

Сергей Науменко



Создаем сайт своими руками
10 практических уроков

<http://sitemanaging.ru>



Начинающий инфобизнесмен?

"Наконец-то вы можете реально избавиться от ВСЕХ технических проблем, которые угрожают вашему инфобизнесу".

Горькая правда о причине нулевого заработка

80% новичков инфобизнеса.



<http://cyberlux.ru/tehmomenty>

Уважаемый читатель! Перед вами уникальная книга, которая всего за 10 уроков научит вас основам замечательного занятия – созданию WEB-сайтов.

Сегодня сайт – это окно в мир, это ваша витрина, это инструмент для заработка денег. Овладев навыками сайтостроения, ваша жизнь изменится, потому что вы станете профессионалом в области создания сайтов. Вы станете экспертом WEB-дизайна. Вы сможете зарабатывать деньги, создавая сайты для себя и ваших клиентов.

Оглавление

Зачем нужна эта книга?	6
Об авторе	6
Что написано в книге?	7
Последние напутствия	7
Задание № 1	12
Изучаем HTML	13
Что такое HTML?	13
Теги раздела head	16
Теги раздела body	17
Задание № 2	25
HTML –редакторы	27
SeaMonkey	28
NVU	34
Dreamweaver	37
Гиперссылки	46
Дополнительные атрибуты тега <a>	49
Задание № 3	56
Задание №4	56
Списки	58
Маркированные списки	59
Нумерованные списки	60
Список определений	63
Вложенные списки	64
Таблицы	66
Представление таблицы	70
Использование таблиц	71
Задание № 5	72
Формы	75
Тег формы	75

Элементы формы.....	76
Текстовое поле.....	76
Чекбокс или флажок.....	77
Радиокнопка.....	77
Поле для передачи файла.....	77
Кнопки.....	78
Кнопка-изображение	79
Скрытое поле.....	79
Многострочное поле для ввода большого текста	79
Список и поле со списком для выбора	80
«Красивая» форма	80
Введение в CSS.....	83
Как выглядит CSS код.....	84
Атрибут class.....	85
Свойство :hover	86
Атрибут id.....	87
Углубление в CSS.....	92
Единицы длины	92
Псевдоэлемент :first letter	93
Псевдоэлемент :first-line	93
Свойство font-family.....	94
Свойство font-size.....	94
Свойство font-style	95
Свойство font-variant	95
Работа с блоками	96
Фоновое изображение	96
Округление вершин блоков.....	97
Публикация сайта в интернете	100
Как действовать дальше?.....	113

Зачем нужна эта книга?

Сегодня в интернете так много материала, посвященного созданию сайтов! Можно задуматься, зачем еще писать какие-то книги на эту тему? Их великое множество!

Да! Действительно! Материалов на эту тему очень много в интернете в свободном доступе. Мало того, сейчас столько программ для создания сайтов, таких как Joomla!, WordPress, Битрикс, NetCat, авторы которых трубят на весь мир об их простоте. О том, что они ориентированы на конечного пользователя, и что каждый, кто загрузит бесплатную программу или приобретет платную, за 15 минут сможет самостоятельно создать сайт профессионального качества.

Но это лишь удачный маркетинговый ход, потому что ни одна из подобных универсальных программ на самом деле не рассчитана на конечного пользователя. Обычный пользователь вряд ли разберется во всех тонкостях работы с такими программами. Ему либо придется долго изучать руководство, либо нанимать профессионалов, которые смогут разобраться в этих программах. Для разработки сайтов нужны базовые знания и практические навыки, без которых нормальный сайт создать невозможно даже при наличии таких умных программ.

Для того чтобы научиться делать сайты, необходима постоянная практика. Без практики любая теория останется лишь теорией. Поэтому в этой книге дается в основном практический материал. Та самая выжимка, необходимая и достаточная для практического создания сайтов.

В этой книге нет воды. Весь материал разбит на 10 практических уроков, после выполнения которых, вы научитесь делать статические WEB-страницы и публиковать их в интернете.

Об авторе

Автор книги, разумеется, Я! Науменко Сергей Станиславович (Oknetuan).



Я более 10 лет профессионально занимаюсь созданием сайтов, интернет-магазинов и автоматизированных систем на базе WEB-технологий. Преподаю WEB-дизайн и WEB-программирование в ВУЗах, центрах дополнительного образования, а также в виде персональных коучингов. Занимаюсь также интернет-маркетингом, консалтингом и продвижением сайтов.

Разрабатывать сайты я начал, когда появилась потребность в продаже физических товаров, а именно, компьютерных комплектующих, через интернет. Именно тогда, в 2003 году я разработал свой первый движок интернет-магазина, который написал за 2 недели и который успешно проработал 3 года, пока я его не модернизировал. Именно тогда я понял, что разработкой сайтов можно зарабатывать деньги, и я открыл новое направление в нашем бизнесе. Кроме продажи компьютерных комплектующих мы начали заниматься разработкой сайтов, интернет-магазинов, обучающих систем и других информационных проектов, в основе которых лежат WEB-технологии.

Что написано в книге?

В книге описаны практические уроки. Их всего десять.

Первый урок посвящен общим представлениям о сайте, о файловой структуре сайта и в качестве практического упражнения предлагается создать первую WEB-страницу и отобразить ее в браузере.

На втором уроке вы узнаете, как устроен язык гипертекстовой разметки HTML, как устроен WEB-документ, и на основе полученных знаний сделаете WEB-страницу с использованием HTML.

На третьем уроке состоится знакомство с HTML редакторами – программами, которые облегчают труд WEB-дизайнера.

На четвертом уроке вы изучите устройство гиперссылок, комментариев и сделаете многостраничный сайт, который работает.

Пятый урок посвящен изучению списков. Списки необходимы не только для представления текста, они также являются основой для создания меню навигации.

Шестой урок посвящен изучению таблиц. Таблицы также имеют двойное назначение. Кроме табличного отображения данных их можно использовать для верстки WEB-страниц. Мы применим табличную верстку на практике.

На седьмом уроке вы изучите устройство и назначение HTML-форм. Формы необходимы для интерактивного взаимодействия пользователя и сайта. На практическом занятии вы отформатируете форму с помощью таблицы.

Восьмой урок повысит ваш уровень на следующую ступень, потому что вы перейдете к изучению и применению каскадных таблиц стилей, без которых невозможно создавать современные сайты. На практическом занятии вы создадите WEB-страницу с помощью блочной верстки.

На девятом уроке вы углубитесь в изучение каскадных таблиц стилей, что позволит вам расширить и закрепить полученные знания и стать специалистом по созданию статичных WEB-сайтов.

Десятый урок научит вас публиковать сайты в интернете.

Последние напутствия

В книге приводится материал, который в достаточной степени пригоден для того, чтобы вы смогли начать проектировать WEB-сайты и публиковать их в интернете. Книга дает знания начального уровня. Полученные знания обязательно надо подкреплять практикой. Здесь не приводятся обширные справочные данные. Здесь только то, что необходимо для освоения практического сайтостроения.

Удачного прочтения!

Урок №1

На этом уроке мы рассмотрим базовые вопросы:

- Общее представление о сайте
- Представление сайта с точки зрения файловой системы
- Ваша первая WEB-страница

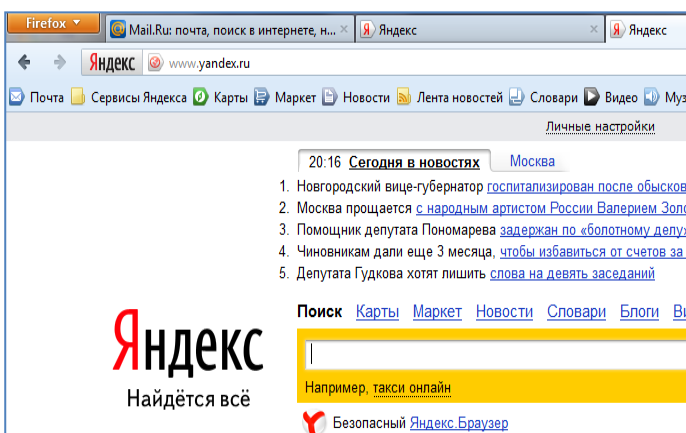
Для выполнения задания вам понадобится компьютер, так как без компьютера сайт не создашь.

Итак, Что же такое сайт?

В старой польской энциклопедии было написано: Что такое лошадь – знают все.

Про сайт можно сказать то же самое: Что такое сайт - знают все!

Сайт, в представлении современных людей это то, что мы видим в браузере нашего компьютера после того, как в строке запроса набираем нечто вроде <http://yandex.ru> и потом нажимаем Enter или специальную кнопку в браузере.



Это наше ментальное представление о сайте. Примерно таким мы хотим видеть наш сайт, и поэтому мы будем учиться его делать пошагово от простых упражнений до более сложных, пока не научимся делать сайты своими руками.

Кстати, Браузер – это основная программа в компьютере с помощью, которой отображаются сайты в том виде, в каком их сделал веб-мастер или веб-дизайнер.

Совершенно не важно, каким браузером вы привыкли пользоваться, Интернет-эксплорером, Файер-фоксом, Оперой или Яндекс-браузером, главное чтобы вы напрочь забыли про старые браузеры, а именно, Интернет-эксплорер 6 и 7 версии. Это плохие браузеры, они неправильно отображают сайты.



Все остальные браузеры работают нормально, и современные сайты рассчитаны на работу именно в этих популярных браузерах. Лично я пользуюсь Файер-фоксом.

Теперь посмотрим на устройства сайта изнутри.

Мы знаем, что есть где-то WEB-сервер, есть интернет и есть клиент – компьютер пользователя.

На WEB-сервере в его файловой системе есть папочка с названием www, в которой находятся папки и файлы конкретного сайта.



Имя	↑ Тип	Размер	Дата
[.]			<Папка> 02.04.2013 20:40
[css]			<Папка> 02.04.2013 20:37
[images]			<Папка> 02.04.2013 20:37
about	html	0	02.04.2013 20:38
contacts	html	0	02.04.2013 20:39
galery	html	0	02.04.2013 20:40
index	html	0	02.04.2013 20:38

Сейчас я привожу упрощенную файловую структуру сайта. В реальном сайте может содержаться гораздо больше папок и файлов. Однако именно эта структура является достаточной для создания большинства статичных сайтов.

Самая первая в нашем списке папка с названием CSS.

Это папка для хранения файлов, описывающих стили нашего сайта.

Аббревиатура CSS обозначает (*Cascading Style Sheets* — каскадные таблицы стилей). Именно эта папка отвечает за внешний вид сайта, цвет и рисунок фона, шрифт заголовков и параграфов, взаиморасположение элементов дизайна на сайте, вид навигационных кнопок меню и многое другое. Каскадные таблицы стилей мы с вами будем изучать позже.

Следующая папка images. В ней содержатся файлы с картинками, которые используются в дизайне сайта. Данные картинки являются элементами дизайна веб-страниц. То есть там хранятся все картинки, без которых нельзя обойтись при создании сайта.

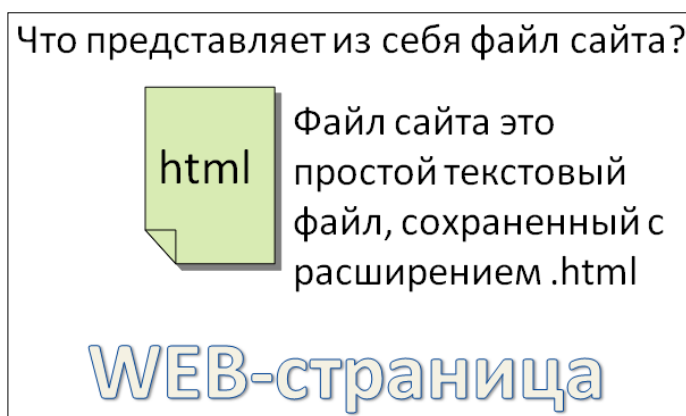
Как я уже говорил, папок может быть больше. Все остальные папки вы будете создавать самостоятельно при необходимости. Для создания вашего первого учебного сайта достаточно этих папок.

Теперь посмотрим на файлы, которые лежат в корне сайта. Это и есть файлы веб-страниц. Одна страница – один файл.

Обратите внимание на файл с названием index.html. Именно этот файл является первым файлом, который открывается при первичном вызове сайта в адресной строке браузера.

Что представляют из себя файлы веб-страниц?

На самом деле в них нет ничего сложного. Это простой текстовый файл, который можно создать в простом текстовом редакторе Блокнот и сохранить его как файл с расширением html.



В нашем примере вы видите три файла, по их названиям можно судить об их назначении.

- index-это файл главной страницы сайта, которая открывается по умолчанию
- about-страница рассказывающая о компании
- contacts-страница с контактами
- gallery-Страница с фотогаллереей.

В зависимости от назначения сайта и его информационной структуры планируются необходимые веб-страницы с необходимым наполнением, или как принято говорить в среде веб-дизайнеров – с необходимым контентом.

Каждая страница сайта в данной файловой структуре представлена в статичном виде. То есть ее контент создается вручную непосредственно в файле. Чтобы его изменить, необходимо вручную залезать в файл, вызывать его с помощью редактора файлов и руками вбивать новый контент.

Такой статичный сайт подобен печатной книге, в которой каждая страница (файл) напечатана раз и навсегда, чтобы исправить текст, необходимо изменить файл, то есть отредактировать страницу.



Каждый файл сайта – это простой текстовый файл, который можно создать вручную в простом текстовом редакторе Блокнот и сохранить его как файл с расширением html.

Например, вызвать блокнот, записать в нем что-нибудь, и сохранить его в папке www с именем index.

По умолчанию он там сохранится с расширением txt, но потом можно будет просто переименовать его расширение на html и он станет WEB-страницей. И если кликнуть по нему мышкой, он откроется в браузере, как настоящая WEB -страница.

Именно этому посвящено первое задание, которое вы должны выполнить для закрепления материала урока.

Задание № 1

Создаем вашу первую WEB-страницу

- Создайте у себя на компьютере папку с названием, например: Уроки по созданию сайта.
- В этой папке создайте папку с названием www (маленькие латинские буквы)
- Необходимо запомнить, что все имена папок и файлов на сайте нужно писать только латинскими буквами, при этом использовать только маленькие буквы. Дело в том, что многие веб-серверы не распознают кириллические буквы и могут появиться ошибки, если файлы будут заданы не латинскими буквами. А маленькие буквы рекомендуется использовать, чтобы не допускать путаницы с именами.
- В папке www создайте с помощью текстового редактора «Блокнот» файл, в котором напишите текст «Это моя первая WEB-страница».
- Сохраните этот файл под именем index.html
- Кликаем по файлу index.html, он у вас откроется в браузере

Урок № 2

Изучаем HTML

На первом уроке вы должны были усвоить, что файл веб-страницы представляет собой обыкновенный текстовый файл, который можно создать с помощью простого текстового редактора Блокнот, сохранив его с расширением html. Мы в этом с вами убедились, создав первую веб-страницу и открыв ее в браузере.

Однако ваша первая веб-страница еще очень далека от идеала и на настоящую веб-страницу мало похожа. Потому что в такой простейшей веб-странице напрочь отсутствует разметка текста, позволяющая отображать его удобным для восприятия.

Сегодня мы будем изучать, каким образом в настоящих веб-страницах выполняется разметка текста.

На сегодняшнем занятии мы рассмотрим следующие вопросы:

- Что такое HTML
- Устройство WEB-документа
- Создание шаблона «пустого» web- документа
- Теги HTML раздела head
- Теги HTML раздела body
- Создание WEB-страницы с использованием HTML

Для занятий вам понадобится ваш компьютер.

На компьютере у вас уже есть созданная вами папка www и файл index.html

Итак, приступим! Минимум теории – максимум практики.

Что такое HTML?

Это аббревиатура, которая расшифровывается как (Хайпотекст мокап лэнгвич), HyperText Markup Language, что в переводе с английского языка обозначает «Язык разметки гипертекста».

HTML не является языком программирования. Это язык инструкций.

Вот принцип его действия:

Поставим перед собой следующую задачу: написать заголовок текста, под ним 2-3 абзаца обычного текста с иллюстрацией, причем текст должен красиво огибать иллюстрацию.



HTML размечает текст следующим образом:

Принцип действия HTML

`<Заголовок самый жирный>Lorem ipsum</Заголовок самый жирный>`

`<картинка images/river.jpg расположенная слева от основного текста шириной 200px>`

`<абзац>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation.</абзац>`

`<абзац>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation.</абзац>`

`<абзац>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation.</абзац>`

Эти самые инструкции, заключенные в угловые скобки «<>» называются тегами. Из примера видно, что есть открывающий тег, который стоит спереди, и закрывающий тег со слешем, стоящий сзади.

Кроме того, в случае с изображением, есть одиночные теги, у которых отсутствует закрывающий тег.

Теги предписывают, каким образом должно отображаться в браузере то, что заключено между ними.

Теперь мы с вами изучим структуру настоящего веб-документа.

Главным тегом HTML является его одноименный тег `<html>...</html>`.

Все, что находится между открывающим и закрывающим тегом `html`, является web-документом.

Сам веб-документ состоит из двух разделов. Это разделы `head` и `body`.

Голова и тело.

Структура WEB-документа



Раздел `head` является служебным разделом и его содержимое не отображается в браузере. Но это очень важный раздел, который отвечает за правильную работу веб-страницы.

Раздел `body` предназначен для размещения в нем контента сайта.

В браузере отображается именно то, что размещено между тегами `<body>` и `</body>`

Вот html код пустого web-документа:

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
    <title> </title>
  </head>
  <body>
  </body>
</html>
```

Чуть позже я расскажу назначение всех незнакомых вам в этом примере слов, а сейчас я вам предлагаю внимательно его переписать к себе в рабочую тетрадь, соблюдая все знаки

препинания, без грамматических ошибок, все кавычки, точки с запятой, угловые скобки, слэши. Все переписываем с пунктуальной точностью.

Затем, создайте у себя в папке `www` файл `blank.html` и набейте в нем этот код ручками, чтобы вы привыкли писать теги именно руками. Первый раз нужно обязательно создать шаблон пустого документа своими руками. Именно в ручном наборе кроется секрет обучения.

В дальнейшем вы сможете его копировать для других страниц, но первый, пожалуйста, сделайте вручную.

Теги раздела `head`

Сейчас я расскажу о назначении служебных тегов в разделе `<head>`

```
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
```

Вот эта строка называется Метатег, что видно из его названия. Тег относится к разряду одиночных, он не имеет закрывающего тега.

Сейчас вам совершенно не обязательно знать, как он устроен, потому что про метатеги в данной книге мы говорить не будем.

Его назначение – правильно установить кодировку русского кириллического текста для отображения в браузере.

Если бы весь мир разговаривал только на английском языке и использовал латинские буквы английского алфавита, то этого тега не было бы в природе.

Из-за того, что в мире кроме латиницы существует множество других национальных шрифтов, таких, как кириллица, арабская вязь, еврейский шрифт, восточные иероглифы, а также национальные шрифты и их разновидности, то для корректного отображения национального шрифта необходима предустановка его кодировки.

В нашем случае, для корректного отображения кириллицы во всех браузерах мира, мы устанавливаем на нашей веб-странице, с помощью данного метатега, кодировку `windows-1251`, которую, специально для российских пользователей, разработала компания MICROSOFT.

На текущем этапе обучения вам просто нужно, чтобы этот метатег всегда присутствовал на всех ваших веб-страницах.

Следующий, очень важный тег раздела `head` это тег

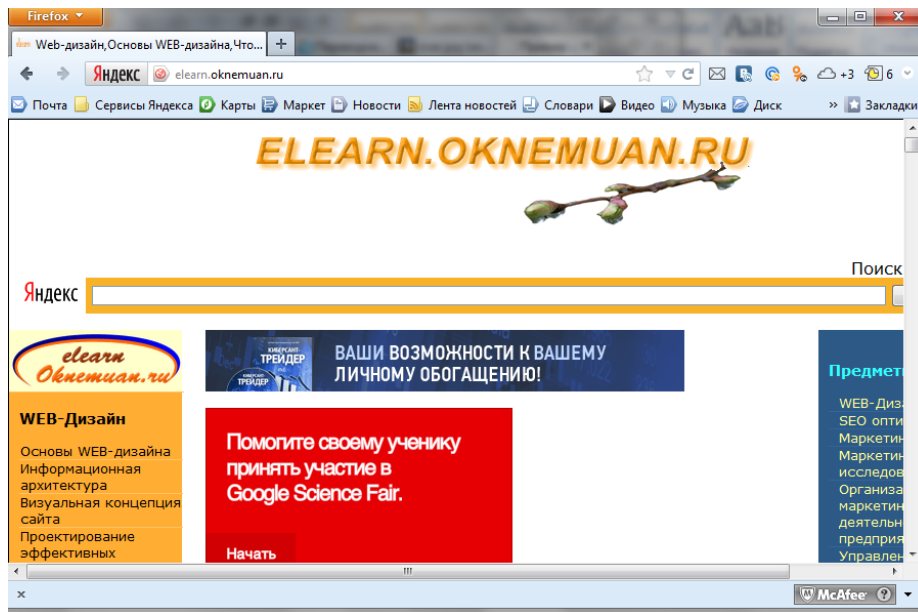
```
<title> </title>.
```

То, что размещено между открывающим тегом `title` и закрывающим тегом `title` называется заголовком сайта.

Это очень полезный тег. Он жизненно необходим для того, чтобы роботы поисковых систем могли находить вашу веб-страницу по ключевым словам, которые помещаются между тегами `title`.

Содержимое этого тега отображается в верхней части браузера, когда в него загружена веб-страница.

Ниже на иллюстрации видно, что заголовок сайта написан на вкладке « WEB-дизайн, Основы WEB-дизайна, Что...»



В разделе head в настоящих боевых сайтах присутствует множество других служебных тегов. На сегодняшний момент вам достаточно знать только о двух, про которые я вам рассказал.

Теги раздела body

Переходим к основному разделу отображения контента. Это раздел body.

Все то, что мы видим в браузере помещено между тегами `<body>` и `</body>`.

Теги раздела body

- Все то, что мы видим в браузере помещено между тегами `<body>` и `</body>`.

`<body>`

Весь контент сайта размещать ЗДЕСЬ!!!!!!

`</body>`

Это важно запомнить. Многие новички при выполнении упражнений по web-дизайну забывают об этом и пишут код контента веб-страницы где угодно, только не между тегами body, а потом удивляются, почему у них веб-страница в браузере ничего не отображает.

Вспомним нашу задачу про заголовок, абзацы и картинку.



Сейчас мы ее реализуем по-настоящему.

Для начала вспомним про заголовок сайта.

В данном примере это «Пример разметки текста с иллюстрацией».

Помещаем его между тегами title раздела head.

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
    <title> Пример разметки текста с иллюстрацией </title>
  </head>
  <body>
    </body>
</html>
```

Теперь в разделе body создаем жирный заголовок для текста. Прошу вас не путать заголовок сайта с заголовками текста. Это разные разделы и разные теги.

Итак! Записывайте в свою рабочую тетрадь!

В HTML существует 6 уровней заголовков для текста.

Это

`<h1>Самый жирный заголовок</h1>`

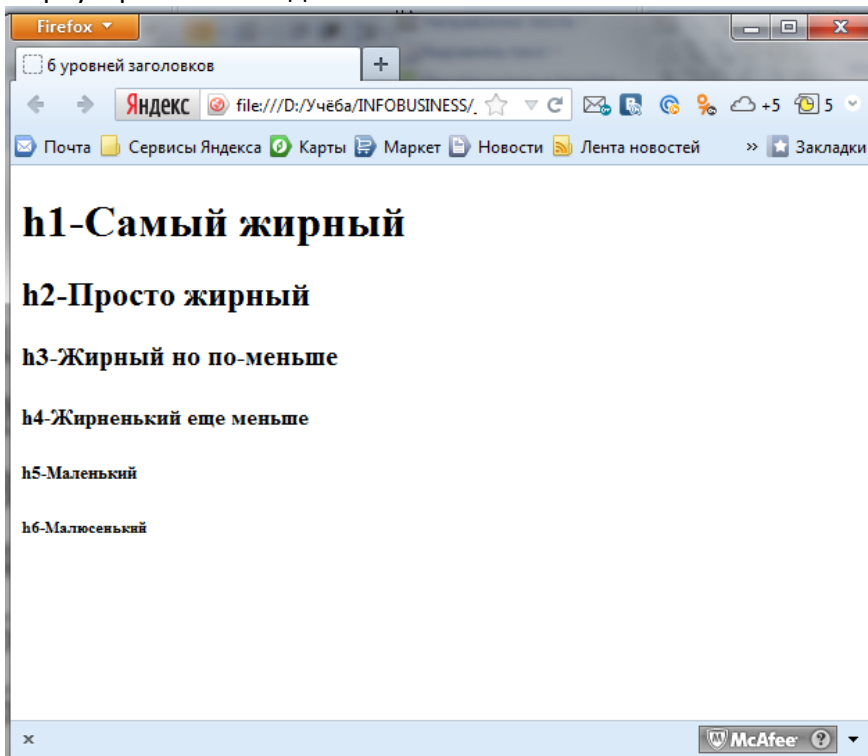
.....

