему хостинга.

Самые популярные сегодня программные средства – это PHP и PERL, чуть менее MySQL. Если провайдер не предоставляет возможности использовать ASP, CronTab и C++, в этом нет ничего страшного. Для решения ваших задач они могут просто не понадобиться. Предоставляемых хостинг-компанией услуг уже готовых программных средств может оказаться достаточным.

### **Домены третьего уровня для вашего сайта**

Если провайдер предоставляет возможность зарегистрировать домены третьего уровня для вашего сайта, то это несомненный плюс. В этом случае вы сможете создать на своем домене еще домены вида **http://forum.fir-ma.ru**, **http://mailfirma.ru** и т. п. Не последнюю роль играет служба поддержки клиентов. Если компания отвечает на вопросы четко и доступно, то это большой

плюс. Будет замечательно, если при покупке хостинга на год вам бесплатно зарегистрируют домен второго уровня в какой-нибудь другой зоне, отличной от той, в которой вы зарегистрированы. А еще лучше, если в стоимость услуг хостинга будет включена бесплатная первоначальная раскрутка проекта. Ведь всем известно, что, каким бы красивым не был сайт, о нем должны знать и его должны посещать.

Помните, что многие хостинг-компании стремятся к тому, чтобы их сервис запомнился своими особенностями и отличался от других, вводя различные бонусы и подарки. Этим стоит пользоваться.

Подведем краткие итоги и обозначим основные свойства, которыми должен обладать хороший хостинг:

- **Надежность.** Сервер должен бесперебойно работать. Вы должны быть уверены, что в один прекрасный момент вас не «отрубят» и все ваши старания не пропадут.

- **Качество доступа.** Сервер должен быстро отвечать на запросы пользователей. Никто не будет ждать полчаса загрузки страниц вашего сайта.

- **Сервис.** Для создания полноценного сайта вам не обойтись без закачки файлов по FTP-протоколу, поддержки PHP, PERL и MySQL для организации форумов, гостевых книг, голосований, баз данных, магазинов.

- **Раскрутка.** Недостаточно просто разместить сайт в Интернете. Необходимо, чтобы его посещали пользователи. Несмотря на то, что эта задача относится к инетрнет-агентствам, все же хорошая хостинг-компания должна позаботиться о первоначальной раскрутке сайта клиента. Эти мероприятия включают целый комплекс вопросов, которые частично раскрыты в следующем вопросе лекции.

Это далеко не все аргументы, по которым необходимо судить о качестве хостинга. Бывает, что с виду хостинг хорош, а на самом деле либо каналы связи работают медленно, либо периодически без ведомой причины сервер отключается. Так что готовых рецептов на все случаи жизни просто не существует и многое следует просто опробовать, чтобы получить опыт в данном вопросе.

### **WEB-сервера, назначение, принципы организации**

*WEB-сервера* - это сервера в сети Интернет, предоставляющие гипертекст, размеченный с помощью языка HTML.

Программа для просмотра и получения документов с WEB-сервера называется браузером. Обычно браузер получает информацию с WEB-сервера по HTTP-протоколу. Скорость передачи информации, которую показывают

многие браузеры - это средняя скорость получения информации с учетом всех задержек (число всех полученных байт, деленное на затраченное на их получение время).

Прежде чем перейти к изучению протокола HTTP, систематизируем сведения по уже изученным протоколам:

1. IP-протокол предназначен для доставки пакетов от одной машины в сети к другой. При этом совершенно неважно, какие это данные и как они будут в дальнейшем использоваться. Вообще говоря, этот протокол даже не гарантирует доставку пакета.

2. TCP-протокол предназначен для установки двунаправленного соединения между двумя машинами. Одна машина - клиент - запрашивает соединение с другой машиной - сервером, и протокол TCP используется для организации канала, по которому эти две машины могут обмениваться данными. В качестве нижележащего протокола используется протокол IP. Упаковка данных в IP-пакеты контролируется TCP. Он пытается передать пакет до тех пор, пока не получит сообщение, что пакет доставлен и получен без искажений.

3. Над этими двумя протоколами находятся протоколы доставки почтовых сообщений, самый распространенный из которых - SMTP (Simple Mail Transfer Protocol) - отвечает за доставку почтовых сообщений между серверами. Он передает следующую информацию: от кого письмо, кому письмо и тело письма, установку соединения и корректность передачи контролирует TCP-протокол, а выбор с кем соединиться на совести почтовой программы (задается в ее настройках или берется из DNS). Протокол POP3 обеспечивает возможность пользователю получать почтовые сообщения из ящика, находящегося на сервере.