`<h6>Самый малюсенький, но тоже жирный заголовок</h6>`

`<h2>,<h3>,<h4>,<h5>` Заголовки с убывающей жирностью.

Все заголовки текста имеют открывающий и закрывающий тег.

В браузере это выглядит так



Возвращаемся к нашей задаче и записываем жирный заголовок первого уровня.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
<title> Пример разметки текста с иллюстрацией </title>
</head>

<body>
  <h1>Lorem ipsum</h1>
</body>
</html>
```

Далее опишем разметку абзацев или как их еще называют – параграфов.

Для этого в арсенале html имеется специальный тег `<p></p>`.

Он собственно и происходит от слова параграф.

То, что заключено между тегами `<p>и</p>` будет параграфом.

Обратите внимание на то, каким образом тег параграфа размечает текст:



Все параграфы отделены друг от друга промежутком. В этом и заключается особенность тега `<p>`. Он по умолчанию создает эти промежутки.

Возвращаемся к коду и добавляем туда три абзаца или параграфа. Для простоты трижды копируем одну и ту же конструкцию:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
<title> Пример разметки текста с иллюстрацией </title>
</head>

<body>
  <h1>Lorem ipsum</h1>
  <p >Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor ...</p>
  <p >Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor ...</p>
  <p >Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor ...</p>
</body>
</html>
```

Для удобства демонстрации в приведенном листинге кода я сократил объем текста. В упражнении, которое я предложу вам сделать в конце урока используйте текст объемом побольше. Текст знаменитой на весь мир рыбы Lorem ipsum вы можете взять на сайте:

<http://www.lipsum.com>

Теперь нам осталось разместить иллюстрацию, которую текст будет плавно огибать справа. Для работы с изображениями существует тег ``. Это одиночный тег, который содержит атрибуты и их параметры.

Вот как он выглядит в нашем примере:

```

```

Его мы помещаем между заголовком текста и первым абзацем. Вот так:

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
    <title> Пример разметки текста с иллюстрацией </title>
  </head>
  <body>
    <h1>Lorem ipsum</h1>
    <br />
    
    <p >Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor ...</p>
    <p >Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor ...</p>
    <p >Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor ...</p>
  </body>
</html>
```

Сейчас мы подробно разберем значение всех атрибутов и параметров тега `img`.

```

```

Атрибут **src** в своем параметре указывает путь к размещению картинки.

В данном случае предполагается, что на сервере, точнее сказать в вашей папке `www` есть папка для хранения изображений, которая называется `images`, а в ней хранится файл иллюстрации с названием `river.jpg`.

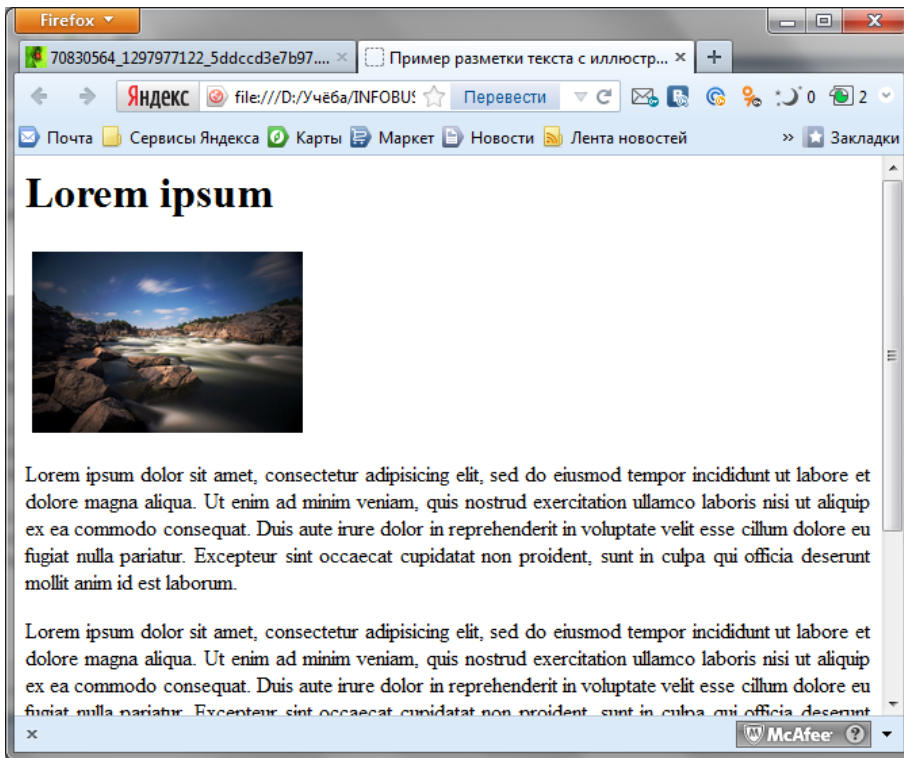
Следует сразу отметить, что если вы хотите отобразить картинку, которая хранится не у вас в папке, а где-то на другом сервере, на чужом сайте, то в качестве параметра **src** следует указывать полный `url` адрес картинки. Например:

```
http://img0.liveinternet.ru/images/attach/c/2/70/830/70830564\_1297977122\_5ddccd3e7b97.jpg
```

Этот адрес я скопировал с Яндекс-картинок, кликнув правой мышкой по картинке и в информации об изображении скопировал ссылку.

Следующий атрибут **align** в своем параметре содержит положение картинки относительно текста. В нашем примере это **left**. То есть благодаря такой установке, картинка будет находиться слева, а текст будет ее обтекать справа и снизу.

Если атрибут **align** не указывать, то текст не будет обгибать картинку.



Следующий атрибут **width** в своем параметре указывает ширину картинка в пикселях. Если удалить этот атрибут, то картинка будет отображаться в своем собственном размере.



Что не всегда удобно. Если у вас очень большие картинки, то они не будут помещаться в браузер, что совершенно не нужно.

Здесь также можно использовать атрибут **height**, параметром которого служит высота картинка


```

```

Однако если вы при задании параметров ошибетесь в размерах, могут нарушиться пропорции изображения.



Поэтому, если вы хотите соблюсти пропорции картинки, но при этом точно не знаете какие параметры ширины и высоты указать, используйте один из атрибутов, либо **width**, либо **height**.

Недостающий атрибут вычислится в браузере автоматически.

Атрибуты **hspace** и **vspace** в своих параметрах задают отступ от картинки до текста в пикселях по вертикали и горизонтали соответственно.

Если эти атрибуты не указывать,

```

```

То текст «прилипнет» к картинке.



Далее мне стоит вам в рамках этого урока рассказать следующее:

Почти все теги могут содержать атрибуты.

Например, тег заголовка можно разместить по центру браузера, используя атрибут **align**.


```
<h1 align="center">Lorem ipsum</h1>
```





Как вы уже успели заметить, если атрибут **align** не указывать, то заголовок по умолчанию прижимается влево. Так же, как если бы мы написали


```
<h1 align="left">Lorem ipsum</h1>
```

Параметры атрибута **align** также могут принимать значения **right** и **justify**.

 align="left"

 align="center"

 align="right"

 align="justify"

Параметр **justify** уместнее применять к параграфам для их выравнивания, тогда они выглядят ровными прямоугольниками.

На этом уроке мы с вами научились форматировать иллюстрированный текст с помощью минимального набора тегов HTML. Уже можно создавать электронную книгу или собственный блог.

Для закрепления пройденного материала, я предлагаю вам самостоятельно выполнить следующее задание, повторить все то, о чем рассказывалось в данном уроке.

Задание № 2

- Создайте в вашей папке `www` папку `images` и поместите туда любое изображение с расширением `jpg`, `gif` или `png`. Вы можете поместить туда сразу несколько изображений
- Создайте веб-страницу `ex1.html` и скопируйте в нее шаблон пустого веб документа из вашего файла `blank.html`.
- Укажите заголовок сайта в тегах `title` раздела `head`.
- В разделе `body` создайте контент, состоящий из заголовка, абзацев с текстом и иллюстраций.
- Попробуйте разместить изображение в любое место текста с разными вариантами отгибания текстом, используя в теге `` варианты параметров атрибута `align` равным `left` и `right`, задавая разные значения параметров атрибутов `hspace` и `vspace`, `width` и `height`.
- Попробуйте применить к заголовку и параграфам различные значения параметров атрибута `align` равные значениям `left`, `center`, `right` и `justify`.

Вот полный текст кода `html` для данного задания. Для компактности здесь сокращен текст `Loren ipsum`

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
    <title> Пример разметки текста с иллюстрацией </title>
  </head>

  <body>
    <h1>Lorem ipsum</h1>

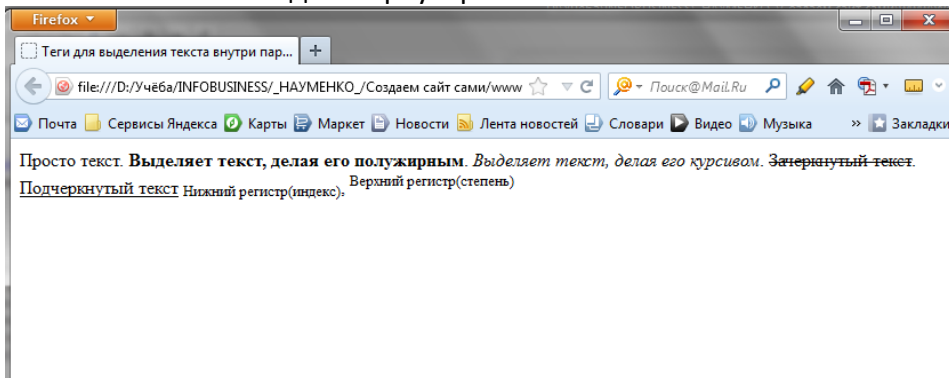
    

    <p >Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor ...</p>
    <p >Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor ...</p>
    <p >Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor ...</p>
  </body>
</html>
```

Еще напоследок несколько тегов для выделения текста:

```
<p>
  Просто текст.
  <strong>Выделяет текст, делая его полужирным</strong>.
  <em>Выделяет текст, делая его курсивом</em>.
  <del>Зачеркнутый текст</del>.
  <ins>Подчеркнутый текст</ins>
  <sub>Нижний регистр(индекс)</sub>,
  <sup>Верхний регистр(степень)</sup>
</p>
```

Вот как все это выглядит в браузере:



Урок № 3

HTML –редакторы

Сегодня я постараюсь облегчить вам жизнь. Я открою вам секрет, с помощью которого вам станет легче писать HTML код.

Если вы добросовестно выполнили задание № 2 прошлого урока, то, наверное, поняли, что писать HTML код руками в блокноте очень трудоемкое и неблагодарное дело.

Задание было очень легкое, объем кода был совсем небольшой, но вы, наверное, почувствовали, что при написании даже такого простого задания очень легко запутаться и ошибиться.

Однако человечество уже давно и успешно решило эту задачу и создало замечательное и удобное программное обеспечение, которое облегчает труд создателя WEB-страниц.

Это программное обеспечение является визуальными редакторами WEB-страниц.

Визуальный редактор WEB-страниц или редактор HTML кода имеет одно замечательное свойство. Он сам создает HTML код и значительно облегчает процесс разработки WEB-страниц.

Вам предстоит много работы по изучению технологии создания WEB-страниц, с каждым уроком код HTML, который мы с вами будем осваивать и применять на практике, будет усложняться. Поэтому мы для удобства работы часто будем пользоваться WEB-редактором.

В настоящее время создано множество редакторов HTML. Все они делятся на платные и бесплатные.

Для работы со статичными WEB-сайтами вполне достаточно использовать бесплатные редакторы HTML кода. Для WEB-программирования необходимо использовать профессиональные платные редакторы.

Самый популярный удобный профессиональный платный редактор web-страниц это, несомненно, Dreamweaver производства компании Adobe. Скачать триальную версию его можно на официальном сайте

<http://www.adobe.com/ru/products/dreamweaver.html>

Сейчас уже вышла новая версия Adobe Dreamweaver CS6. Я до сих пор пользуюсь версией CS3.

Это очень хороший редактор. Единственный его недостаток – относительно высокая цена. Но для профессиональной работы по созданию сайтов это наилучший редактор.

Если Вы пока не планируете вкладывать деньги в приобретение профессионального ПО, на начальном этапе можно пользоваться бесплатными редакторами.

Из бесплатных редакторов HTML кода на сегодняшний день могу с уверенностью рекомендовать два.

Это Компоновщик из состава продукта SeaMonkey от Mozilla и редактор HTML кода NVU, который тоже создан на основе продуктов от Mozilla.

Скачать и установить их можно по нижеприведенным ссылкам

<http://mozilla-russia.org/products/seamonkey/>

<http://nvu.mozilla-russia.org/index2.php>

На нашем уроке я покажу как пользоваться этими тремя редакторами, а вы для себя определитесь, что вам больше подходит.



Adobe **Dreamweaver** CS6

<http://www.adobe.com/ru/products/dreamweaver.html>



Компоновщик из состава продукта **SeaMonkey**

<http://mozilla-russia.org/products/seamonkey/>



HTML-редактор **NVU**

<http://nvu.mozilla-russia.org/index2.php>

Начну с бесплатных редакторов от Mozilla.

SeaMonkey

В принципе проект Seamonkey это три в одном. Это великолепный браузер, это HTML редактор, к тому же еще это почтовый клиент.

Заходим по ссылке на сайт и скачиваем его себе на компьютер.

<http://mozilla-russia.org/products/seamonkey/>

Наши проекты: Bugzilla MozUtil Nvu Pastebin Планета

mozilla Россия
russian mozilla team

поиск по сайту: Поиск

Продукты Поддержка Форум Документация Ссылки О проекте

Главная
Продукты
Mozilla Firefox
Mozilla Thunderbird
Bugzilla
ChatZilla

SeaMonkey

Русские сборки

- Windows (инсталлятор) (21,8МБ)
- Windows (zip-архив) (27,4МБ)
- GNU/Linux
- Mac OS X (42,0МБ)
- FreeBSD [en]
- Mac OS X PPC [en]
- Для других операционных систем [en]

SeaMonkey

SeaMonkey — имя нового проекта, основанного на добрых традициях Mozilla Suite. Классическая Mozilla развивалась много лет и обрела свой круг преданных фанатов.

За основу SeaMonkey взят Mozilla Suite, как проверенный годами продукт, обладающий колоссальной стабильностью и выверенным кодом. К этой основе было добавлено множество функций, ставших привычными для пользователей Mozilla Firefox и Mozilla Thunderbird. В состав SeaMonkey входит несколько компонентов.

Portable

Где бы вы ни находились, вы всегда можете иметь при себе именно ваш SeaMonkey, с вашими закладками, почтой, настройками и расширениями. Для этого вам нужно всего лишь установить на ваш флеш-диск специальную версию SeaMonkey для переносных носителей, и SeaMonkey всегда будет с вами!

Portable SeaMonkey 2.17.1

- Windows (Русская версия, 20,8МБ)

MozBackup

С помощью MozBackup вы можете сделать резервную копию данных профиля SeaMonkey или восстановить данные из резервной копии профиля SeaMonkey. Вы можете сохранять и восстанавливать такие данные как настройки, почту, адресные книги, закладки, историю, расширения, пароли, куки, сохраненные формы, список загрузок, сертификаты и стили

Возможности SeaMonkey:

Навигатор

поддерживает просмотр во вкладках, блокировку всплывающих окон, имеет встроенные инструменты управления загрузкой изображений, осуществляет поиск информации по мере набора текста.

Почта и конференции

Загружаем на компьютер Windows(инсталлятор)

Запустив установку из папки загрузок, устанавливаем программу.

После установки запускаем.

SeaMonkey | Mozilla Россия - SeaMonkey

Файл Правка Вид Перейти Закладки Инструменты Окно Справка

Назад Вперед Обновить Остановить http://mozilla.org

Домой Закладки Часто посещаемые SeaMonkey

Наши проекты: Bugzilla MozUtil Nvu Pastebin Планета

mozilla россия team

поиск по сайту: Поиск

Продукты Поддержка Форум Документация Ссылки О проекте

Главная
Продукты
Mozilla Firefox
Mozilla Thunderbird
Bugzilla
ChatZilla

SeaMonkey

Русские сборки

- Windows (инсталлятор) (21,8МБ)
- Windows (zip-архив) (27,4МБ)
- GNU/Linux
- Mac OS X (42,0МБ)
- FreeBSD [en]
- Mac OS X PPC [en]
- Для других операционных систем [en]

SeaMonkey

SeaMonkey — имя нового проекта, основанного на добрых традициях Mozilla Suite. Классическая Mozilla развивалась много лет и обрела свой круг преданных фанатов.

За основу SeaMonkey взят Mozilla Suite, как проверенный годами продукт, обладающий колоссальной стабильностью и выверенным кодом. К этой основе было добавлено множество функций, ставших привычными для пользователей Mozilla Firefox и Mozilla Thunderbird. В состав SeaMonkey входит несколько компонентов.

Portable

Где бы вы ни находились, вы всегда можете иметь при себе именно ваш SeaMonkey, с вашими закладками, почтой, настройками и расширениями. Для этого вам нужно всего лишь установить на ваш флеш-диск специальную версию SeaMonkey для переносных носителей, и SeaMonkey всегда будет с вами!

Portable SeaMonkey 2.17.1

- Windows (Русская версия, 20,8МБ)

MozBackup

С помощью MozBackup вы можете сделать резервную копию данных профиля SeaMonkey или восстановить данные из резервной копии профиля SeaMonkey. Вы можете сохранять и восстанавливать такие данные как настройки, почту, адресные книги, закладки, историю, расширения, пароли, куки, сохраненные формы, список загрузок, сертификаты и стили

Возможности SeaMonkey:

Навигатор

поддерживает просмотр во вкладках, блокировку всплывающих окон, имеет встроенные инструменты управления загрузкой изображений, осуществляет поиск информации по мере набора текста.

Мы видим, что программа открывается в виде Интернет-браузера и по умолчанию открывает сайт, откуда мы ее скачали. Кстати, вот вам еще один нормальный браузер!

Но в данный момент нас интересует компоновщик, собственно редактор HTML кода.

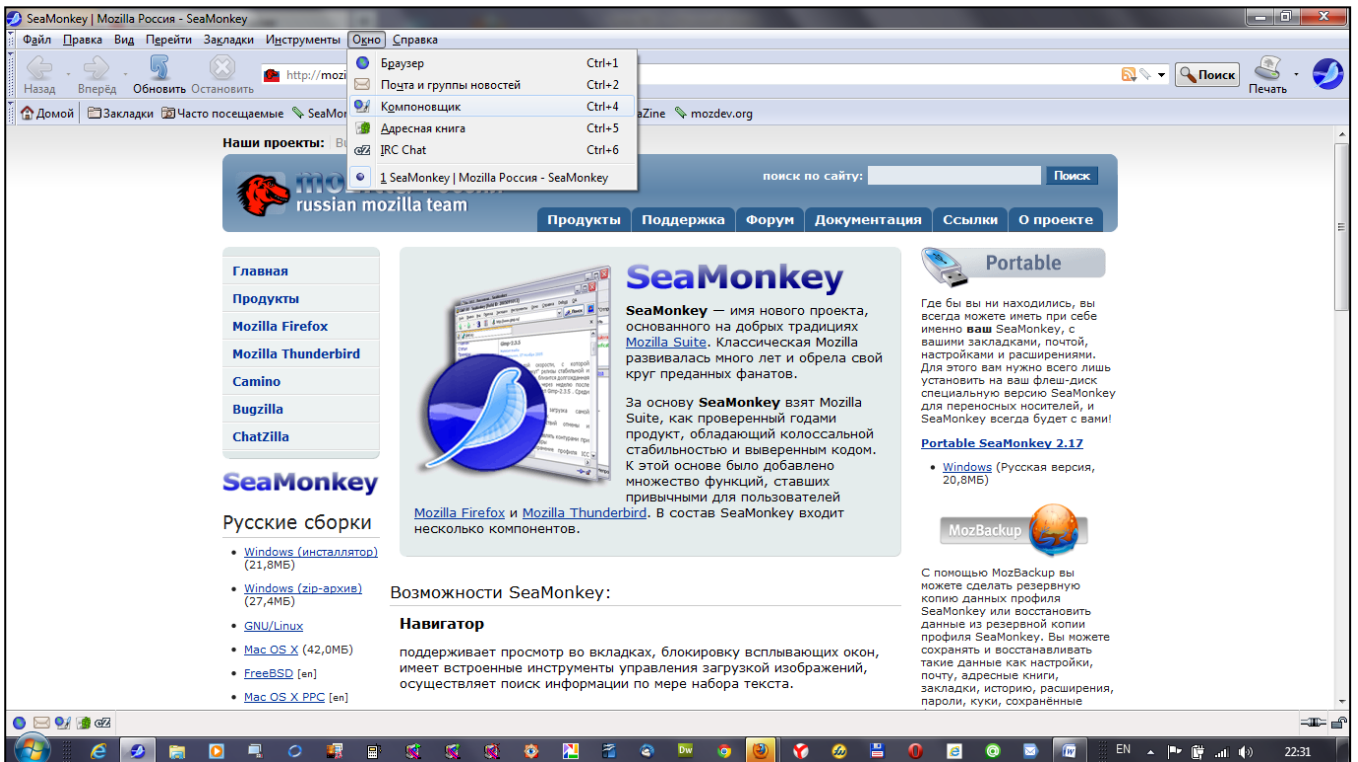
Мы можем его запустить двумя способами: через меню Файл

Файл->Создать->Страницу компоновщика Ctrl+Shift+N

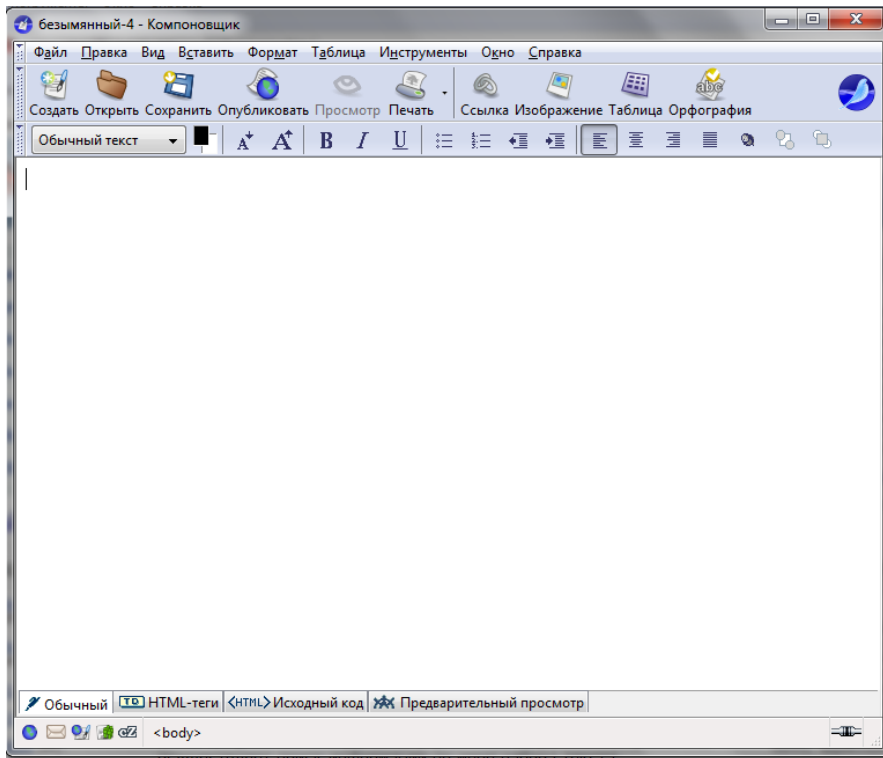


или через меню Окно.