А теперь обсудим HTTP-протокол. Этот протокол также работает над TCP. Одной из его задач является регламентация передачи данных на WEB-страницах. Он используется при передаче любых данных в соответствующем формате и содержит средства для указания, какие данные он передает в настоящий момент и как с ними следует поступить.

Рассмотрим с точки зрения сети этапы просмотра WEB-страницы

1. Клиентская программа просмотра WEB-страниц (браузер) устанавливает соединение с сервером (TCP).

2. Получив ответ об установленном соединении, программа направляет запрос серверу. В запросе указывается, что ей нужен такой-то

ресурс (файл, динамическая страница или каталог), желательно в такой-то кодировке и на таком-то языке.

3. Сервер направляет ответ из 3 частей:

- Код ответа (все в порядке, или документ в другом месте, или этот документ не имеют права читать все подряд, сообщите свое имя и пароль, или пароль неверный, или нет таких документов, или сервер не работает)

- Заголовки страницы (тип данных (текст+кодировка, звук, рисунок и т.д.), иногда размер, можно ли сохранять этот документ в кэше, срок хранения и т.п.)

- сам запрошенный документ.

Браузер, получив этот ответ, сохраняет его у себя либо в оперативной памяти, либо на диске (в дисковом кэше) и, исходя из заголовка документа, решает, что с ним делать... Если выводился текст в формате HTML, то там могут быть ссылки на другие ресурсы. Для каждого ресурса, например, картинки, устанавливается новое соединение с сервером, и она точно так же запрашивается.

### **Файловые сервера. Пересылка и получение файлов.**

FTP-протокол регламентирует работу с файлами на FTP-серверах, например такие операции, как копирование файлов, удаление файлов, перемещение, получение файлов с сервера на локальную станцию, или наоборот - передачу локальных файлов на сервер, просмотр каталогов и т.д.. Мы будем рассматривать только просмотр каталогов и получение файлов с сервера. FTP-протокол передает любые данные в одном из двух форматов - текстовом или двоичном.

Этапы получения файла по FTP-протоколу:

1. Установка контрольного (управляющего, для команд) соединения с сервером; выполняет FTP-клиент. В этот этап входит получение ответа от сервера о том, что соединение установлено.

2. Сообщение серверу имени пользователя, который собирается с ним работать. В отличие от HTTP-протокола имя пользователя нужно всегда. В случае необходимости анонимного доступа используются зарезервированные имена: anonymous или ftp.

3. Сообщение серверу пароля данного пользователя. FTP-клиент может это сделать автоматически. В случае анонимного доступа стандарт требует, чтобы указывался почтовый адрес пользователя.

4. Далее следует процесс управления FTP-сессией с помощью команд. Самые типичные: сменить директорию (каталог), показать содержимое каталога.

5. Команда, подготавливающая передачу (пассивное или активное соединение). Тип зависит от того, кто будет устанавливать дополнительное соединение - сервер или станция. Если дополнительное соединение устанавливает сервер - активное, клиент - пассивное. Не все сервера и не все клиенты поддерживают пассивное соединение.

6. Команда на передачу файла

7. Собственно передача файла. Она происходит по независимому соединению. После команды на передачу файла клиент или сервер устанавливают дополнительное соединение - так называемое соединение для данных. С помощью TCP-протокола организуется еще один канал связи, по которому и передается файл.

8. Далее клиент решает, что делать с файлом - запустить на выполнение, просмотр, записать на диск и т.п.

9. Последний этап - команда на разрыв соединения.

### **Регистрация ресурса и дальнейший промоушен Web-страниц и сайтов.**

Общий порядок размещения и продвижения Web-сайтов в сети Интернет (SEO-оптимизация).

1. "Теговая" подготовка htm-страниц перед публикацией
2. Загрузка подготовленных материалов на сервер, предоставляющий услуги размещения или хостинга.
3. Регистрация ресурса в поисковых системах и каталогах.
4. Регистрация на Досках объявлений.
5. Организация обмена баннерами в Баннерных системах и линками.
6. Выполнение последующих мероприятий по увеличению посещаемости ресурса.

### **Регистрация.**

Одним из важных этапов в раскрутке сайта является регистрация его в поисковых системах и тематических каталогах России, а если сайт имеет английскую версию, то и в мировых. Бывалые веб-дизайнеры утверждают, что тратят по несколько часов на этот этап, а веб-студии берут за регистрацию только что испеченного сайта от \$5 до \$20.