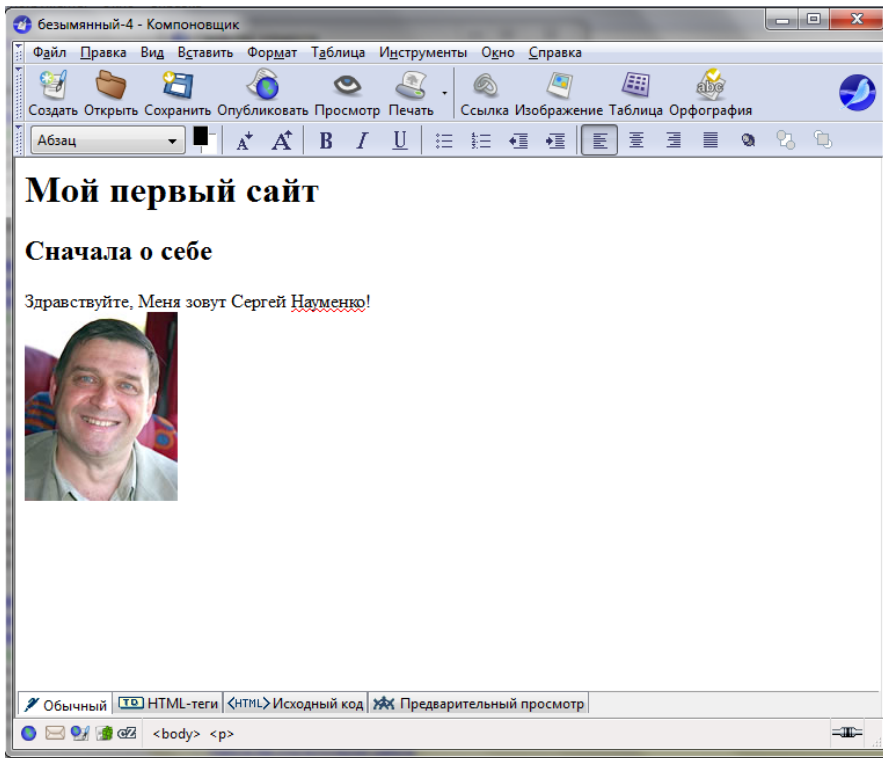
Окно->Компоновщик



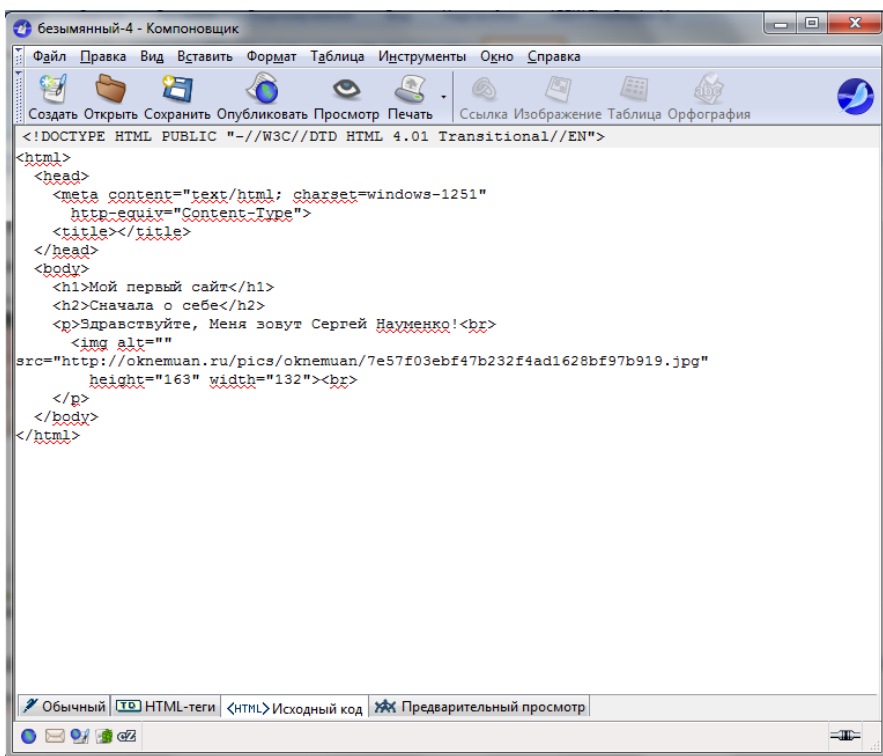
В любом варианте у нас откроется окно компоновщика.



Главная прелесть заключается в том, что мы можем прямо в этом окне начинать создавать WEB-страницу в соответствии с нашими задачами.



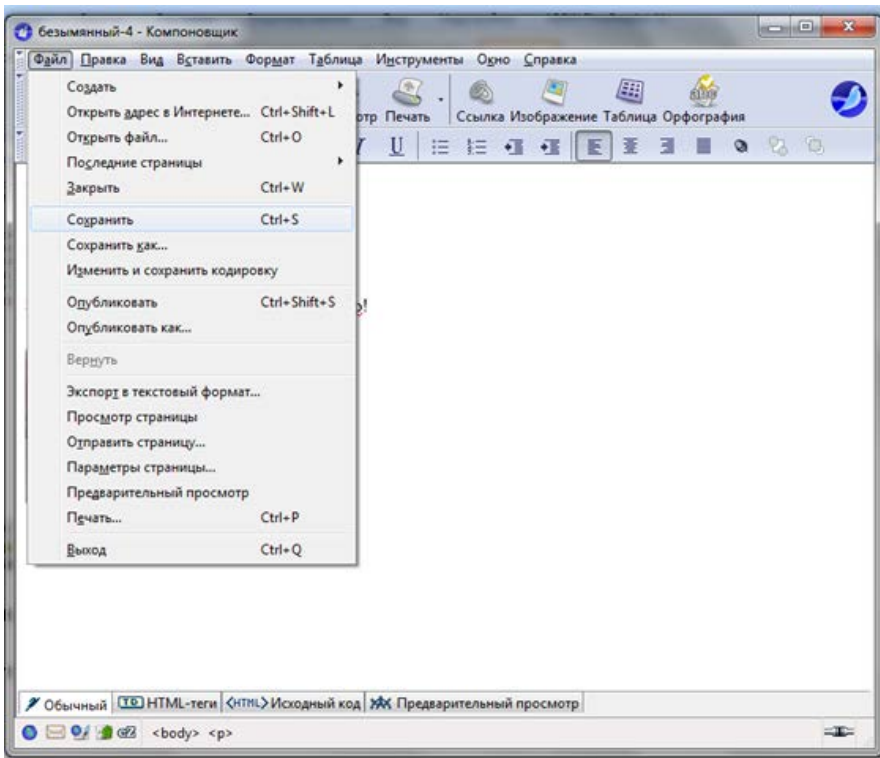
И о, Чудо! Мы можем увидеть, что HTML код пишется сам! Для этого нам просто надо внизу открыть вкладочку <HTML> Исходный код



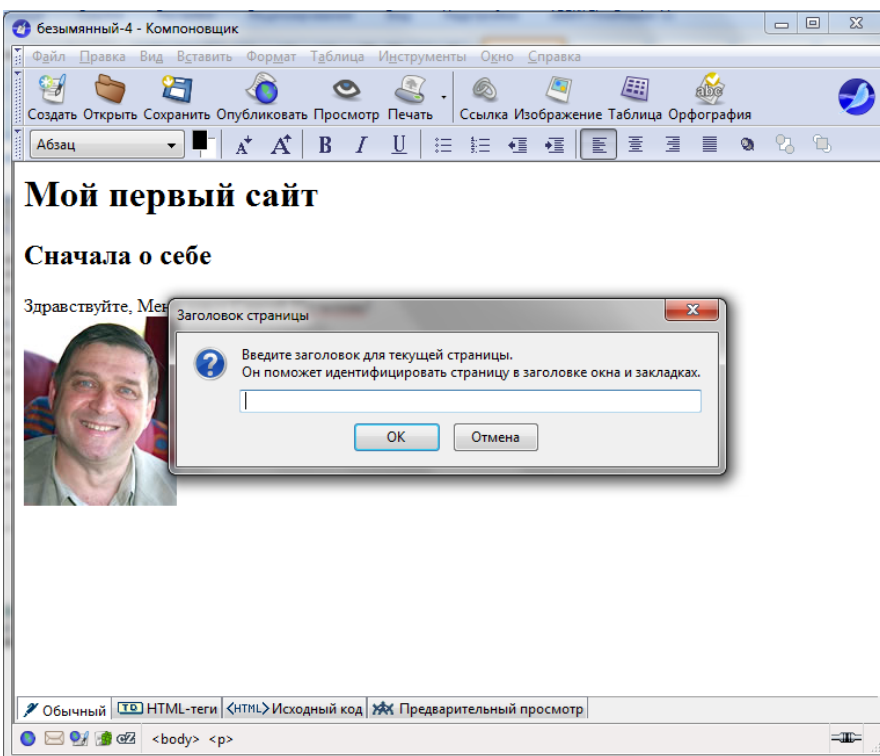
И этот код я также могу редактировать.

После того, как я создам страницу, я могу ее сохранить в папочке сайта www

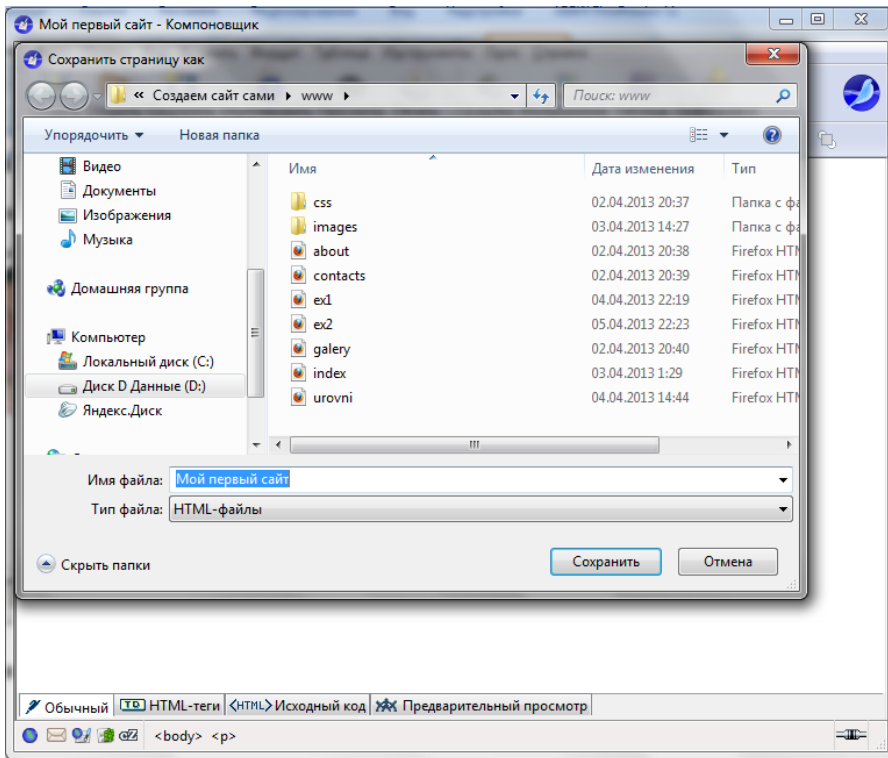
Файл->Сохранить



После того, как я нажму Сохранить, Редактор Компоновщик предлагает создать заголовок страницы, то есть заполнить пространство между тегами<title></title>



Вбиваем туда заголовок и ждем ОК. Открывается окно сохранения странице, где вы должны найти место для сохранения страницы



По умолчанию задается имя, которое вы указали в заголовке страницы, его надо изменить и написать латинскими буквами, после чего нажать кнопку Сохранить.

Файл сохранится с расширением html.

Очень удобный редактор для HTML страниц. Недостаток – возможность работы только с HTML и текстовыми файлами. Для серверного программирования непригоден. Можно использовать только для верстки статичных HTML страниц и макетов.

NVU

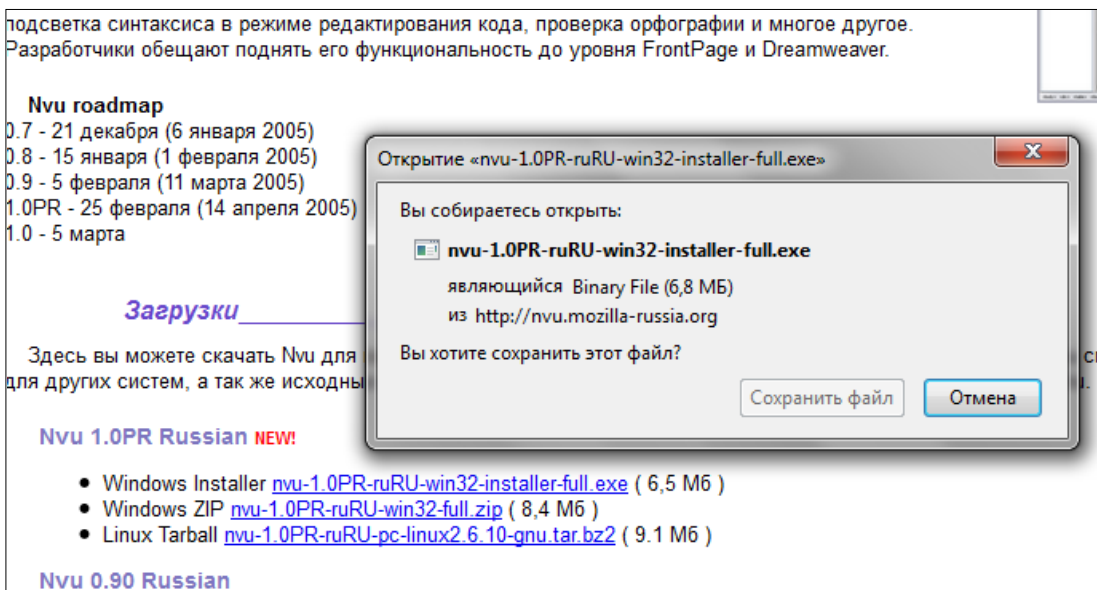
Другой бесплатный редактор HTML кода –NVU

Заходим по ссылке на сайт

<http://nvu.mozilla-russia.org/index2.php>

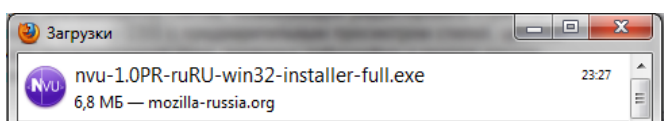


Скачиваем ссылку с установщиком для Windows в разделе Загрузки.

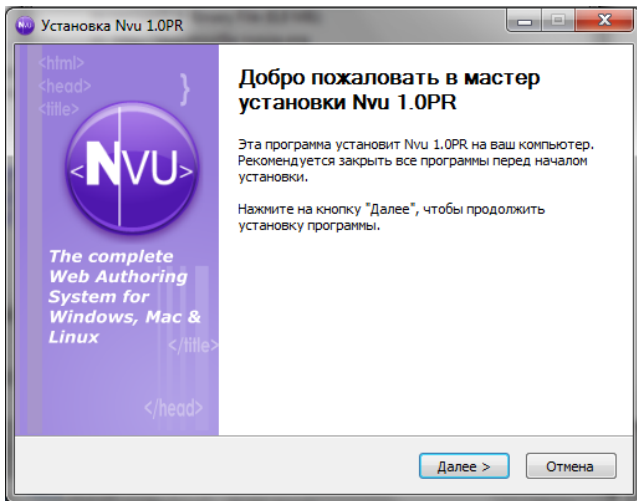


Выбираем сохранить файл.

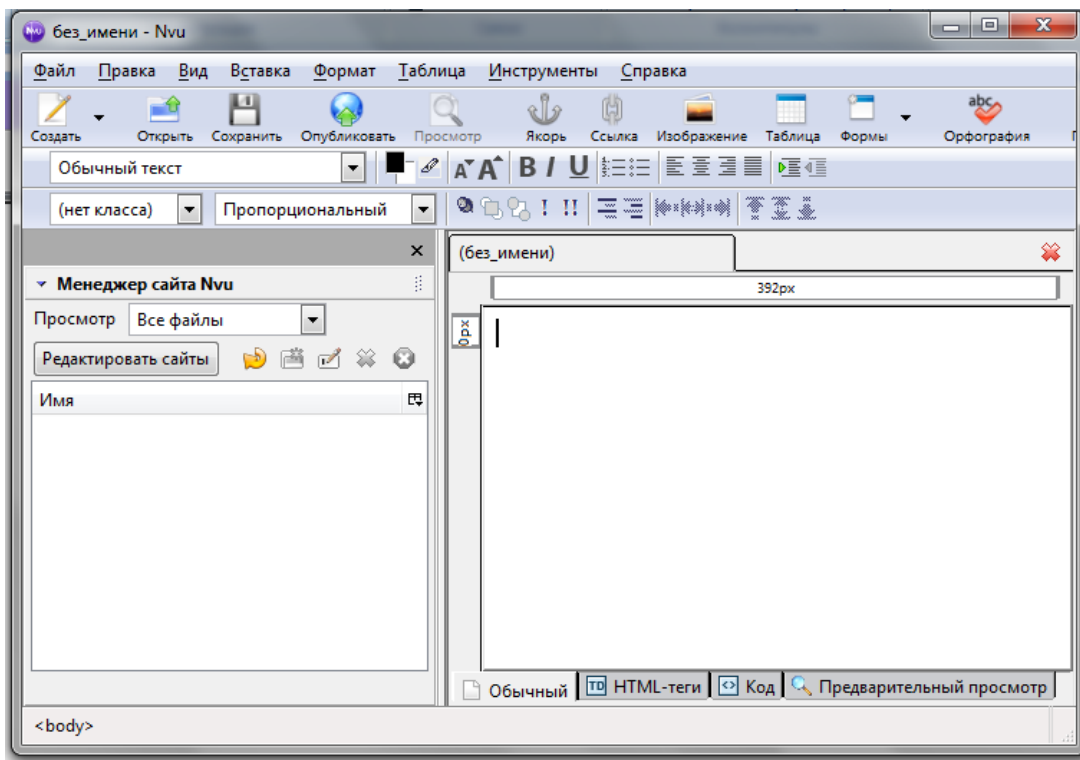
После чего запускаем сохраненный установочный файл.



И выполняем установку, нажав кнопку «Далее», соглашаясь со всеми условиями, которые будут предложены.



После установки запускаем программу, откроется окно редактора

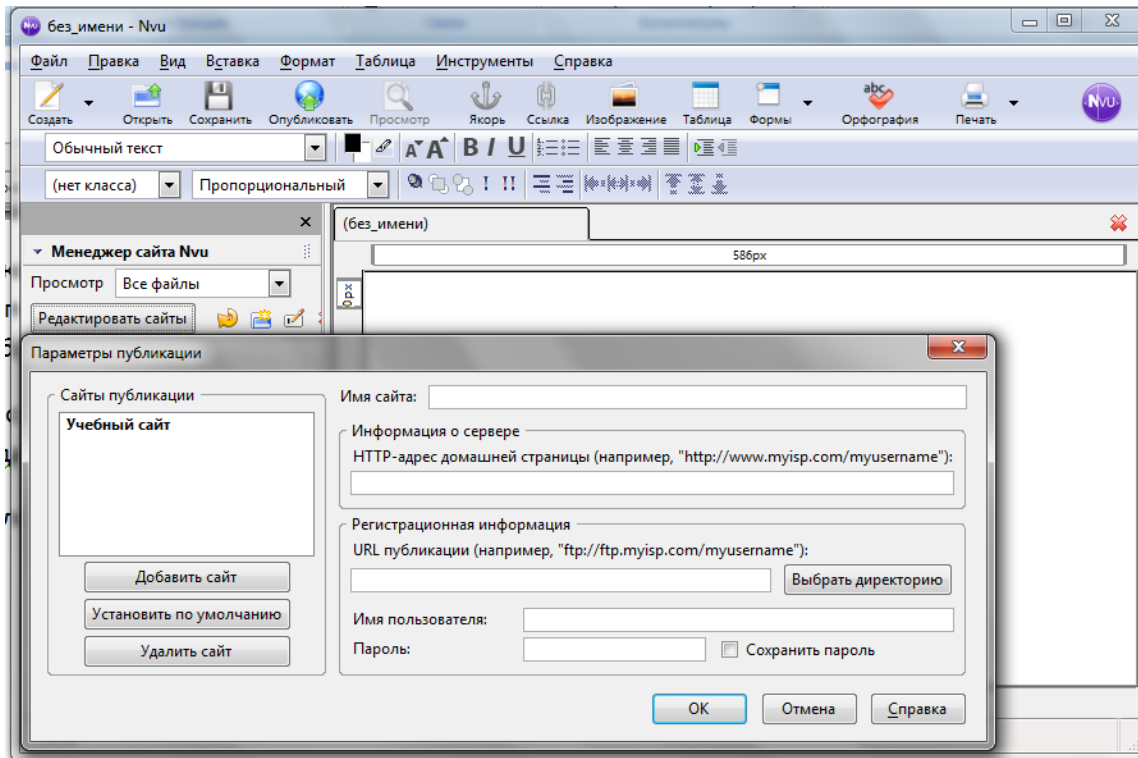


Так как эти продукты созданы одной компанией, то работа с этой программой аналогична работе с компоновщиком. Страницы можно создавать в окне справа, редактировать, вставлять изображения и другие элементы и потом сохранять их в папке сайта.

Присутствующий здесь в левом окне менеджер сайтов позволяет отображать папки и файлы создаваемого сайта.

Сделать это можно следующим образом:

Нажать кнопку «Редактировать сайты» И в раскрывшемся диалоговом окне указать имя сайта, например «Учебный сайт» и в окошке «регистрационная информация» указать путь к папке WWW где хранятся файлы сайта помощью кнопки Выбрать директорию.



Нажать ОК.

В окошке отобразится файловая система сайта. Очень удобная вещь.

Dreamweaver

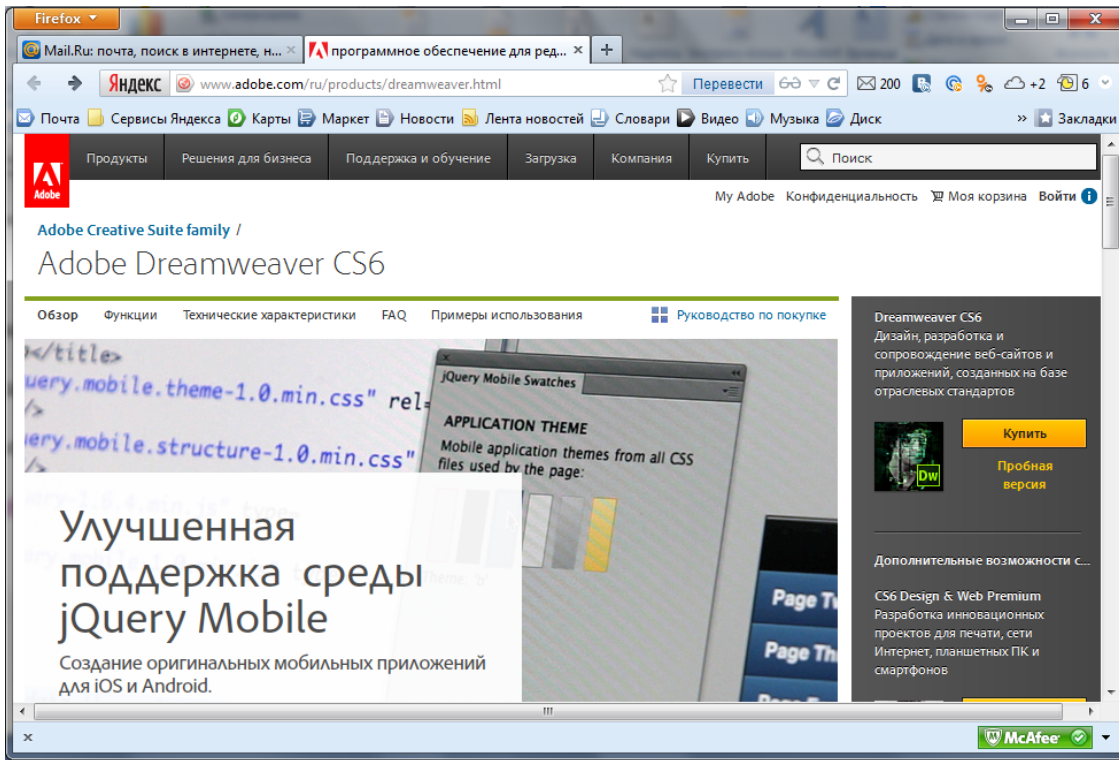
Ну, и наконец, самый крутой редактор всех времен и народов – Это Dreamweaver.

Заходим по ссылке

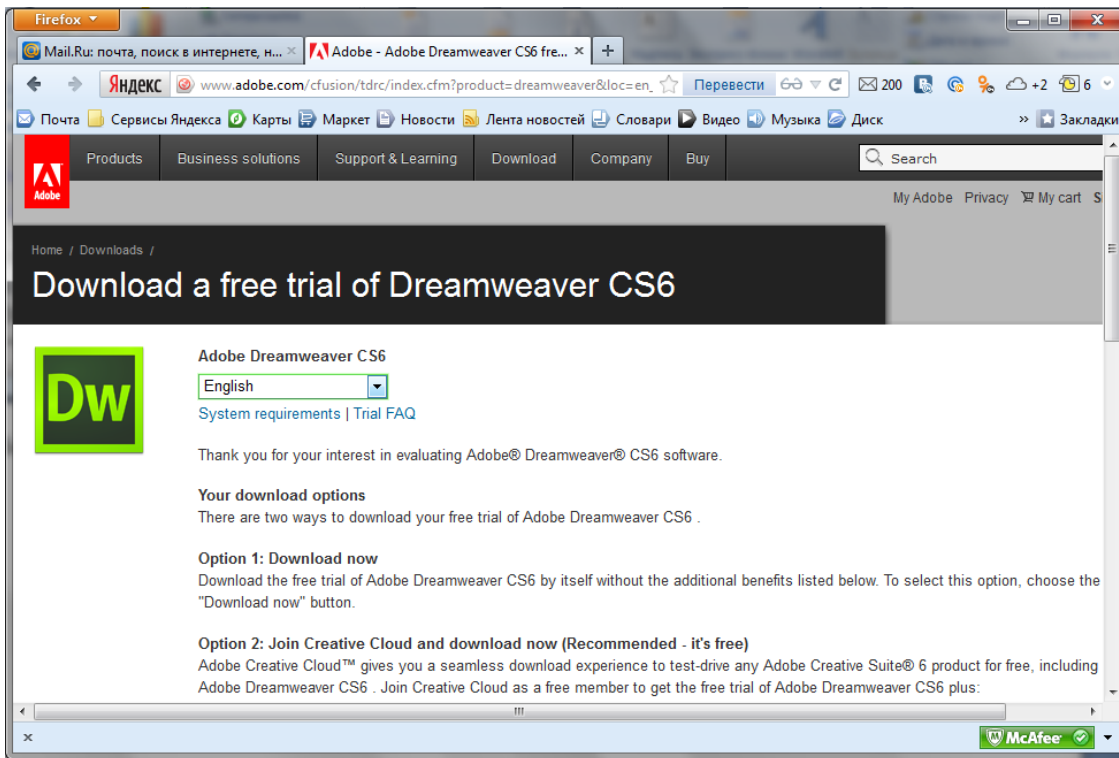
<http://www.adobe.com/ru/products/dreamweaver.html>

Установочная программа может запросить ID Adobe(Адрес электронной почты) и пароль учетной записи Adobe. Если у вас нет ID Adobe, то необходимо зарегистрироваться на официальном сайте <http://adobe.com> ввести свои персональные данные, это адрес электронной почты, который и будет вашим ID Adobe и пароль, подтвердить регистрацию, кликнув на ссылку в пришедшем от Adobe.com письме.

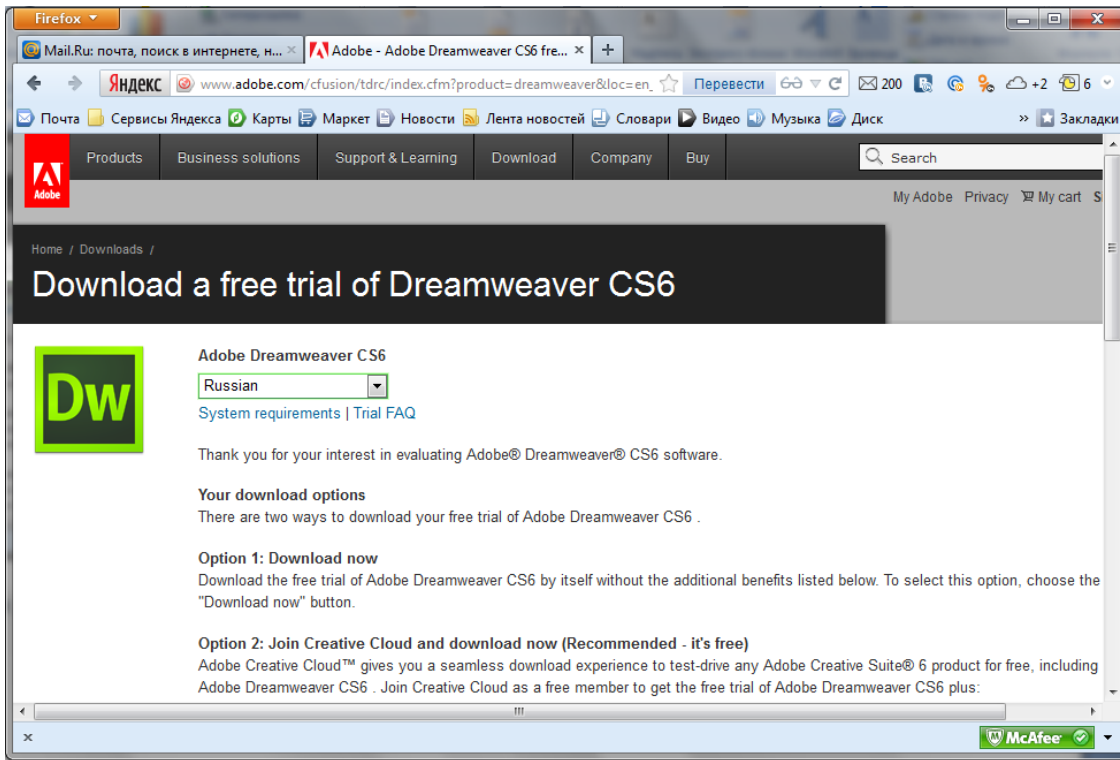
В дальнейшем применяйте ID Adobe для покупки и обновления товаров Adobe.



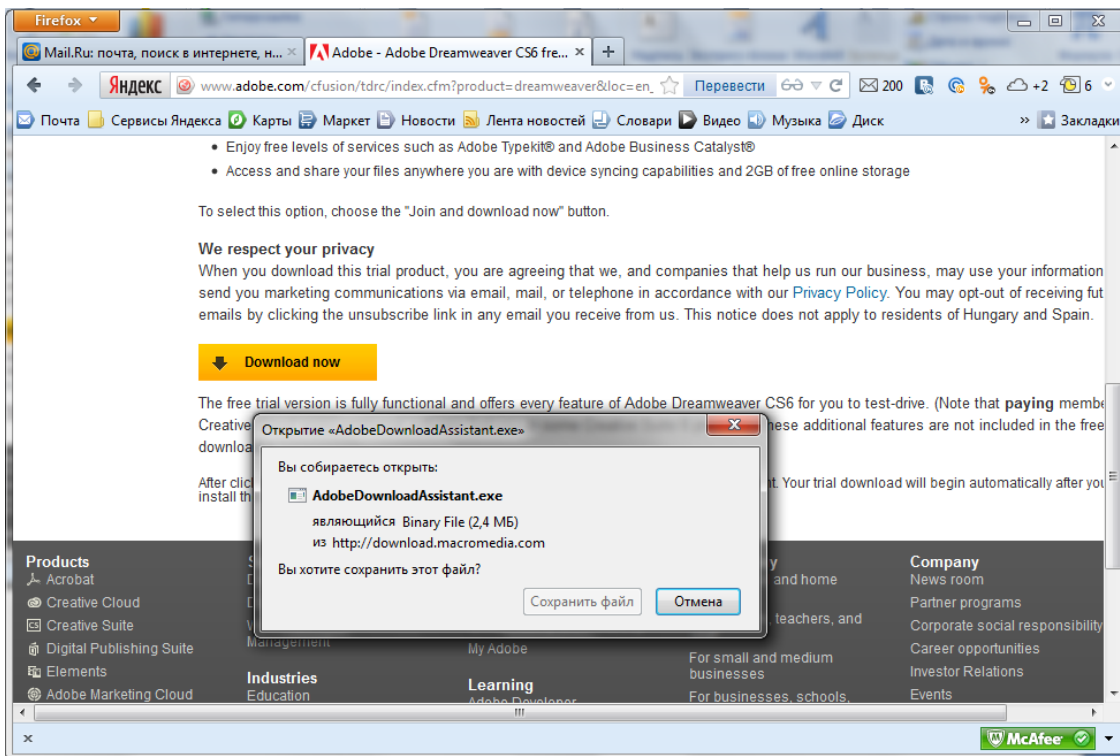
Выбираем пробную версию



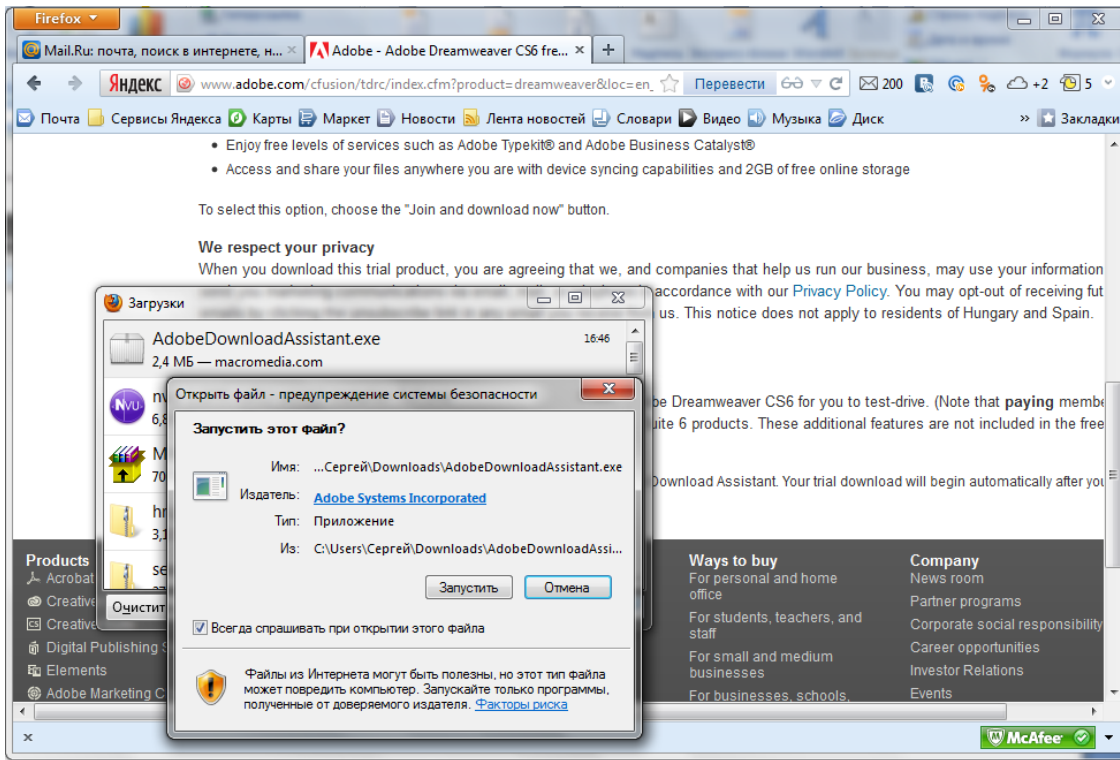
Выбираем язык



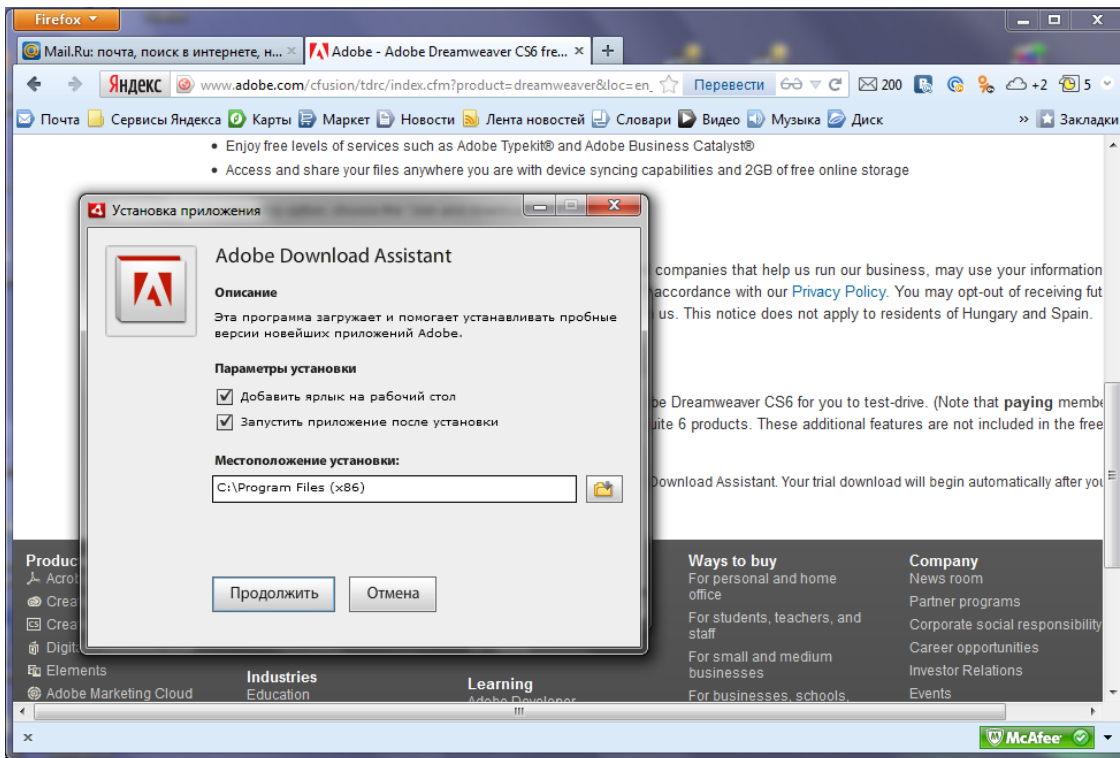
Прокручиваем экран вниз до появления кнопки загрузки нажимаем ее и скачиваем установочный файл нажатием на кнопку «Сохранить файл».



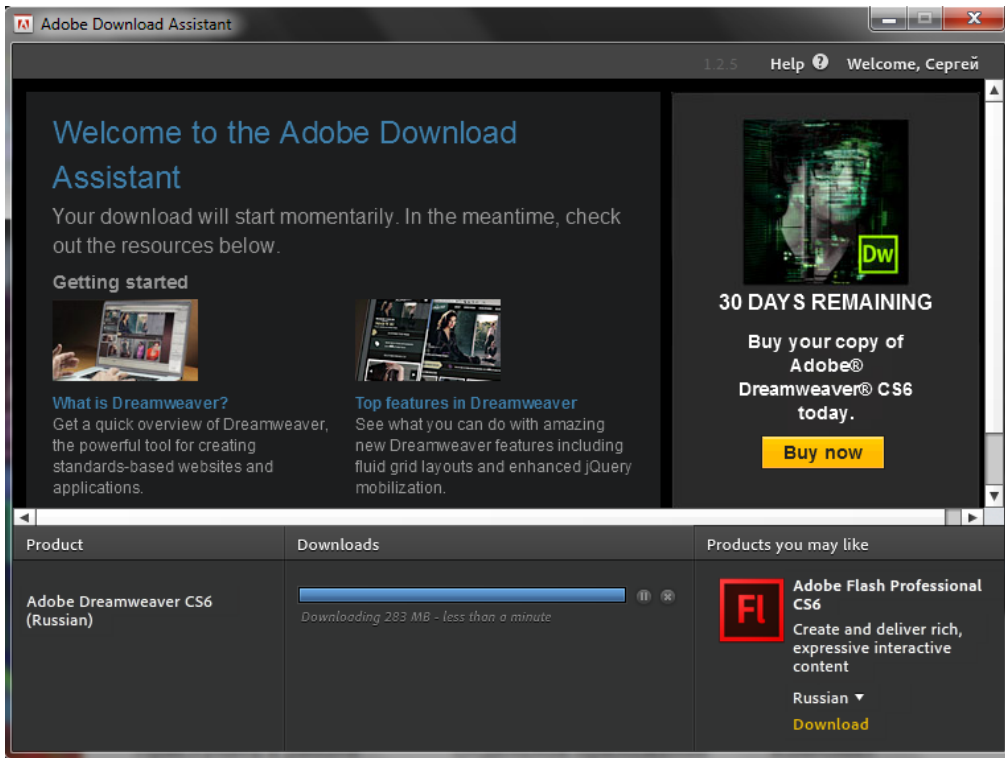
После загрузки запускаем установочный файл.



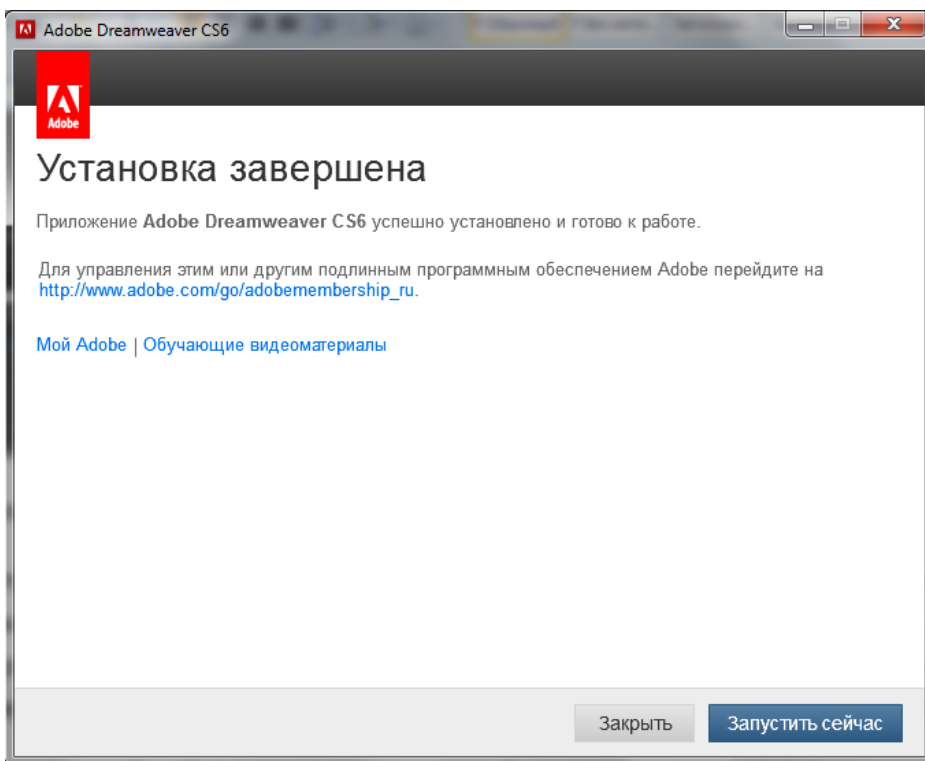
После запуска открывается окно загрузки



Нажимаем кнопку продолжить

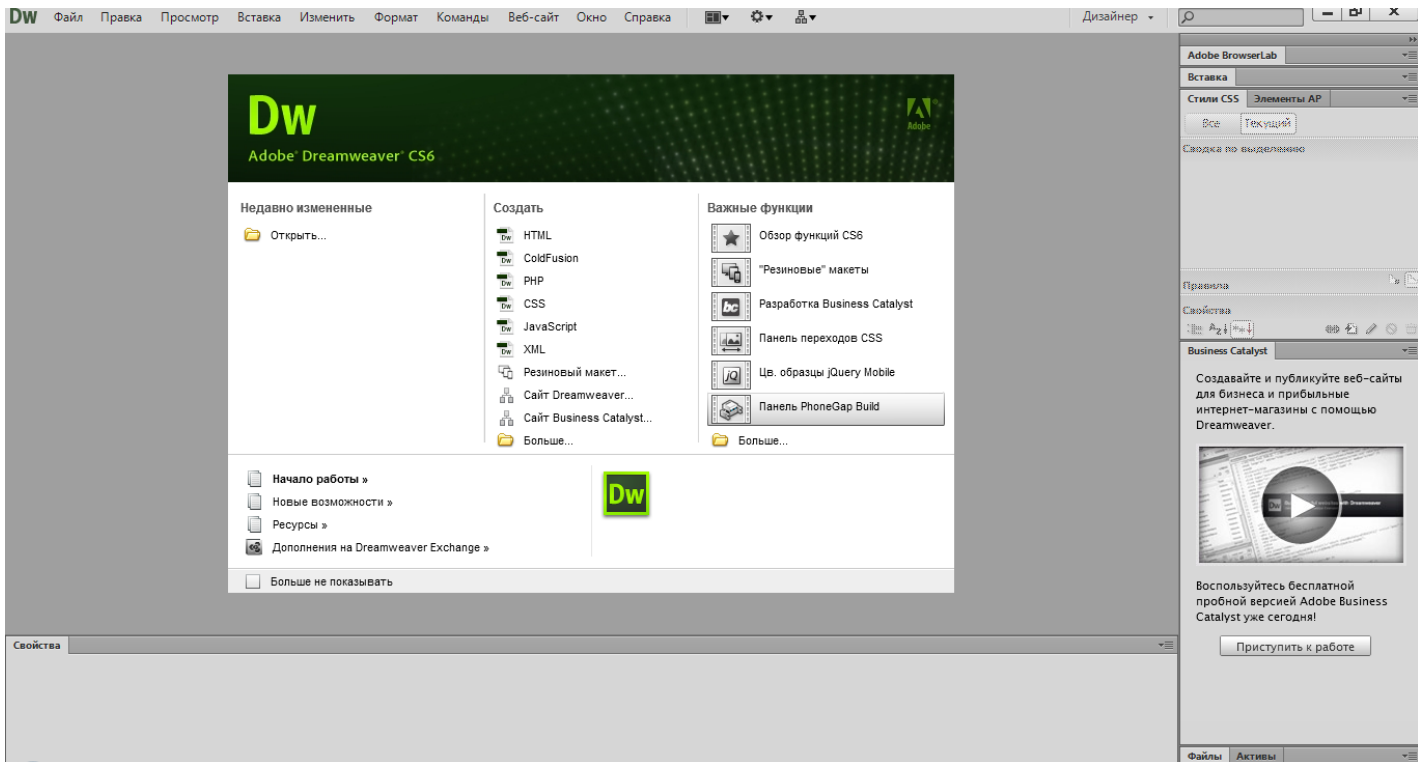


Начнется загрузка и установка файлов на компьютер.