Способы регистрации сайта:

Во-первых, стоит поговорить о бесплатных (и условно-бесплатных) услугах как [Add Me](#), [Submit It](#) и о нашем аналоге [Submitter.ru](#). На этих сайтах Вам необходимо заполнить специальную форму информацией о Вашем сайте (название, описание, ключевые слова ...). Затем выбрать из списка поисковых систем нужные и система сама автоматически регистрирует Ваш сайт в выбранных поисковиках.

Какой же эффект от подобных сайтов? Обещают 80-100% точность регистрации. На самом деле подтверждение о регистрации на e-mail приходит с не более 30% выбранных поисковиков. Так что судите сами. Наверное, не стоит пренебрегать данным способ регистрации, но и надеяться только на него было бы глупо.

Ниже сравнительная характеристика самых известных мировых сайтов регистраторов:

<a href="#">Add Me</a>	34	0	Very popular
<a href="#">Submit It</a>	400	59	Free Trial
<a href="#">Self Promotion</a>	100	0	-
<a href="#">Usubmit</a>	800	0	-
<a href="#">Submit Plus</a>	1000	39	Free Trial
<a href="#">LinkoMatic</a>	450	0	Submit to FFA
<a href="#">FreeSubmit</a>	16	0	-
<a href="#">Broadcaster</a>	?	82	-
<a href="#">Add4Free</a>	?	0	-
<a href="#">Position Agent</a>	?	0	Only rating
<a href="#">Submit Shack</a>	?	0	-
<a href="#">Recommend-it</a>	?	0	Affiliate Program

Есть также программы регистраторы, которые автоматически регистрируют Ваш сайт в выбранных поисковых системах и тематических каталогах. Принцип действия схож с сайтами регистраторами. При автоматической регистрации тоже используются встроенные скрипты. Безусловно, эффект от таких программ гораздо больше, так как за каждую надо заплатить от \$15 до \$300.

Итак, что же делать, если регистрация с помощью специализированных сайтов не достаточна, а платить по \$20 за программы не хочется. Конечно же, делать всё своими руками. А в чем сложность регистрации руками?

Страницы, где можно заполнить регистрационную форму и зарегистрировать свой сайт:

Aport	+	<p>Поисковая система работает со всеми русскими кодировками, имеет гибкий язык запросов, есть возможность перевода запроса с русского на английский язык и наоборот. Результаты поиска сортируются по степени значимости, вместе со ссылкой отображается фрагмент текста, где встречается термин, а также дата и время последней модификации файла.</p>
Rambler	+	<p>Информационно-поисковая система. 2млн. страниц на 13 тыс. серверов. Учет времени создания. Поиск в группах новостей. Одна из самых популярных российских поисковых систем. Поисковая система Rambler поддерживает рейтинг русских страниц Top100. Списки страниц разбиты на группы и многие пользователи используют данный рейтинг как каталог.</p>
List	+	<p>Интересным аспектом каталога является наличие гидов - реальных людей, поддерживающих определенные разделы каталога. Сегодня данный каталог является самым обширным.</p>
LOOK	+	<p>Каталог ресурсов Интернета. Каждый ресурс имеет описание и баннер (88x31). Ведутся списки TOP10.</p>
WebList	+	<p>"List of Russian Servers" - один из старейших каталогов. Имеется русская и английская версия каталога. Второй адрес <a href="http://www.yahoo.ru">www.yahoo.ru</a>.</p>

Ivan Susanin	+	Каталог только российских интернет-ресурсов. Каждая ссылка имеет подробный комментарий. Ресурсы выстроены по названию в алфавитном порядке. URL-адрес ресурса появляется внизу страницы при наведении курсора на название ресурса.
Весь RUcский интерNET	+	Тематический каталог аннотированных ресурсов. Обновляется ежедневно.
UP	+	UP.ru; каталог РуНета, бизнес, рефераты, чаты, пр...
Find It!	+	Каталог ресурсов Интернета. Поиск, халява, путешествия, библиотеки, бизнес, развлечения, архитектура и др. Рейтинг.

Вторая проблема, на которую требуется больше всего времени это заполнение регистрационный форм на сайтах соответствующих поисковых систем и тематических каталогов. Слишком долго и неприятно набирать одну и ту же информацию о своём сайте по десять раз.

Так как информация, необходимая для заполнения всегда одна, то можно один раз записать её в какой-нибудь текстовой файл и потом, используя буфер обмена вставлять в формы регистрации.