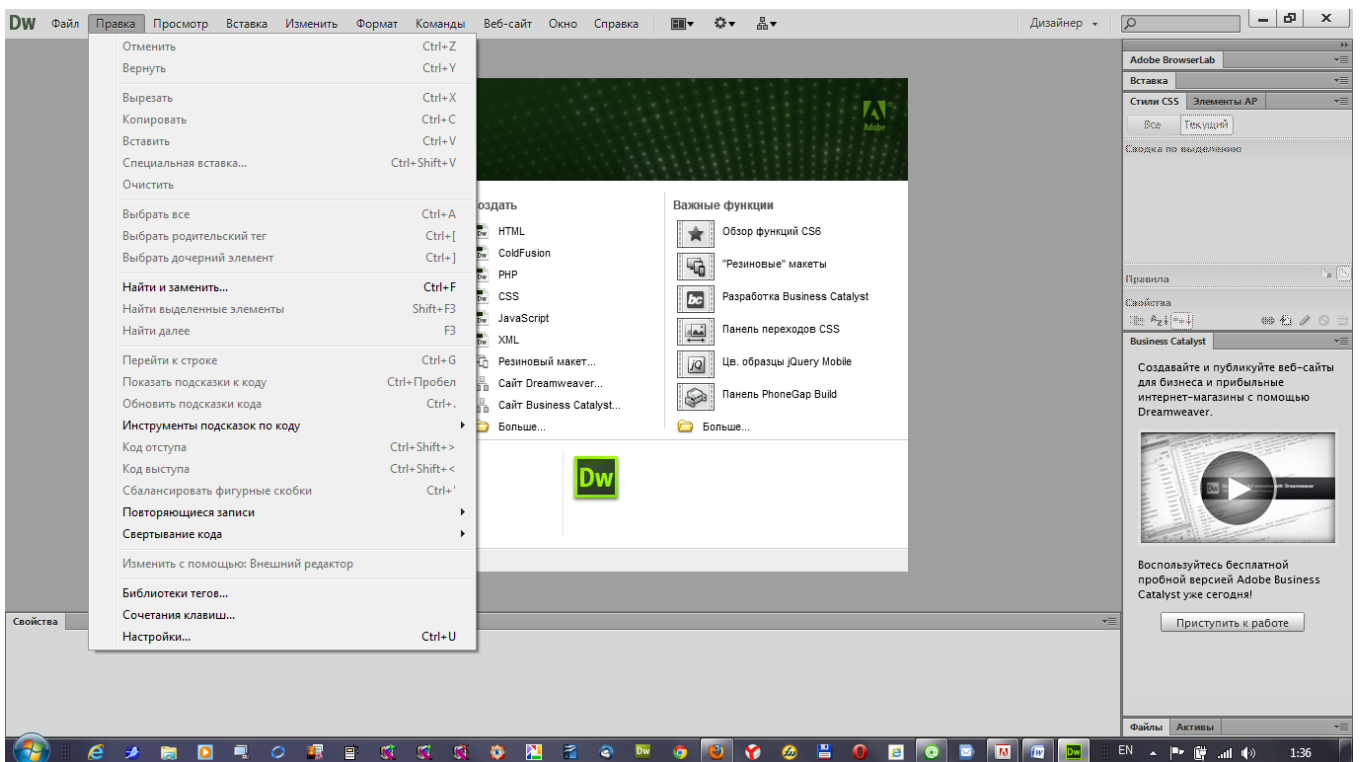
После установки нажимаем кнопку Запустить Сейчас.

Откроется окно работы с программой.

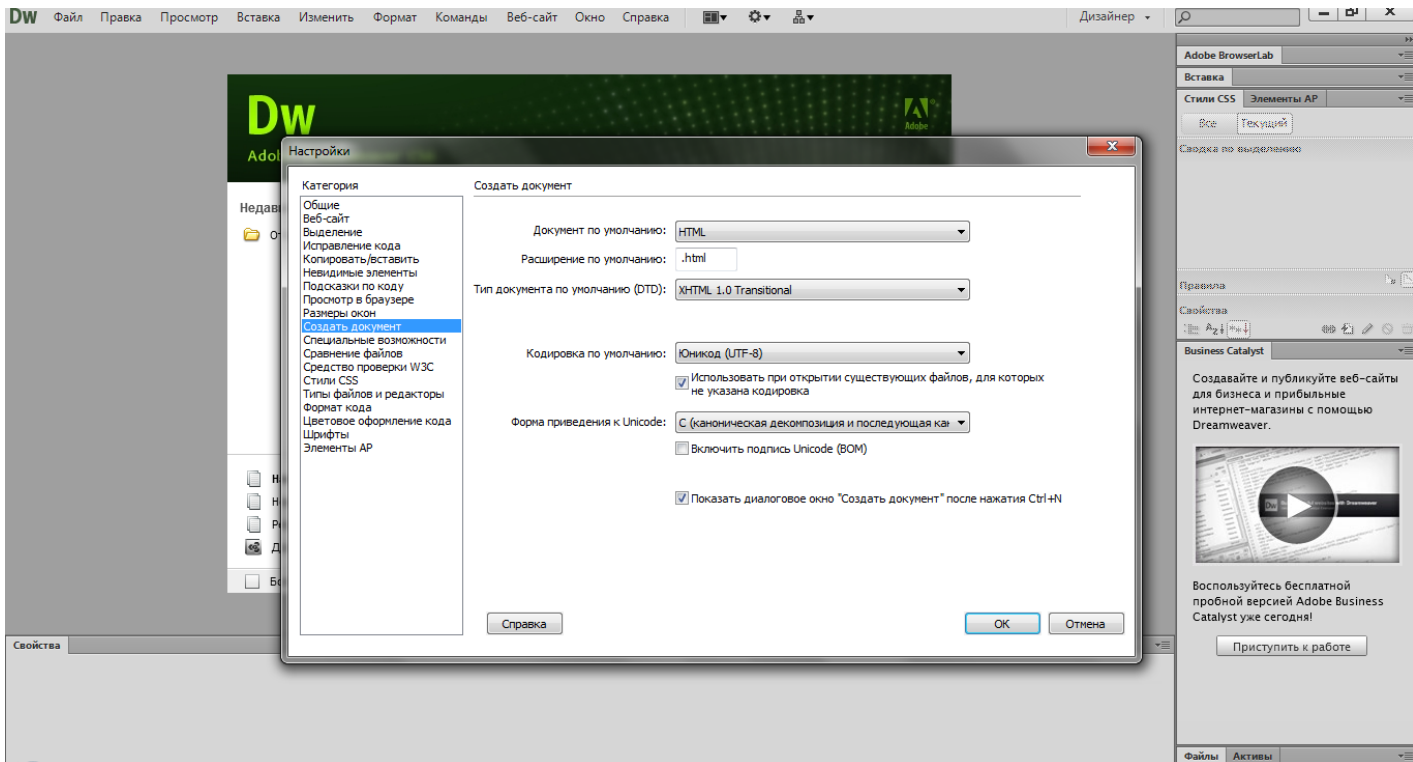


Первое, что необходимо сделать, это установить настройки кодировки кириллицы.

Открываем Правка->Настройки

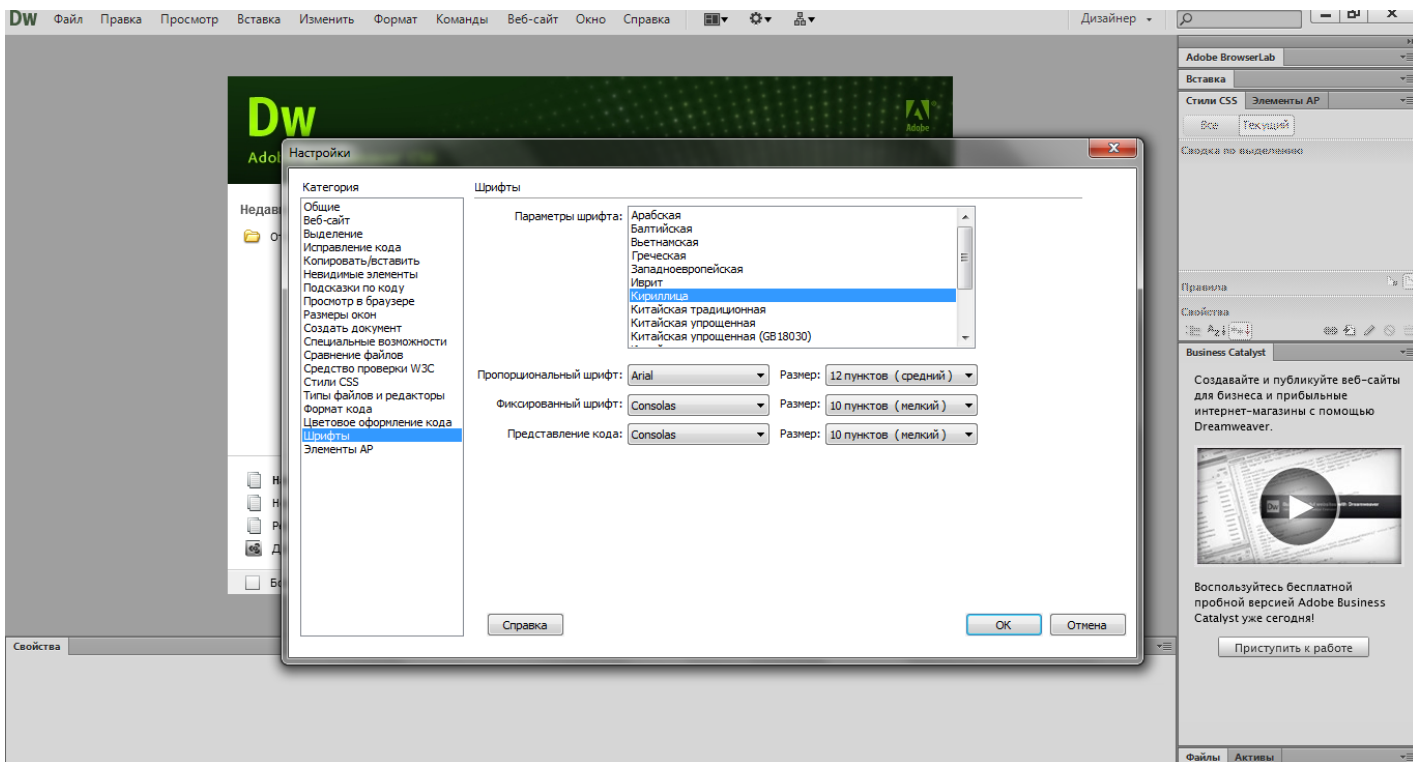


В настройках выбираем «Создать документ»



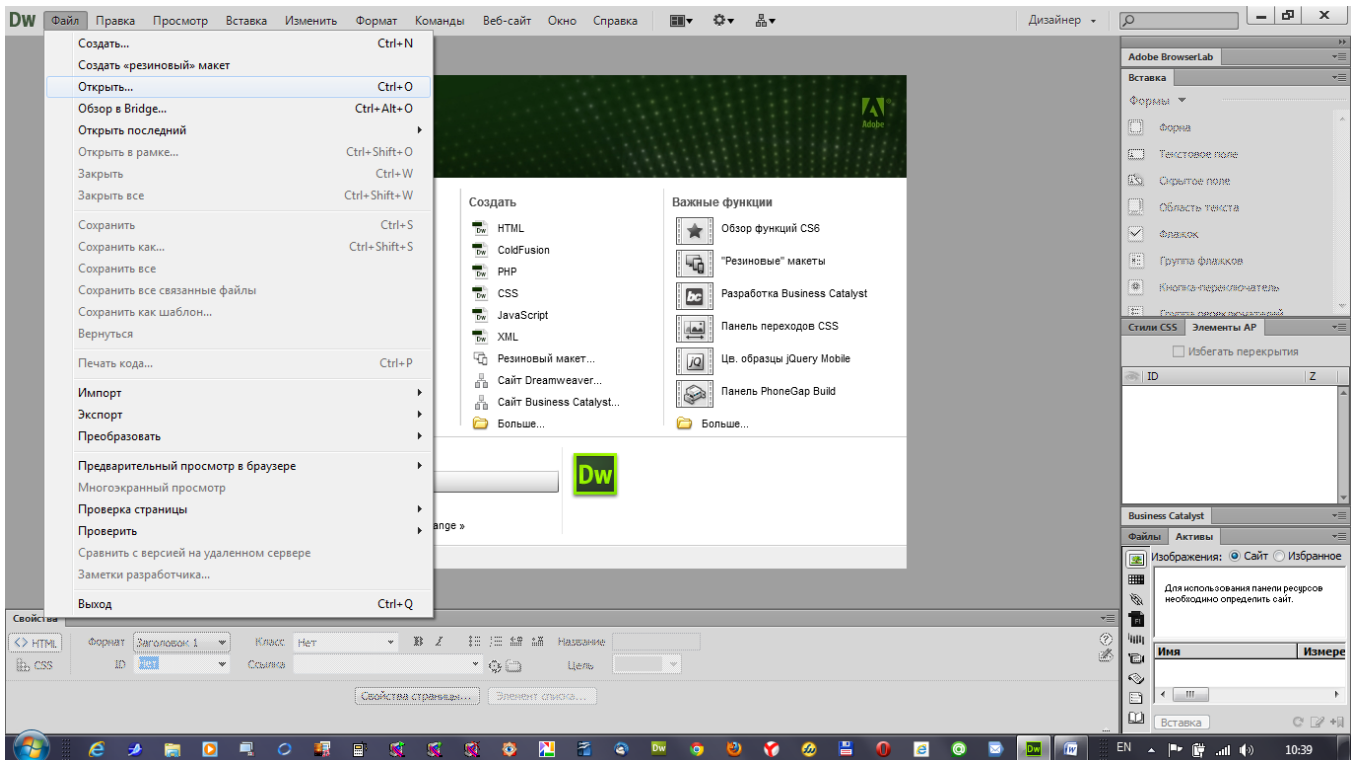
В списке Кодировка по умолчанию вместо «Юникод (UTF-8)» устанавливаем «Кириллица (Windows)» и нажимаем ОК.

Снова вызываем Окно настроек, а в нем выбираем «Шрифты».

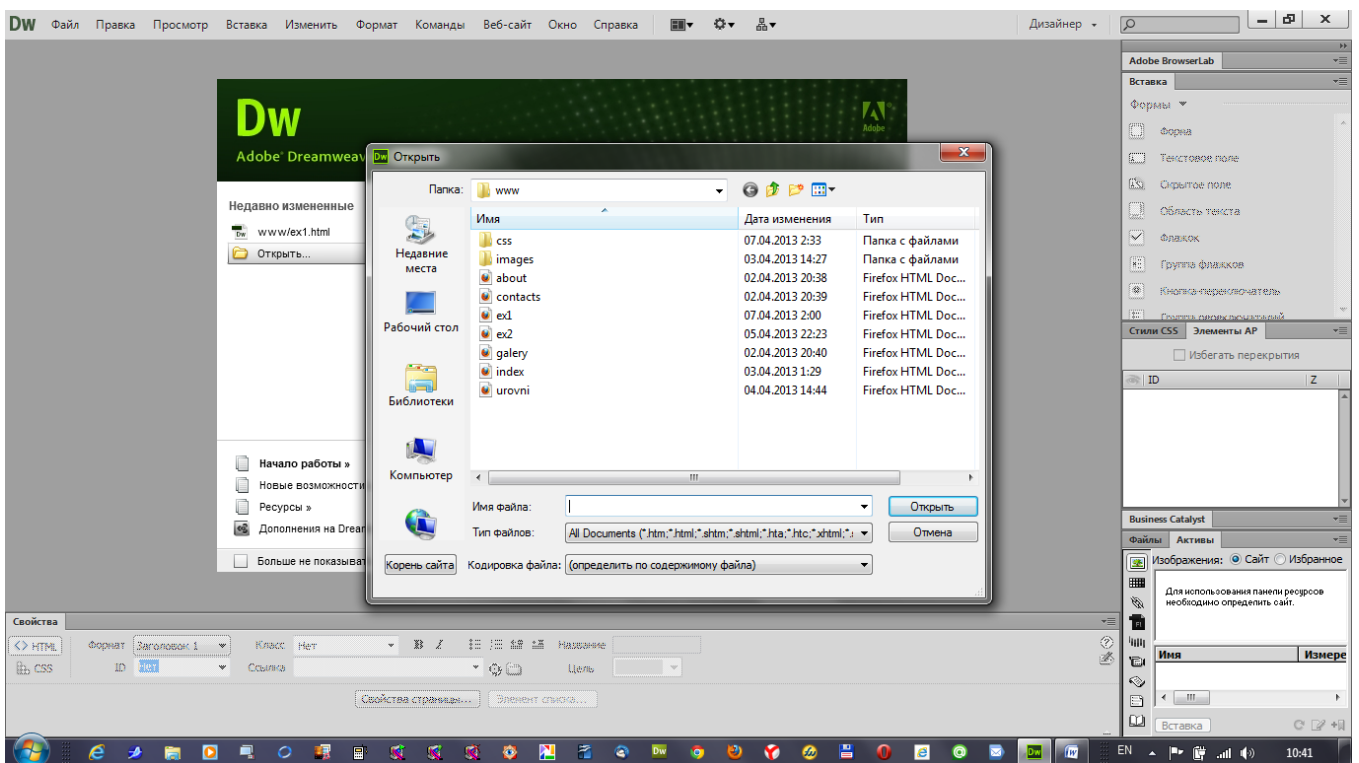


Параметры шрифта устанавливаем «Кириллица», нажимаем ОК.

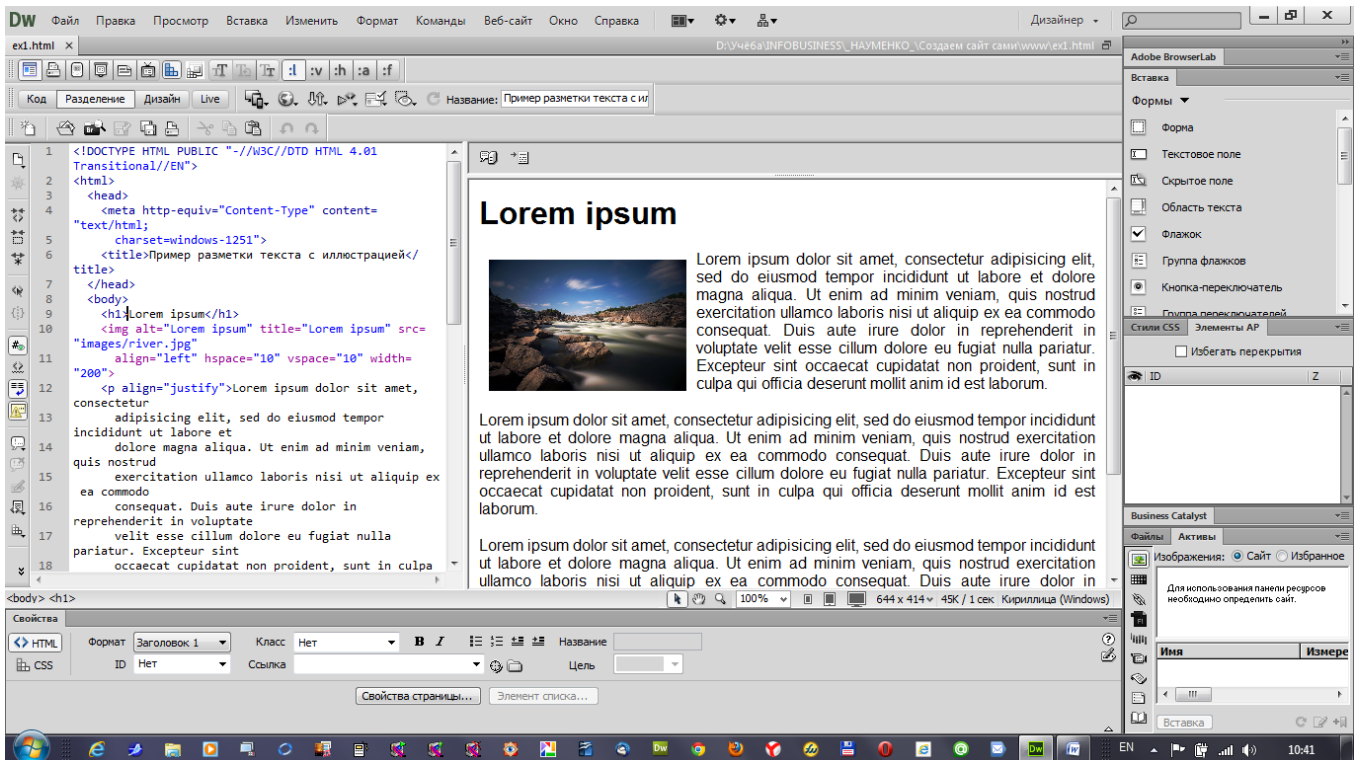
Теперь можно открыть наш файл. Выбираем Файл->Открыть



Находим на компьютере наш файл



Нажимаем кнопку Открыть



Все. Теперь можно работать с файлом в режиме кода и в режиме дизайна.

Обучение работе в DREAMWIEVER выходит за рамки наших занятий. Программа довольно сложная и ее изучение достойно отдельного курса.

Сегодняшний урок должен был расширить Ваш кругозор относительно возможностей использования редакторов HTML для облегчения работы с HTML кодом.

Очень Важно понять, что наличие такого помощника в работе с HTML кодом не заменят вам знаний HTML.

Многие новички меня спрашивают, а зачем нам изучать HTML, если есть такие программы, которые сами его пишут.

Ответ один. Если вы хотите стать профессионалом, вы должны держать весь код под своим контролем.

Редактор HTML лишь помогает вам в этом.

Урок № 4

Гиперссылки

Мы продолжаем изучать HTML и сегодня я научу вас делать ссылки, писать комментарии, рисовать прямые линии, к HTML коду, изучим новые теги `<div>` и `
` а также специальный символ неразрывного пробела ` `;

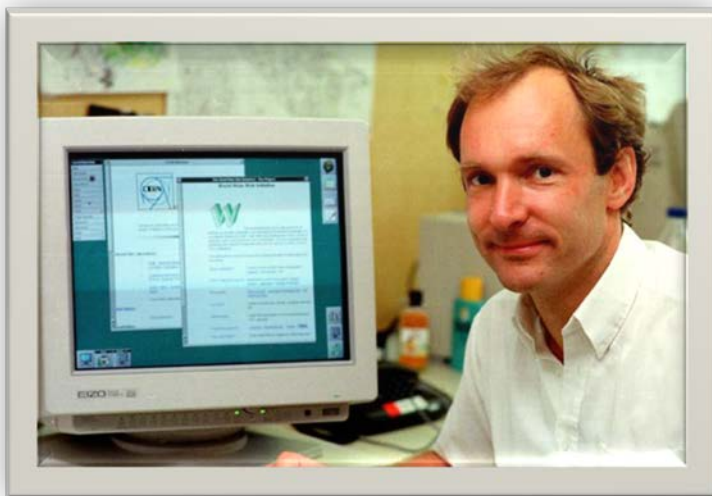
В конце урока вас как всегда ждет задание для закрепления материала.

Я надеюсь, что под впечатлением предыдущего урока вы установили себе редактор HTML. Он нам поможет в изучении HTML кода.

Я установил себе 30 дневную версию Adobe Dreamewiever CS6.

Интернет не был бы интернетом, если бы не умел связывать между собой WEB-страницы. Для перехода с одних WEB-страниц на другие и были придуманы гиперссылки.

Все это придумал изобретатель и создатель современного интернета Сэр Уильям Тимоти Джон Бернерс-Ли.



Именно он, в 1989 году, работая в CERN придумал и предложил воплотить глобальный проект, под названием World Wide Web. Что в переводе означает Всемирная паутина. Этот проект подразумевал публикацию WEB-страниц, связанных между собой гиперссылками. Кстати HTML тоже придумал Тим Бернерс-Ли. Он также придумал протокол HTTP, первый WEB-Сервер, который назывался httpd, первый браузер, который назывался WorldWideWeb, который в 1991 году начал бесплатно распространяться под названием Nexus.

Итак, что же такое гиперссылка или как ее называют в быту – просто ссылка?

Гиперссылка это текст или изображение, кликнув по которому, осуществляется переход либо на другую WEB-страницу, либо какое-нибудь место на той же самой WEB-странице.

Вы конечно же знаете, как выглядит текстовая ссылка.

Например вот так:

[Поисковая система Яндекс](#)

Или так:



Нам главное разобраться, как она устроена изнутри.

Для организации ссылок существует специальный тег HTML.

Это :

```
<a></a> .
```

Происходит он от английского слова anchor, что в переводе означает Якорь.