В текстовой файл необходимо записать:

Название сайта - отнеситесь к названию сайта очень серьёзно. Обычно сайт называется по названию проекта.

Владелец сайта - то есть Ваше ФИО.

URL сайта - не забудьте, что адрес сайта должен начинаться с http://....

E-mail владельца - на этот адрес будет приходить уведомление о том, что Вы занесены в каталог.

Логин - Часто поисковые системы идентифицируют новые сайты по ID (числу), но иногда используется логин для изменения и удаления информации о ресурсе. Желательно слово из 4-6 символов.

Пароль - Иногда поисковые системы высылают свой пароль, но чаще предлагают задать его при регистрации. Желательно слово из 4-6 символов.

Описание сайта - Краткое описание сайта от 150 до 250 символов. Попробуйте передать суть сайта, а не рекламировать свои продукты или услуги. Желательно не использовать слова “единственный”, “лучший” и т. п., так как за такое описание Вам могут отказать в регистрации или поменяют

описание по своему усмотрению. Так, например, поступают в тематическом каталоге List.ru.

Ключевые слова - Около двадцати слов, по которым поисковые системы найдут сайт. Можно посмотреть на сайт конкурентов, у которых хороший рейтинг и использовать их ключевые слова (неужели и это плагиат?). Лучше всего записать ключевые слова два раза: через запятую и пробел.

Когда текстовый файл создан можно смело начинать регистрацию, используя таблицу поисковых систем Рунета из этой статьи, созданный текстовый файл и буфер обмена. На регистрацию сайта на одном сервере уходит от 30 секунд до 3 минут. Т. е. На регистрацию в десяти поисковиках необходимо от 5 минут до получаса.

Программа автоматической регистрации [AddSite](#). Она загружает страницу добавления сайта, используя большие базы данных по российским поисковым системам и тематическим каталогам и автоматически заполняет форму регистрации. Остается проверить правильность заполнения и нажать кнопку “Послать”. Еще программа может систематизировать информацию, полученную от поисковых систем после регистрации (HTML-код, ID, свои комментарии...). За программу надо заплатить от 1600 до 10000 рублей в зависимости от возможностей. Но есть Trial-период 20 дней без функциональных ограничений.

Итак, пора подводить некоторые итоги. Хочется сказать, что Рунет далеко не впереди планеты всей в области регистрации сайтов в поисковиках. Мало программ и сайтов облегчающих регистрацию. Но, все же воспользовавшись способами можно хорошо повысить трафик сайта, хотя большую роль играют описания и (самое главное) содержание сайта. Если приходится регистрировать десятки сайтов ежемесячно, то, наверное, целесообразней было бы купить какую-нибудь программу, т.к. она очень быстро окупится.

#### **Полезные ссылки.**

<http://www.iwr.com/free/submitmaster.htm> - перечень разнообразных средств для регистрации вашего сайта в десятках основных и второстепенных поисковых машин, каталогов и на бесплатных досках объявлений.

<http://myfreeofficelinks.com/submit/index.cgi?s=myfreeoffice&p=losttema> - одно нажатие и вы прописались в 9 главных поисковых машинах.

<http://www.all4one.com/all4submit/> - регистрация в главных поисковых машинах с верификацией.

<http://jimtools.com/> - мощное средство для регистрации сайта в огромном количестве каталогов: 52 поисковых машины, 1160 FFA (Free For All message boards), 144 каталога.

<http://www.aaa.com.au/submit/>- одно нажатие и вы прописались в 13 главных поисковых машинах.

[http://www.tti.co.jp/add\\_url/](http://www.tti.co.jp/add_url/)- одно нажатие и вы прописались в 33 главных поисковых машинах.

<http://linkplace.com/winner/> - регистрация в поисковых машинах, линкообменниках, обмен баннерами и много других "вкусностей".

**SUBMITBLASTER** - ваша ссылка будет добавлена в каталог 21-й поисковой машины и вы увидите результат.

**WEST COLORADO SUBMIT** - ссылка на Ваш сайт будет добавлена в каталог 9 поисковых машин.

<http://iwr.com/free/reciprocal.htm> - обмен баннерами.

Вопросы для закрепления.

- 1) Дайте понятие Web – хостинга?
- 2) Дайте понятие web – сервера .
- 3) Назовите способы регистрации сайта в сети Интернет

Домашние задание:

Подготовить доклад на тему: «Публикация сайтов в Интернет и их дальнейшее сопровождение, «Установка web-сервера Apache», «Администрирование web-сервера Apache»