Для перехода на внешний документ, будь это WEB страница или любой другой файл, ссылку можно записать следующим образом:

```
<a href="contacts.html">Наши контакты</a>
```

Как вы видите, у тега `<a>`, для перехода существует атрибут `href`, что происходит от понятия **Hyper REFerence**- гиперссылка.

Параметром атрибута `href` является путь к документу, на который нужно осуществить переход. В данном примере переход нужно осуществить к веб-странице под именем `contacts.html`.

Между открывающим `<a>` и закрывающим тегом `` записывают, в нашем примере, текст, который будет отображаться как ссылка, то есть синего цвета с подчеркиванием. В нашем случае это [Наши контакты](#).

Во всех браузерах по умолчанию текстовые ссылки отображаются именно так: [синие и подчеркнутые](#).

Как я уже говорил, вместо текста можно использовать изображение.

В этом случае ссылку можно записать так:

```
<a href="contacts.html"></a>
```

То есть вместо текста «Наши контакты» мы в качестве ссылки показываем изображение кнопки перехода в раздел Контакты, которое у нас хранится в папке images под именем button_contacts.jpg.

В браузерах Интернет-эксплорер картинка – ссылка отображается в некрасивой синей рамочке, чтобы не показывать синюю некрасивую рамочку, мы в теге img указываем атрибут border равным нулю.

Если ссылка указывает на внешний сайт, то ее адрес должен содержать полный URL адрес, например <http://yandex.ru>

```
<a href="http://yandex.ru">Поисковая система Яндекс</a>
```

Как я уже говорил, ссылки могут быть не только на внешние файлы и сайты, но также и на места в пределах одной страницы.

Для этого необходимо создать те самые места, которые называются Якорями.

<a> - anchor-Якорь



Якорь выглядит следующим образом:

```
<a name="ch_1"></a>
```

Здесь, для обозначения имени якоря используется атрибут name, параметром которого может быть любой текст, в нашем случае ch_1, что может обозначать место на WEB-странице, где начинается глава 1(**chapter1**).

Такая конструкция располагается непосредственно перед заголовком или перед параграфом (абзацем).

Например:

```
<a name="ch_1"></a><h1>Глава 1</h1>
<p>Текст главы 1...</p>
<a name="ch_2"></a><h1>Глава 2</h1>
<p>Текст главы 2...</p>
<a name="ch_3"></a><h1>Глава 3</h1>
<p>Текст главы 3...</p>
```

Пространство между тегами <a> и якоря не заполняется.

Теперь, чтобы сделать ссылку на Главу, можно поступить примерно так:

```
<a href="#ch_3">Глава 3</a>
```

В атрибуте href указывается решетка и имя якоря.

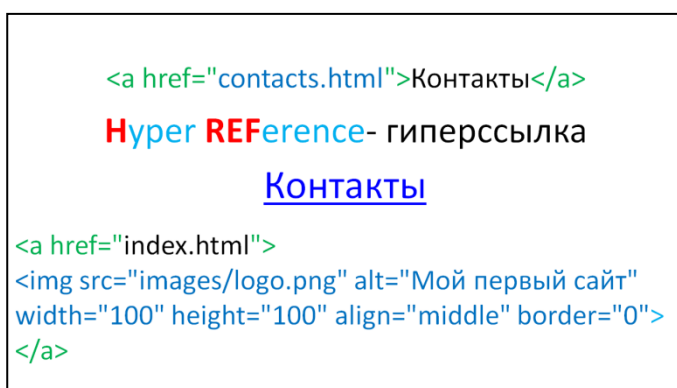
Если текст достаточно объемный, браузер прокрутит страницу до нужного якоря и сделает строку с якорем самой верхней строкой в окне браузера.

Если вы хотите сделать ссылку с одной страницы на якорь, который находится на другой странице, то решетка с именем якоря добавляется в конец адреса страницы.

Например:

```
<a href="book1.html#ch_3">Глава третья книги 1</a>
```

Ссылка переводит нас к странице book1.html и к якорю с именем ch_3.



Дополнительные атрибуты тега <a>

Теперь рассмотрим дополнительные атрибуты тега <a>, которые расширяют возможности ссылок.

Вообще-то у тега <a> 14 атрибутов. Два из них href и name мы с вами рассмотрели. Теперь рассмотрим еще два: это target и title.

атрибут target указывает, в каком окне будет открываться ссылка. Если атрибут target отсутствует, то ссылка будет открыта в текущем окне, что соответствует параметру _self, который задает открытие ссылки в текущем окне по умолчанию.

Например:

```
<a href="http://yandex.ru" target="_self">Яндекс</a>
```

Эта ссылка откроется в текущем окне.

Если атрибуту target присвоить параметр _blank, то ссылка откроется в новом окне.

Например:

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

Эта ссылка откроется в новом окне.

Переходим к рассмотрению атрибута title.

Прошу не путать с тегом <title>!!!

Атрибут title добавляет поясняющий текст к ссылке в виде всплывающей подсказки. Такая подсказка отображается, когда курсор мыши задерживается на ссылке.

Например:

```
<a href="http://yandex.ru" title="Поисковая система Яндекс">Яндекс</a>
```

При наведении курсора на ссылку, отобразится подсказка «Поисковая система Яндекс».

А теперь я покажу, как можно применить только что полученные знания и создать меню со ссылками на страницы для нашего учебного сайта.

Для этого я покажу вам еще пару-тройку тегов и специальных символов.

Тег <hr> отображает прямую линию.

Тег <!-- Комментарий--> Комментарий к коду. В браузере не отображается.

Комментарии нужны веб-дизайнеру, то есть вам, для того, чтобы описывать участки HTML кода.

Вообще писать комментарии к коду это не только правило хорошего тона, но и признак профессионализма. С помощью комментариев легче разобраться вам и тем, кто после вас будет работать с вашим кодом, что и где находится в коде.

Опыт показывает, что без комментариев очень тяжело потом искать нужные участки, особенно когда количество строк превышает сотню-другую.

А комментарий всегда можно найти с помощью поиска по тексту

Итак, ставим перед собой задачу:

Создаем сайт из трех страниц Главная, О нас, Контакты.

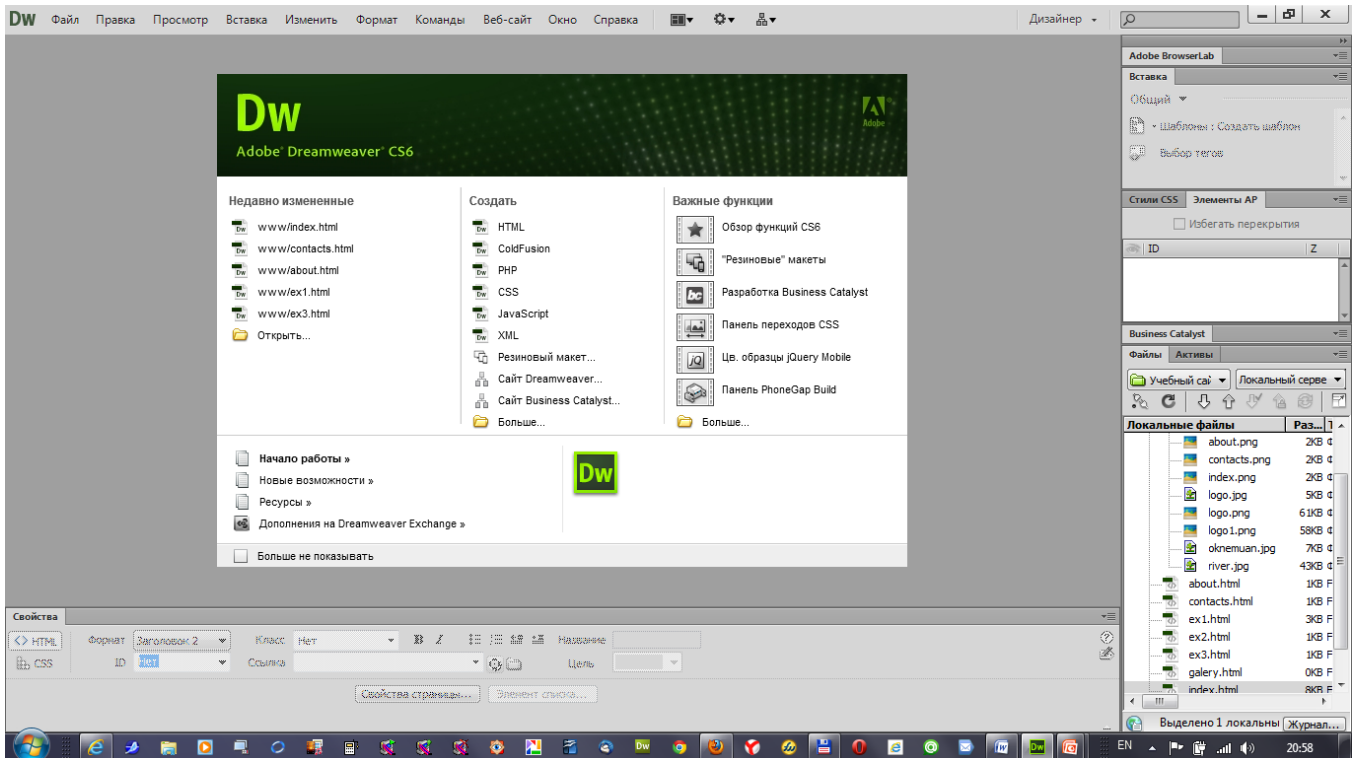
Сайт содержит логотип, заголовок, меню для перехода на другие страницы.

Я тут в графическом редакторе создал простенький логотипчик нашего сайта. Вы можете нарисовать что-нибудь более интересное для вас.



Логотип помещаем в папку images. Будем использовать его для украшения сайта.

Переходим в редактор WEB-страниц.



Создаем новый документ

Делаем заголовок <title>

Делаем заголовок первого уровня

Вставляем комментарий, о том, что это шапка нашего сайта

```
<!--Шапка сайта-->
```

```
<h1>Мой первый сайт</h1>
```

Вставляем в него логотип

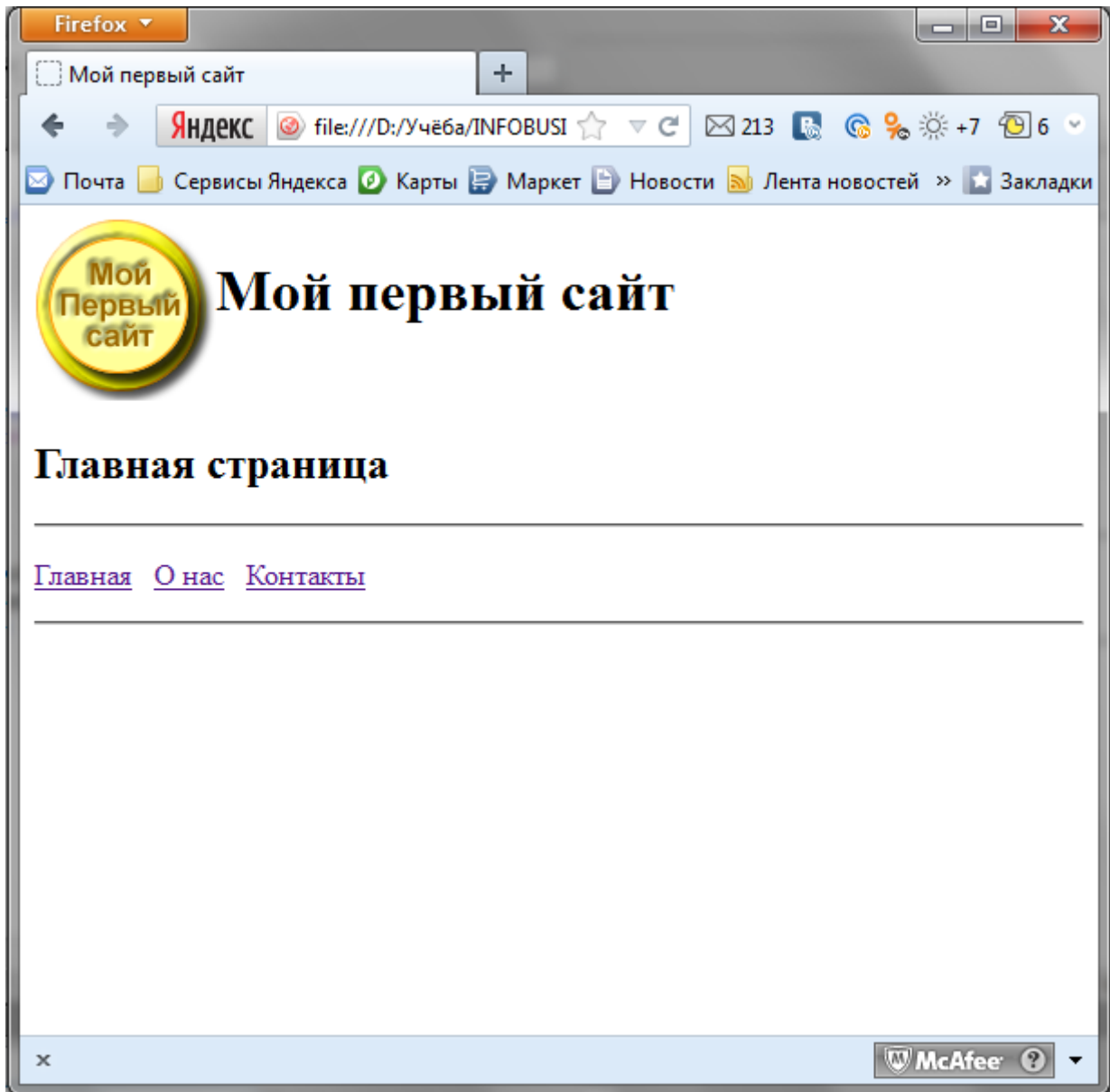
```
<h1><a href="index.html"></a>Мой первый сайт</h1>
```

Логотипу добавляем атрибут align="middle", что позволит ему красиво расположиться относительно заголовка.

Делаем логотип ссылкой, которая будет нас переадресовывать на главную страницу index.html.

Ставим комментарий о том, что здесь заканчивается шапка сайта

```
<!--Конец Шапка сайта-->
```

Кликаем по кнопочкам меню и по логотипу, проверяем работу сайта.

Теперь потренируемся работать с якорями.

Возьмем страницу `index.html` и заполним его каким-нибудь длинным текстом, разбитым на параграфы. Можно использовать рыбу Lorem ipsum можно какой-нибудь другой текст. Лишь бы он был длинным.

Lorem ipsum берем на сайте <http://www.lipsum.com/>

Если скопировать сгенерированный текст прямо из браузера, а затем для скорости вставить его в окно дизайнера программы DREAMWIEVER, то текст самостоятельно разобьется на параграфы. Потому что DREAMWIEVER сохраняет разметку текста в первоизданном виде.

Обратите внимание, что с сайта ipsum.com скопировался незнакомый вам тег `<div></div>`

Это очень полезный тег, в недалеком будущем он нам очень пригодится. Называется он контейнер.

Он так же как и `<pre>` является блочным тегом, но в отличии от `<pre>` он не оставляет между последовательными контейнерами промежутков и позволяет внутри себя использовать другие блочные теги.

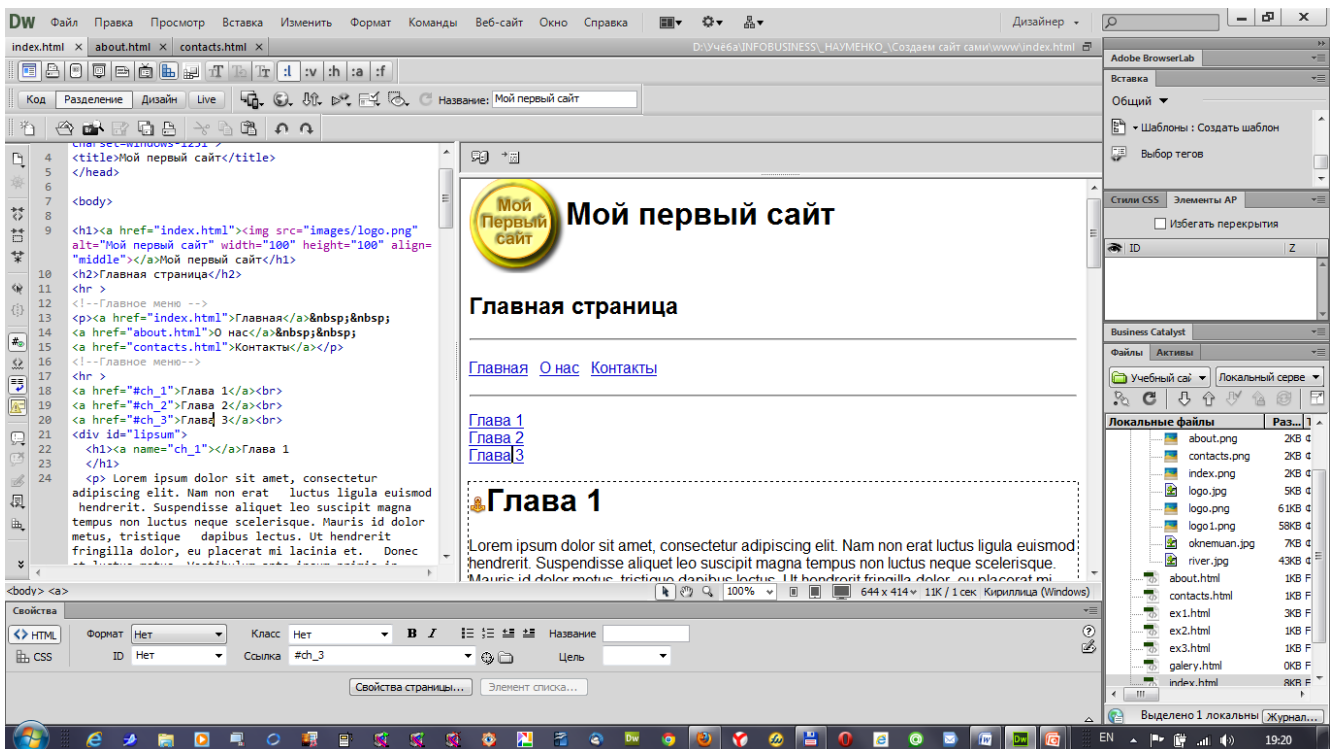
```
<div>
    <div>
        <div>
            </div>
        </div>
    </div>
</div>
```

После того, как текст добавлен, разбиваем его на главы в произвольном порядке.

Добавляем заголовки к каждой главе

Перед заголовками расставляем якоря.

делаем меню для навигации по главам.



Меню навигации разбиваем тегом перевода строки `
`

Попробуйте все сделать сами, читая задания №3, №4 и заглядывая в подсказки на иллюстрациях. Уверен, что справитесь!

Задание № 3

1. Создание многостраничного сайта
2. Создайте новый пустой HTML документ
3. Создайте заголовок `<title> Мой первый сайт</title>`
4. Создайте шапку сайта заголовок `<h1>`, например «Мой первый сайт» и расположите слева от него Логотип
5. Логотипу присвойте ссылку на `index.html`
6. Заключите шапку сайта в комментарии

например: `<!-- Шапка сайта-->`

`<!-- Конец Шапка сайта-->`

7. Создайте заголовок страницы `<h2> Главная страница`
8. Вставьте горизонтальную линию `<hr>`
9. Создайте главное меню ссылок Главная О компании Контакты
10. Разрядите их двойными неразрывными пробелами ` `;
11. Каждому разделу присвойте соответственно ссылку на `index.html`, `about.html`, `contacts.html`.
12. Под главным меню вставьте горизонтальную черту `<hr>`
13. Заключите главное меню в комментарии в начале и в конце, например:

`<!-- Главное меню-->`

`<!-- Конец Главное меню-->`

14. Сохраните документ как `index.html`
15. Сохраните документ как `about.html`
16. Сохраните документ как `contacts.html`
17. Отредактируйте документы `about.html` и `contacts.html` изменив заголовки `<title>` и заголовки страниц, не трогая шапку и главное меню.
18. Проверьте работу сайта в действии, открыв `index.html` в браузере и покликав по ссылкам и логотипу.

Задание №4

1. Создание меню навигации по странице
2. Откройте файл `index.html`
3. Вставьте объемный текст, например Lorem ipsum (<http://lipsum.com>)
4. Разбейте текст произвольно на главы, к каждой главе добавьте заголовок
5. Перед каждым заголовком поставьте якорь с уникальной меткой главы, например: ``. Метки не должны иметь одинаковые имена.

6. Наверху перед началом текста создайте меню глав. Разбейте его вертикально с помощью тега переноса строки `
`
7. Каждому элементу меню присвойте ссылку на якорь соответствующей главы. Например: `Глава 1
`
8. Сохраните документ и просмотрите его в браузере, проверьте работоспособность меню навигации.

Урок № 5

Списки

Сегодня мы с Вами научимся делать списки.

Что такое список?

Список это маркированный или нумерованный набор текстовых строк или изображений.

Например:

- Кошка
- Собака
- Корова
- Лошадь

Или:

1. AUDI
2. BMW
3. Mercedes
4. Peugeot

Для чего нужны списки?

Списки нужны для упорядочения и систематизации набора данных, а также для представления их в удобном для прочтения виде.

Многие новички мне часто задают вопрос: Ну и что? Зачем изучать эти списки, если мы и так умеем переносить строки и сделаем любой список, используя цифры или специальные символы например:

1. AUDI

2. BMW

3. Mercedes

4. Peugeot

Или:

- •Кошка

- •Собака

- •Корова

- •Лошадь

«•» как вы догадались, это спецсимвол, рисующий черненький кружочек-маркер «•».

Все это правильно, для наглядного представления списка можно поступить и так. Только на самом деле список HTML это основа для автоматизации конструирования и отображения иерархических древовидных систем. На основе списков с использованием каскадных таблиц стилей и языка программирования JavaScript и его расширения JQuery делаются всевозможные меню для сайтов. Линейные и вложенные. Поэтому списки – очень полезная вещь, и мы сейчас научимся создавать их.

В HTML существует три вида списков:

- Маркированные
- Нумерованные
- Списки определений

Начнем по-порядку.

Маркированные списки

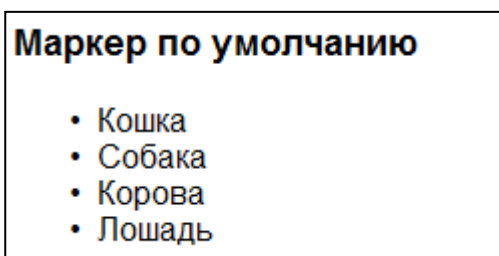
Тег маркированного списка ``

Между открывающим и закрывающим тегами `ul` содержатся элементы списка.

Каждый элемент списка заключен между тегами ``

```
<ul>
  <li>Кошка</li>
  <li>Собака</li>
  <li>Корова</li>
  <li>Лошадь</li>
</ul>
```

В таком виде по умолчанию отобразится список с маркерами в виде черного закрашенного кружочка •



Маркер может отображаться также в виде окружности «○» и черного квадратика «■». Чтобы изменить вид маркера необходимо в теге `` задать параметры атрибуту `type`. Атрибут `type` принимает параметр `«circle»` для отображения окружности:

```
<ul type="circle">
  <li>Кошка</li>
  <li>Собака</li>
  <li>Корова</li>
  <li>Лошадь</li>
</ul>
```

Маркер окружность

- Кошка
- Собака
- Корова
- Лошадь

square-для отображения черного квадратика:

```
<ul type="square">
  <li>Кошка</li>
  <li>Собака</li>
  <li>Корова</li>
  <li>Лошадь</li>
</ul>
```

Маркер черный квадратик

- Кошка
- Собака
- Корова
- Лошадь

и disc- для отображения черного закрашенного кружочка, как по умолчанию:

```
<ul type="disc">
  <li>Кошка</li>
  <li>Собака</li>
  <li>Корова</li>
  <li>Лошадь</li>
</ul>
```

Маркер кружочек

- Кошка
- Собака
- Корова
- Лошадь

Нумерованные списки

Тег нумерованного списка ``

Каждый элемент списка заключен между тегами ``

```
<ol>
  <li>AUDI</li>
  <li>BMW</li>
  <li>Mercedes</li>
  <li>Peugeot</li>
</ol>
```

По умолчанию в качестве нумерующего маркера отображаются арабские цифры.

По умолчанию

1. AUDI
2. BMW
3. Mercedes
4. Peugeot

С помощью параметров атрибута **type** тега `` можно задавать другие значения нумерующего маркера:

Арабские числа (1, 2, 3, ...); `type="1"`;

```
<ol type="1">
  <li>AUDI</li>
  <li>BMW</li>
  <li>Mercedes</li>
  <li>Peugeot</li>
</ol>
```

Арабские числа (1, 2, 3, ...)

1. AUDI
2. BMW
3. Mercedes
4. Peugeot

Прописные латинские буквы (A, B, C, ...); `type="A"`;

```
<ol type="A">
  <li>AUDI</li>
  <li>BMW</li>
  <li>Mercedes</li>
  <li>Peugeot</li>
</ol>
```

Прописные латинские буквы (A, B, C, ...)

- A. AUDI
- B. BMW
- C. Mercedes
- D. Peugeot

Строчные латинские буквы (a, b, c, ...); `type="a"`;

```
<ol type="a">
  <li>AUDI</li>
  <li>BMW</li>
  <li>Mercedes</li>
  <li>Peugeot</li>
</ol>
```

Строчные латинские буквы (a, b, c, ...)

- a. AUDI
- b. BMW
- c. Mercedes
- d. Peugeot

Прописные римские числа (I, II, III, ...); type="I";

```
<ol type="I">
  <li>AUDI</li>
  <li>BMW</li>
  <li>Mercedes</li>
  <li>Peugeot</li>
</ol>
```

Прописные римские числа (I, II, III, ...)

- I. AUDI
- II. BMW
- III. Mercedes
- IV. Peugeot

Строчные римские числа (i, ii, iii, ...); type="i".

```
<ol type="i">
  <li>AUDI</li>
  <li>BMW</li>
  <li>Mercedes</li>
  <li>Peugeot</li>
</ol>
```

Строчные римские числа (i, ii, iii, ...)

- i. AUDI
- ii. BMW
- iii. Mercedes
- iv. Peugeot

Другой атрибут тега ``, **start** позволяет начинать нумерацию списка с заданного значения. Значение задается в качестве параметра атрибута **start**.

При этом тип списка значения не имеет.

```
<ol type="1" start="5">
  <li>AUDI</li>
  <li>BMW</li>
  <li>Mercedes</li>
  <li>Peugeot</li>
</ol>
```

Демонстрация атрибута start="5"

5. AUDI
6. BMW
7. Mercedes
8. Peugeot

Список определений

Еще одна разновидность списка, состоящего из терминов и их определений.

Тег списка определений `<dl></dl>`

Между открывающим и закрывающим тегами `<dl>` содержатся термины, заключенные в теги `<dt></dt>` и их определения, заключенные в теги `<dd></dd>`.

```
<dl>
  <dt>КОШКИ и СОБАКИ — (англ. «cats and dogs»)</dt>
  <dd>В высшей степени спекулятивные и обычно низко оцениваемые рынком фондовые ценности, по которым дивиденды не выплачиваются.<br >
  [Словарь «Борисов А.Б. Большой экономический словарь. — М.: Книжный мир, 2003. — 895 с.»]
  </dd>
  <dt>ДОЙНАЯ КОРОВА</dt>
  <dd>Название товаров, предприятий, приносящих постоянный, устойчивый доход, обеспечивающих практически неиссякаемый приток прибыли.<br >
  [Словарь «Борисов А.Б. Большой экономический словарь. — М.: Книжный мир, 2003. — 895 с.»]
  </dd>
  <dt>ВЗРОСЛАЯ ЛОШАДЬ</dt>
  <dd>Жеребец, кобыла или мерин старше трех лет<br >
  [ГОСТ 16020-70 Скот для убоя. Термины и определения]
  </dd>
</dl>
```

Список определений

КОШКИ и СОБАКИ — (англ. «cats and dogs»)

В высшей степени спекулятивные и обычно низко оцениваемые рынком фондовые ценности, по которым дивиденды не выплачиваются.

[Словарь «Борисов А.Б. Большой экономический словарь. — М.: Книжный мир, 2003. — 895 с.»]

ДОЙНАЯ КОРОВА

Название товаров, предприятий, приносящих постоянный, устойчивый доход, обеспечивающих практически неиссякаемый приток прибыли.

[Словарь «Борисов А.Б. Большой экономический словарь. — М.: Книжный мир, 2003. — 895 с.»]

ВЗРОСЛАЯ ЛОШАДЬ

Жеребец, кобыла или мерин старше трех лет

[ГОСТ 16020-70 Скот для убоя. Термины и определения]

Вложенные списки

Для создания древовидной иерархии существует возможность создания вложенных списков.

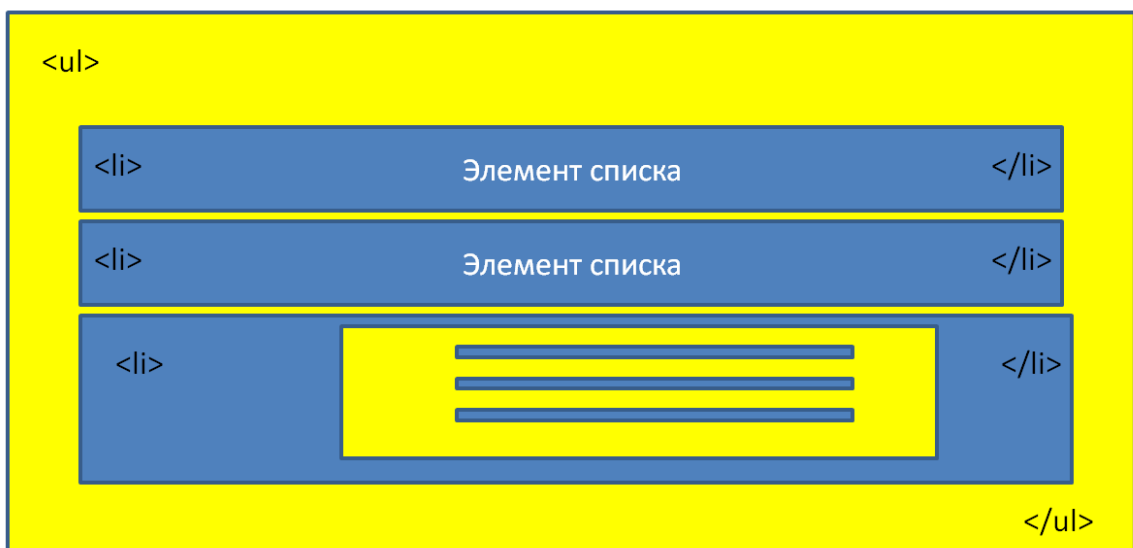
Для того, чтобы понять как устроены вложенные списки, рассмотрим структуру списка.

Тег `` или `` представляет собой контейнер, в котором находятся элементы списка, заключенные в мини-контейнеры ``.

В обычном списке элементом списка может быть текст или изображение.

Во вложенном списке элементом списка-родителя может быть другой список `` или ``, заключенный в теги ``.

Вложенные списки



```
<ul>
<li>Животные
  <ul>
    <li>Кошка</li>
    <li>Собака</li>
    <li>Корова</li>
    <li>Лошадь</li>
  </ul>
</li>
<li>Автомобили
  <ol>
    <li>AUDI</li>
    <li>BMW</li>
    <li>Mercedes</li>
    <li>Peugeot</li>
  </ol>
</li>
</ul>
```

Вложенные списки

- Животные
 - Кошка
 - Собака
 - Корова
 - Лошадь
- Автомобили
 1. AUDI
 2. BMW
 3. Mercedes
 4. Peugeot

Урок №6

Таблицы

Сегодня мы с Вами изучим, как устроены таблицы в HTML и прикоснемся к табличной верстке

Как выглядит таблица, знают все! Это строки и столбцы.

В HTML также существуют понятия строки, столбца и ячейки.

Тег таблицы `<table></table>`

Между открывающим и закрывающим тегами `<table>` размещаются теги строк

`<tr></tr>` Между тегами строк размещаются теги ячеек `<td></td>`. Сколько в таблице предполагается столбцов, столько ячеек помещается в строку. Таким образом получается, что столбцы по вертикали выстраиваются из ячеек.

- Тег таблицы `<table></table>`
- Теги строк `<tr></tr>`
- Теги столбцов `<td></td>`

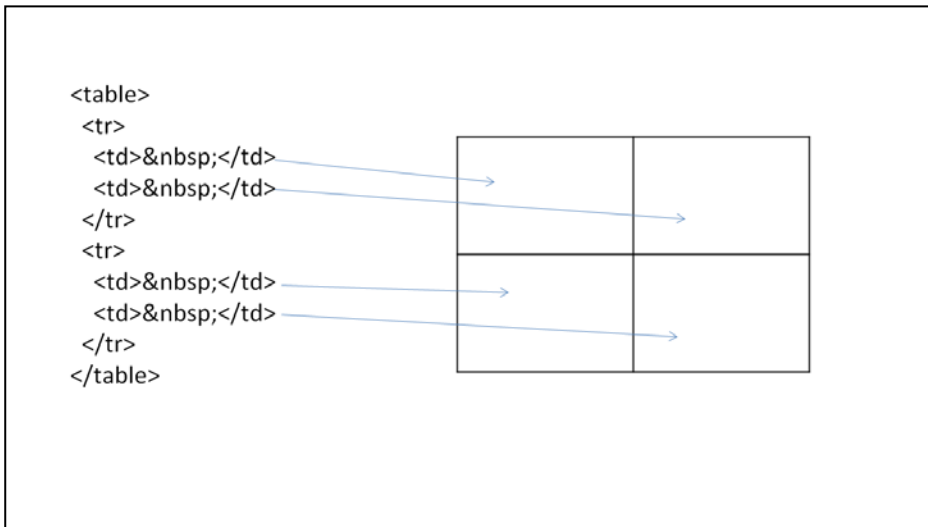
В простом виде таблица, состоящая из двух строк и двух столбцов выглядит так:

```
<table>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
</table>
```

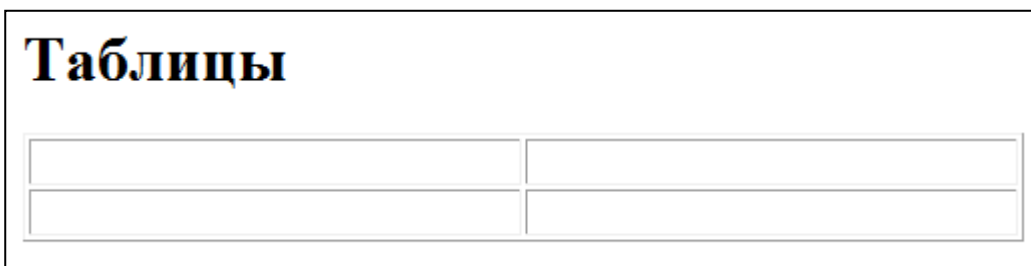
Однако в таком виде таблица отображаться в браузере не будет, так как границы таблицы в данном представлении будут прозрачными.

Для того, чтобы таблица стала видна, необходимо в теге `<table>` установить необходимые атрибуты.

```
<table width="500" border="1">
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
</table>
```



Мы установили ширину таблицы равной 500 пикселей и задали рамку с помощью атрибута `border` шириной в 1 пиксел. По умолчанию размеры ячеек установились равной ширины.



Обратите внимание – границы таблицы выглядят по умолчанию рельефно. Это нормально, так как у нас не задано больше никаких параметров.

Пробуем изменить размер ячейки, установим ширину ячейки 100 пиксел

```
<table width="500" border="1">
  <tr>
    <td width="100">&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
</table>
```

Меняем размер ячейки

Обратите внимание: Мы изменили только ширину одной ячейки, а получилось, что мы изменили ширину всего столбца. Второй столбец увеличился пропорционально автоматически.

Зададим расстояние между ячейками и поменяем цвет рамки:

```
<table width="500" border="1" cellspacing="0" bordercolor="black">
  <tr>
    <td width="100">&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
</table>
```

Устанавливаем расстояние между ячейками для всей таблицы с помощью атрибута тега таблицы cellspacing равным нулю и задаем цвет границ с помощью атрибута bordercolor черным.

Меняем расстояние между ячейками

Нарращиваем количество столбцов, вставляем в ячейки текст, форматированный текст, изображения


```
<table width="500" border="1" cellspacing="0" bordercolor="#000000">
  <tr>
    <td width="100">Текст</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
  <tr>
```

```

<td>&nbsp;</td>
<td><h3 align="center">Форматированный текст</h3>
<p>Форматированный текст абзац</p></td>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td></td>
<td>Текст</td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
</table>

```

Наращиваем количество столбцов и строк, вставляем текст и картинки

Текст			
	Форматированный текст Форматированный текст абзац		
			Текст

Устанавливаем отступы от содержимого ячеек до ее границ с помощью атрибута cellpadding.

В ячейку таблицы можно вставить другую таблицу.

```

<table width="500" border="1">
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
<tr>
<td>&nbsp;</td>
<td>
<table width="500" border="1">
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
</table>
</td>
</tr>
</table>

```

В ячейку таблицы можно вставить другую таблицу



Представление таблицы.

Можно задать заголовок таблицы с помощью тега `<caption>` `</caption>`

который должен находиться внутри тегов `<table>` `</table>`

Можно также использовать тег `<tbody>` `</tbody>`.

Для заголовков столбцов вместо тега `<td>` можно использовать тег `<th>`.

Для объединения ячеек по горизонтали можно использовать атрибут `colspan` в параметре которого указывать количество объединяемых ячеек.

Для объединения ячеек по вертикали используется атрибут `rowspan`, параметром которого является количество объединяемых ячеек.

```
<table width="500" border="1">
<caption>Представление таблицы</caption>
<tbody>
  <tr>
    <th>1 колонка</th>
    <th>2 колонка</th>
    <th>3 колонка</th>
  </tr>
  <tr>
    <td rowspan="2">2 объединенные строчки</td>
    <td colspan="2">Две объединенные колонки</td>
  </tr>
  <tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
</tbody>
</table>
```

Представление таблицы

1 колонка	2 колонка	3 колонка
2 объединенные строчки	Две объединенные колонки	

Использование таблиц

В HTML таблицы были придуманы, прежде всего, для представления и отображения данных в виде таблицы, например:

Прайс-лист

Артикул	Наименование	Цена Розн	Цена Опт
1102615	Пинбол Cars2	1697	1131
CAM001C	MP3 плеер 2GB. Disney Cars	2590	1750
HEM120C	MP3 плеер 2GB с наушниками и динамиком. Hello Kitty	2690	1750
HEM060C	MP3 плеер 2GB в форме персонажа. Hello Kitty	1590	1050
HEM010C	MP3 плеер 2GB. Hello Kitty	1590	1050
HEC001N	MP4 плеер со встроенной камерой. Hello Kitty	2490	1700
HES010Z	Колонки. Hello Kitty	1090	730
HEE130Z	Наушники. Hello Kitty	790	550
HEA021Z	Сумка для фотоаппарата пластиковая. Hello Kitty	440	290
HEA030Z	Сумка для фотоаппарата. Hello Kitty	370	250
PKC001M	Фотоаппарат 5Мп с сумкой. Hello Kitty	2890	1950

Обратите внимание на оформление заголовков столбцов в верхней строке. Теги столбцов в верхней строке `<td></td>` заменены на `<th></th>`. Именно теги `<th>` оформляют заголовки столбцов жирным шрифтом и посередине ячейки.

Позже, web-дизайнеры стали использовать таблицы для разметки страницы и появился, так называемый, табличный дизайн. Суть табличного дизайна состоит в том, что все элементы дизайна вставляются в ячейки таблицы, которая формирует макет страницы.

Задание № 5



1. Создаем таблицу в два столбца и три строки. Ширина таблицы 800 px. Установите ширину рамки 0.
2. В верхней строке в левую ячейку вставляем картинку логотипа и устанавливаем ширину ячейки 20%. Во вторую ячейку первой строки вставляем картинку графического заголовка «Мой первый сайт»
3. Во второй строке объединяем ячейки и вставляем в нее картинки кнопок меню.
4. В первой ячейке третьей строки формируем вертикальное меню глав со ссылками на якоря в основном контенте.
5. Во второй ячейке третьей строки формируем основной контент страницы: главы текста с иллюстрациями и якорями.
6. Вот полный код страницы, выполненной в табличной верстке.

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
<title>Документ без названия</title>
</head>

<!--Задаем голубой фон для всей страницы-->

<body bgcolor="#66CCFF">

<!--Задаем Таблицу шириной 800px с нулевой рамкой, выровненную по центру, расстояние между текстом и границами ячеек 10 px, расстояние между ячейками 0 -->

<table width="800" border="0" align="center" cellpadding="10" cellspacing="0">

<tr>
<!--Левая ячейка первой строки с логотипом-->
  | |
```

```
<a href="index.html"></a>
<a href="about.html"></a>
<a href="contacts.html"> </a>

</td>
</tr>
<tr>

<!--Первая ячейка третьей строки с вертикальным меню глав-->

<td align="center" valign="top" bgcolor="#FFFF9D"><a href="#ch_1">Глава 1</a><br>
<a href="#ch_2">Глава 2</a><br>
<a href="#ch_3">Глава 3</a></td>

<!--Вторая ячейка третьей строки с основным контентом страницы-->

<td align="left" valign="top" bgcolor="#FFFFFF">
<h2><a name="ch_1"></a>Глава 1</h2>

<div align="justify">
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse id
dui vel nulla pellentesque
vehicula. Vestibulum tincidunt velit non arcu convallis a sollicitudin
mauris sollicitudin. Curabitur
at ipsum justo, sed cursus justo. Proin adipiscing vestibulum justo id
tincidunt. Quisque accumsan
sapien aliquet arcu faucibus vitae gravida velit egestas. Aliquam et
sem sem. Donec ut sapien a enim
tempus pharetra ut at diam. Maecenas et hendrerit augue. Phasellus
pellentesque condimentum mollis.
Fusce rhoncus ligula vel lorem pretium sit amet bibendum nibh tempus.
Integer eget leo elit, eget
rhoncus tellus. Sed suscipit euismod dolor a dictum. Nulla semper
tellus quam, vitae laoreet tortor.
Aenean eget dui quam.

<p> Vestibulum sodales libero a sem fermentum vitae vehicula mi blandit.
Nam convallis est sed
felis tempus tincidunt. Quisque iaculis turpis vel urna mattis at
interdum metus semper. Aliquam
augue dui, ullamcorper vel pharetra quis, aliquam id sem. Praesent
sagittis diam a sem hendrerit
interdum. Sed fringilla pharetra ornare. In molestie cursus fringilla.
Proin dapibus magna iaculis
velit pellentesque rutrum. Praesent elementum pharetra odio, quis
interdum eros cursus ut. Vestibulum
ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia
Curae; Aliquam erat volutpat.
Cras vel ipsum dolor. In sit amet nibh sem, sit amet scelerisque urna.
Phasellus purus velit,
malesuada nec interdum sit amet, volutpat ut ante. Suspendisse
consequat consequat velit quis
hendrerit. Nulla id massa in enim dictum volutpat a auctor urna. </p>

<p> In imperdiet nunc tellus. Donec quis lorem velit, dignissim
vehicula enim. Curabitur ultricies
posuere mi at dapibus. Vestibulum ante ipsum primis in faucibus orci
luctus et ultrices posuere
cubilia Curae; Curabitur ut neque libero. Nam sed odio a lorem
vulputate ultrices a in risus.
Pellentesque est justo, varius sit amet congue in, volutpat gravida
arcu. Etiam quis bibendum erat.
Etiam tempus sapien eu felis scelerisque fringilla. Nulla malesuada
leo nec justo lobortis eget
dignissim odio fermentum. Mauris condimentum lacinia neque at
vulputate. Nam risus dui, laoreet non
imperdiet eu, vehicula eget lectus. Curabitur tristique nisi et
turpis tristique vel dapibus neque
rutrum. Phasellus non turpis a eros euismod ultricies. Pellentesque
eu porta sem. Fusce in erat
erat. </p>

<h2><a name="ch_2"></a>Глава 2</h2>

<p> Sed eros metus, fringilla a bibendum eu, porta ut felis. Vivamus
placerat, neque et egestas
elementum, massa elit sodales est, at feugiat neque metus non elit.
Donec eu iaculis erat. Class
aptent taciti sociosqu ad litora torquent per conubia nostra, per
inceptos himenaeos. Sed sit amet
dapibus lectus. Praesent et nisl purus. Donec sagittis fermentum
bibendum. Integer vel lectus magna.
Mauris gravida risus vitae justo dictum ut tincidunt risus commodo.
Mauris porta vestibulum turpis.
Proin ut convallis nunc. Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Pellentesque
commodo pharetra libero eu ultricies. Vestibulum venenatis felis non
leo tempor dapibus. Integer
euismod, massa sed congue eleifend, augue diam pretium nisi, et
suscipit purus ante eget massa.
Aliquam id nisi quis dui viverra facilisis. </p>

<p> Sed pulvinar iaculis ante quis feugiat. Mauris mollis iaculis
convallis. Nam sodales accumsan
mi eget malesuada. Etiam molestie facilisis eros et semper. Aliquam
interdum faucibus sapien id
gravida. Phasellus massa tellus, volutpat tristique hendrerit
dignissim, mattis sit amet tellus.
Nunc tempor fermentum risus, a auctor urna ultrices ac. Curabitur ac
magna justo, non tristique
dolor. Nulla consectetur nibh sed enim pretium ut sodales mauris
tempus. Duis pellentesque ante vitae
libero viverra vitae laoreet leo mollis. </p>

<p> Pellentesque dapibus, eros at accumsan consectetur, quam nisl
porta turpis, in imperdiet elit
mi in leo. Ut libero eros, euismod in placerat in, commodo vel nisi.
Vestibulum at sapien enim. Donec
id porta odio. Curabitur odio magna, pellentesque in rutrum sit amet,
malesuada at dui. Nulla
mollis tincidunt leo, sed imperdiet metus dictum at. Donec faucibus
convallis est non condimentum.
Mauris quis rutrum orci. Mauris varius velit porttitor est ultrices
pellentesque. Vivamus accumsan
consequat fringilla. Sed eget purus sit amet sapien tempus
adipiscing et suscipit eros. </p>

<p> Praesent in libero vel nisl placerat laoreet. Curabitur enim mi,
vestibulum in malesuada quis,
mollis egestas augue. Phasellus bibendum ultrices nunc nec varius.
Nam in nisi tellus. Curabitur
tincidunt quam fringilla quam sollicitudin viverra. Nulla a nunc
volutpat erat laoreet aliquet id
vel ante. Etiam viverra, nunc ut aliquet imperdiet, nunc nisl
tristique lacus, sit amet pulvinar dui
leo et leo. Fusce placerat convallis ante a luctus. Nullam at ante
felis. </p>

<p> Pellentesque tristique scelerisque tincidunt. Phasellus
consectetur, orci ullamcorper convallis
lacinia, turpis orci blandit libero, quis congue velit arcu at mi.
Ut urna elit, dictum sit amet
tincidunt ut, rutrum tincidunt mauris. Nulla venenatis adipiscing
justo, nec tincidunt neque
pretium ac. Curabitur in mauris neque. Aenean felis diam, fringilla
nec facilisis ut, egestas et
```

```
risus. Aenean dignissim hendrerit iaculis. Suspendisse lacus lectus, dapibus eget aliquam at, luctus
ut lorem. Maecenas et arcu orci, ut tristique tortor. Fusce interdum nibh quam. Etiam elit est,
sagittis vel facilisis eu, interdum consectetur nisl. Aliquam et eros elit, eleifend mollis
metus.</p>
<h2><a name="ch_3"></a>Глава 3</h2>
<p> Pellentesque pulvinar, dolor non hendrerit tristique, augue nibh vulputate tortor, eget
eleifend lacus magna at lorem. Morbi leo sapien, tempor at accumsan sed, vestibulum eget nisi. Morbi
et nisi lorem, a ultricies nisl. Proin est ipsum, luctus et semper et, rutrum id mi. Proin odio
nunc, consectetur a faucibus quis, iaculis et diam. Praesent nec nunc non enim tempus faucibus ut et
libero. Mauris eget commodo massa. Vivamus sodales laoreet enim, sed porttitor quam scelerisque
hendrerit. Nam porta, purus sit amet semper varius, ipsum est consequat augue, a tristique nibh massa
id urna. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae;
Morbi posuere suscipit velit, in ultricies orci euismod et. Cras nec nulla turpis, eget luctus est.
Curabitur lacinia consectetur tortor, at vestibulum nisi mollis ac. Sed risus tortor, rutrum eget
rhoncus ac, auctor at elit. </p>
<p> Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos.
Nullam magna metus, consequat ut tempus a, faucibus sit amet orci. Duis in auctor ante. Nulla
vehicula sapien elit, quis porta diam. Curabitur vitae dui libero, in semper risus. Donec malesuada
risus sed nisi convallis sit amet iaculis nisi sollicitudin. Sed luctus pretium est varius mattis.
Nam facilisis dui pulvinar elit hendrerit congue. In purus est, lacinia non ullamcorper non,
imperdiet vitae lacus. In hac habitasse platea dictumst. Nunc leo tellus, tincidunt nec eleifend
vel, ornare id nisi. Curabitur sit amet neque et magna suscipit lacinia. Vestibulum ipsum tortor,
rutrum ac scelerisque at, tristique a erat. Praesent sit amet ipsum ut metus ultricies auctor non non
lacus. </p>
</div>
</td>
</tr>
</table>
</body>
</html>
```

А это результат. Так должно выглядеть в браузере

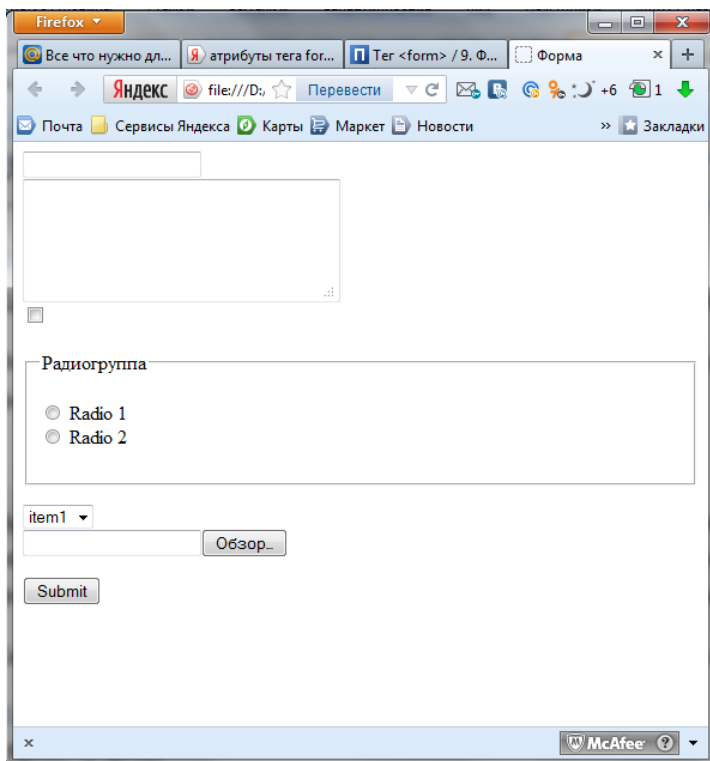


Урок №7

Формы

Для возможности ввода данных и интерактивного общения с WEB-сервером в HTML придуманы формы.

Формы представляют собой поля для ввода текста, списки для выбора данных, поле для отправки файлов, скрытые поля, чекбоксы, радиокнопки и кнопки.



Вы наверное знаете, как выглядят формы, потому что сталкиваетесь с ними в интернете повсеместно, когда, например, вводите логин и пароль, осуществляете поиск в Яндексе или загружаете файлы. В форму вы также вводите текст, когда пишете электронное письмо и общаетесь в социальных сетях.

Сегодня мы с вами изучим, как устроены формы и научимся их создавать

Тег формы

`<form></form>`

Между этими тегами располагаются элементы формы.

Тег формы имеет множество атрибутов, например:

```
<form
action="http://elearn.oknemuan.ru/testform/"
method="post"
enctype="multipart/form-data"
```

```
name="form">
```

Разберем их по-порядку.

Атрибут **action** в своем параметре содержит URL адрес программы-обработчика формы. В нашем примере все данные которые введены в Форму после нажатия кнопки будут отправлены на WEB-сервер с доменом elearn.oknemun.ru, где в папке `testform` содержится файл-обработчик формы `index.php`, который я написал специально для того, чтобы продемонстрировать работу учебных форм.

Изучение устройства обработчиков форм лежит за рамками данного учебного курса. Это тема моего следующего курса, посвященного WEB-программированию.

Сейчас вам просто следует понять, что форма сама по себе не обрабатывает данные, а лишь только служит для их передачи на сервер или программе-обработчику событий, которая обрабатывает их прямо в браузере.

Следующий атрибут **method** указывает, каким методом данные передаются программе-обработчику.

Существует два метода `post` и `get`. Их мы тоже не будем подробно обсуждать в рамках данного курса. Скажу лишь, что методом `get` можно передать лишь 255 символов, а методом `post` гораздо больше, поэтому для передачи больших объемов информации следует указывать метод `post`.

Атрибут **enctype** в данном примере указывает, что через форму будут переданы файлы, поэтому значение параметра устанавливается **multipart/form-data**.

Атрибут **name** задает имя формы.

Элементы формы

Группа `<input>`

Текстовое поле

Предназначено для ввода текста и паролей.

```
<input name="textfield" type="text">
```

Атрибут `name` является обязательным атрибутом для всех элементов форм. Он должен быть уникальным в пределах одной формы. Именно имя элемента и его значение передается в дальнейшем обработчику.

Атрибут **type** очень важный именно он определяет вид элемента группы `input`. В данном случае `type="text"`, это обозначает, что поле выглядит как прямоугольник и в него можно вводить любой текст.

Можно ввести текст по умолчанию, используя атрибут `value`:

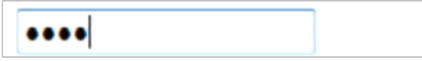
```
<input name="textfield" type="text" value="Введите любой текст">
```

Вот результат:

Для ввода паролей обычно используют другое значение атрибута type:

```
<input type="password" name="password" >
```

В этом случае все введенные в поле символы скрываются и отображаются в виде символов. Кружочки или звездочки в IE.



Чекбокс или флажок

Для отображения чекбокса используется тип checkbox

```
<input name="checkbox" type="checkbox" value="1" >
```



Чекбоксу можно присваивать любые значения, которые при отправке формы будут передаваться серверу. В данном случае будет передано значение 1.

Можно сделать так, чтобы чекбокс по умолчанию был отмечен.



```
<input name="checkbox" type="checkbox" value="1" checked>
```

для этого устанавливается атрибут checked, который не имеет параметров.

Радиокнопка

Следующий элемент группы input это радиокнопка

```
<input type="radio" name="RadioGroup1" value="radio1" >
```



С помощью атрибута checked, можно также задать ее по умолчанию, как выбранную.



У радиокнопки есть одно свойство, которое отличает ее от чекбокса. Радиокнопки можно объединять в группы. Если задать несколько радиокнопок с одним именем и разными значениями, то выбирать (включать) можно лишь одну кнопку.

```
<input type="radio" name="RadioGroup1" value="radio2" >  
<input type="radio" name="RadioGroup1" value="radio1" >
```

Поле для передачи файла

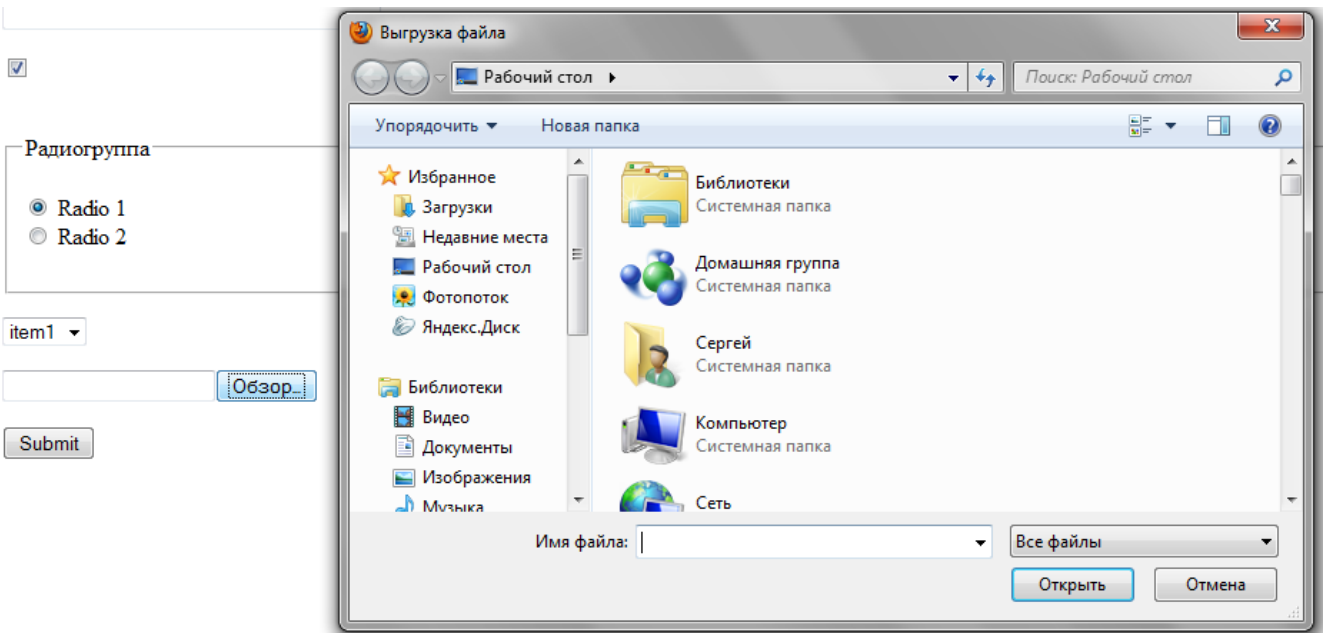
Следующий элемент группы input служит для передачи на сервер файлов.

```
<input type="file" name="userfile" >
```

Это тип file. Отображается он в виде прямоугольничка с кнопкой Обзор



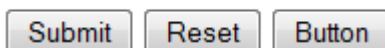
При нажатии на кнопку Обзор открывается доступ к файловой структуре компьютера, где можно осуществить выбор любого файла для передачи его серверу – обработчику формы.



Кнопки

Следующий элемент формы группы input это кнопка. Кнопки бывают трех типов:

```
<input type="submit" name="submit" id="button" value="Submit" >  
<input type="reset" name="reset" id="reset" value="Reset" >  
<input type="button" name="button" id="button" value="Button" >
```



Тип submit позволяет отправлять данные формы обработчику по адресу, указанному в атрибуте action тега form.

Тип reset позволяет очистить форму от введенных данных.

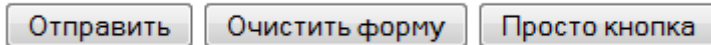
Тип button создает просто кнопку, при нажатии на которую ничего не происходит.

Возникает резонный вопрос – зачем вообще нужна такая кнопка, которая ничего не делает? На самом деле, к такой кнопке можно применить любое событие с помощью языка javascript. И тогда кнопка будет выполнять запрограммированное событие. Но это опять из области WEB-программирования, которое мы будем изучать в соответствующем курсе.

С помощью атрибута value мы можем задавать любые надписи на кнопках.

Например:

```
<input type="submit" name="submit" id="button" value="Отправить">
<input type="reset" name="reset" id="reset" value="Очистить форму">
<input type="button" name="button" id="button" value="Просто
кнопка">
```



Кнопка-изображение

Еще один элемент группы input это картинка

```
<input name="image" type="image" src="images/logo.jpg" >
```



В атрибуте src устанавливается путь к файлу картинки. Работает так же, как и кнопка Submit, то есть отправляет данные формы на сервер, дополнительно передает координаты верхнего левого угла картинки на сервер.

Скрытое поле

```
<input name="hidden" type="hidden" value="Я есть, но меня не видно">
```

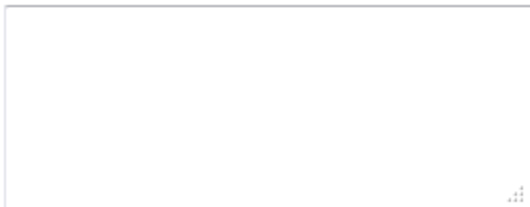
Скрытое поле не отображается в браузере. Оно предназначено для передачи данных, обозначенных в атрибуте value.

Итак, мы рассмотрели с вами элементы формы группы input.

Осталось рассмотреть еще два элемента

Многострочное поле для ввода большого текста

```
<textarea name="textarea" cols="30" rows="5"></textarea>
```



Для ввода текста по умолчанию его помещают между тегами `<textarea></trxtarea>`
Изменить площадь поля можно увеличив параметры атрибутов cols и rows

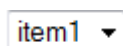
Список и поле со списком для выбора

Наиболее сложный элемент формы, состоит из контейнера

`<select></select>` в который вложены элементы для выбора `<option></option>`

```
<select name="jumpMenu" id="jumpMenu" >
  <option value="1">item1</option>
  <option value="2">item2</option>
  <option value="3">item3</option>
</select>
```

Такая конструкция выглядит как поле с выпадающим списком

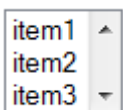


При клике по стрелочке раскрывается список. Каждый элемент списка отображает то, что заключено между тегами `<option></option>`, а после выбора и отправки формы передает обработчику параметр атрибута `value`.

Если в теге `<select>` добавить атрибут `size`, то список будет отображаться, как открытый.

```
<select name="list" id="list" size="3" >
  <option value="1">item1</option>
  <option value="2">item2</option>
  <option value="3">item3</option>
  <option value="4">item4</option>
  <option value="5">item5</option>
  <option value="6">item6</option>
</select>
```

В данном примере при `size =3` будет отображаться три элемента списка.



Остальные доступны при прокрутке.

При отправке формы на сервер будет передаваться значение, соответствующее выбранному элементу, равно параметру атрибута `value` тега `<option>`.

Если атрибут `value` не указывать, то будет передано значение, заключенное между тегами `<option></option>`

«Красивая» форма

А теперь я вам покажу, как можно сделать ровную красивую форму с помощью табличного форматирования на примере анкеты регистрации.

Поставим перед собой задачу: Создать форму регистрации клиента. Клиент должен ввести своё имя, фамилию, пол, электронный адрес, телефон и пароль.

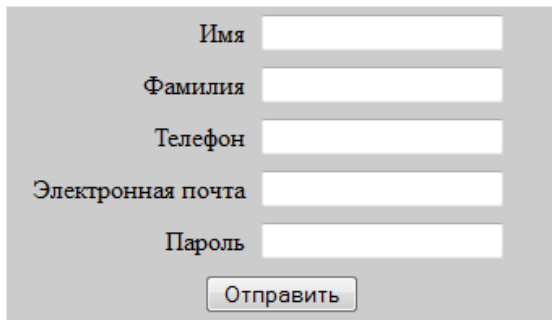
Для этого в новом файле создадим форму и поместим в нее таблицу, состоящую из двух столбцов и шести строк.

```
<form name="form1" method="post" action="">
  <table width="800" border="0" cellspacing="0" cellpadding="5">
    <tr>
      <td>&nbsp;</td>
      <td>&nbsp;</td>
    </tr>
    <tr>
      <td>&nbsp;</td>
      <td>&nbsp;</td>
    </tr>
    <tr>
      <td>&nbsp;</td>
      <td>&nbsp;</td>
    </tr>
    <tr>
      <td>&nbsp;</td>
      <td>&nbsp;</td>
    </tr>
    <tr>
      <td>&nbsp;</td>
      <td>&nbsp;</td>
    </tr>
    <tr>
      <td>&nbsp;</td>
      <td>&nbsp;</td>
    </tr>
  </table>
</form>
```

Теперь в ячейки таблицы вставим названия полей и сами поля:

```
<h1 align="center">Красивая форма</h1>
<form name="form1" method="post" action="http://elearn.oknemuan.ru/testform/">
  <table width="339" border="0" align="center" cellpadding="5" cellspacing="0" bgcolor="#CCCCCC">
    <tr>
      <td align="right" width="142">Имя</td>
      <td align="left" width="177"><input type="text" name="firestname" id="firestname"></td>
    </tr>
    <tr>
      <td align="right">Фамилия</td>
      <td align="left"><input type="text" name="secondname" id="secondname"></td>
    </tr>
    <tr>
      <td align="right">Телефон</td>
      <td align="left"><input type="text" name="telefon" id="telefon"></td>
    </tr>
    <tr>
      <td align="right">Электронная почта</td>
      <td align="left"><input type="text" name="email" id="email"></td>
    </tr>
    <tr>
      <td align="right">Пароль</td>
      <td align="left"><input type="text" name="password" id="password"></td>
    </tr>
    <tr>
      <td align="center" colspan="2"><input type="submit" name="submit" id="submit"
value="Отправить"></td>
    </tr>
  </table>
</form>
```


Красивая форма



Имя

Фамилия

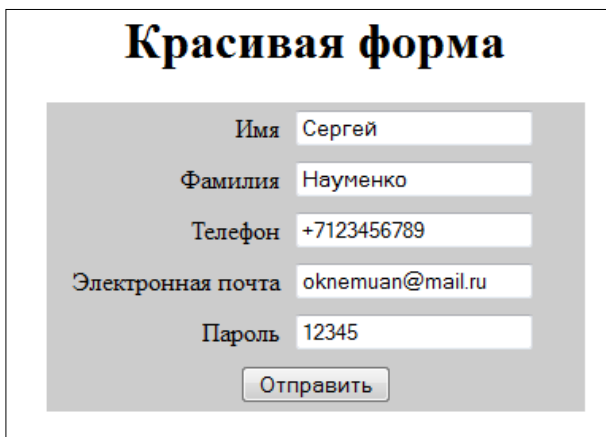
Телефон

Электронная почта

Пароль

Обратите внимание, что в атрибуте action тега form указан URL адрес обработчика формы: <http://elearn.oknemuan.ru/testform/>

Этот обработчик, который я специально написал для тестирования учебных форм, обработает вашу форму после ее отправки нажатием кнопки «Отправить» и отобразит значения, которые вы введете в поля. Компьютер ваш во время отправки формы должен быть обязательно подключен к интернету.



Имя

Фамилия

Телефон

Электронная почта

Пароль

А вот результат работы обработчика после отправки формы:

Результаты обработки формы	
Имя поля	Значение
firstname	Сергей
secondname	Науменко
telefon	+7123456789
email	oknemuan@mail.ru
password	12345
submit	Отправить

Писать программы - обработчики форм я вас научу в специальном курсе про WEB-программирование.

Урок №8

Введение в CSS

Мы с вами изучили основные элементы построения простых сайтов. Уже сейчас вы можете создавать текстовые иллюстрированные web-страницы, связанные между собой гиперссылками.

Однако для того, чтобы сайт выглядел красиво, необходимо научиться применять блочную верстку.

Блочная верстка пришла на смену табличной верстке, потому что табличная верстка не отвечает современным требованиям сайтостроительства. В основном из-за скорости загрузки страницы. Потому что сначала грузится таблица, а потом ее содержимое. С табличной версткой связаны и другие недостатки, мешающие SEO продвижению сайта в поисковых системах. Ну об этом поговорим в другом курсе, посвященному продвижению сайтов.

А сейчас прикоснемся к CSS.

Сегодня в арсенале веб-разработчиков есть замечательный инструмент, который называется Каскадные таблицы стилей *CCS Cascading Style Sheets*.

Сейчас уже трудно встретить в сети сайт, который сверстан без применения технологии CSS.

Основная идея применения CSS для оформления сайта это разделение структуры сайта, которая формируется с помощью HTML, от описания его оформления.

CSS задает правила отображения места и стиля элементов HTML.

CSS представляет собой код, который записывается либо в отдельный файл, либо в файл, содержащий HTML код страницы.

Отдельный файл CSS представляет собой текстовый файл с расширением .css.

Для того, чтобы присоединить CSS файл к WEB-страницам, в разделе <head> записывают следующую конструкцию:

```
<link rel="stylesheet" type="text/css" href="css/styles.css">
```

Данная строка HTML кода привязывает к странице файл styles.css, который расположен в папке css.

Другой способ определения стилей CSS - это размещение кода css непосредственно в странице HTML. Его помещают в разделе head и заключают между тегами <style></style>

Например:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
<title>Заголовок страницы</title>
  <style type="text/css">
    //Здесь размещается код CSS
  </style>
</head>

<body>
</body>
</html>
```

Существует также третий способ присвоения стиля CSS непосредственно в теле тега для которого предназначен стиль. Но о нем поговорим позже.

Удобство применения CSS, когда он хранится в отдельном файле несравнимо для многостраничных сайтов. Потому что, изменения, которые проводятся в одном файле с CSS кодом, распространяются сразу на все страницы сайта, к которым он присоединен с помощью тега link.

Как выглядит CSS код

Рассмотрим пример:

```
p {color: #000000}
```

То, что стоит слева от фигурных скобок называется дескриптором. Выражение, заключенное в фигурные скобки называется декларацией.

В качестве дескрипторов могут выступать все обозначения тегов (в нашем примере это параграф), а также параметры атрибутов HTML тегов: class и id.

В данном примере всем параграфам web-страницы устанавливается черный цвет.

Добавим описание стиля заголовков первого, второго и третьего уровня

```
h1,h2,h3 {
  color:yellow;
  font-family:Verdana, Geneva, sans-serif;
  background:#003399;
  text-align:center
}
```

Здесь всем заголовкам первого, второго и третьего уровня мы установили желтый цвет, семейство шрифтов типа Verdana, фон темно-синий и расположение текста посередине.

Создадим файл styles.css и поместим его в папку css. В файл скопируем созданные нами стили для параграфа и заголовков.

Присоединим стиль к веб-странице index.html, добавив в раздел head строку

```
<link rel="stylesheet" type="text/css" href="css/styles.css">
```

Добавьте эту же строку в файлы `about.html` и `contacts.html`. Посмотрите страницы в браузере. Вы видите, как изменился облик страниц.

Атрибут `class`

Теперь займемся видоизменением верхнего горизонтального меню. Для этого тегу `<a>` во всех ссылках меню добавим атрибут `class` с параметром `tophref`, как на данном примере.

```
<a href="index.html" class="tophref">Главная</a>
<a href="about.html" class="tophref">О нас</a>
<a href="contacts.html" class="tophref">Контакты</a>
```

Атрибуту `class` можно присваивать различные параметры (имена). Этим именам в коде CSS можно присваивать любые доступные свойства, которые будут отображаться, когда какому-нибудь тегу будет приписан данный класс.

Запишем в файле `styles.css` свойства для класса `tophref`. Все имена классов в коде CSS начинаются с точки:

```
.tophref {
}
```

Предположим, что мы хотим отобразить наше меню в виде ряда красивых кнопочек синего цвета.

Поэтому превратим наш тег `<a>` с помощью класса `tophref` в прямоугольные блоки заданной ширины и высоты. Комментарии в CSS принято заключать между `/*и*/`.

```
.tophref {
    text-decoration: none; /*убираем подчеркивание ссылок*/
    display: block; /*Превращаем в блоки*/
    float: left; /*располагаем блоки в один ряд (слева друг от друга)*/
    width: 100px; /*ширина блока*/
    height: 30px; /*Высота блока*/
    /*Назначаем цвет фона, цвет текста и расположение текста относительно блока*/
    background-color: #6633CC; /*Цвет фона, синий*/
    color: #FFFFFF; /*Цвет текста, белый*/
    margin-right: 1px; /*Отступ от блока справа*/
    margin-bottom: 10px; /*Отступ от блока снизу*/
    vertical-align: middle; /*Вертикальное выравнивание текста*/
    font-family: Verdana, Geneva, sans-serif; /*Шрифт текста*/
    text-align: center; /*Горизонтальное выравнивание текста по-середине*/
    padding-top: 15px; /*Отступ текста от верхней границы блока*/
}
```

Теперь, чтобы дальнейшие элементы HTML страницы не выстраивались ряд за рядом вслед за кнопками меню, нам необходимо «остановить» эту рядность, которая была задана свойством `float:left`.

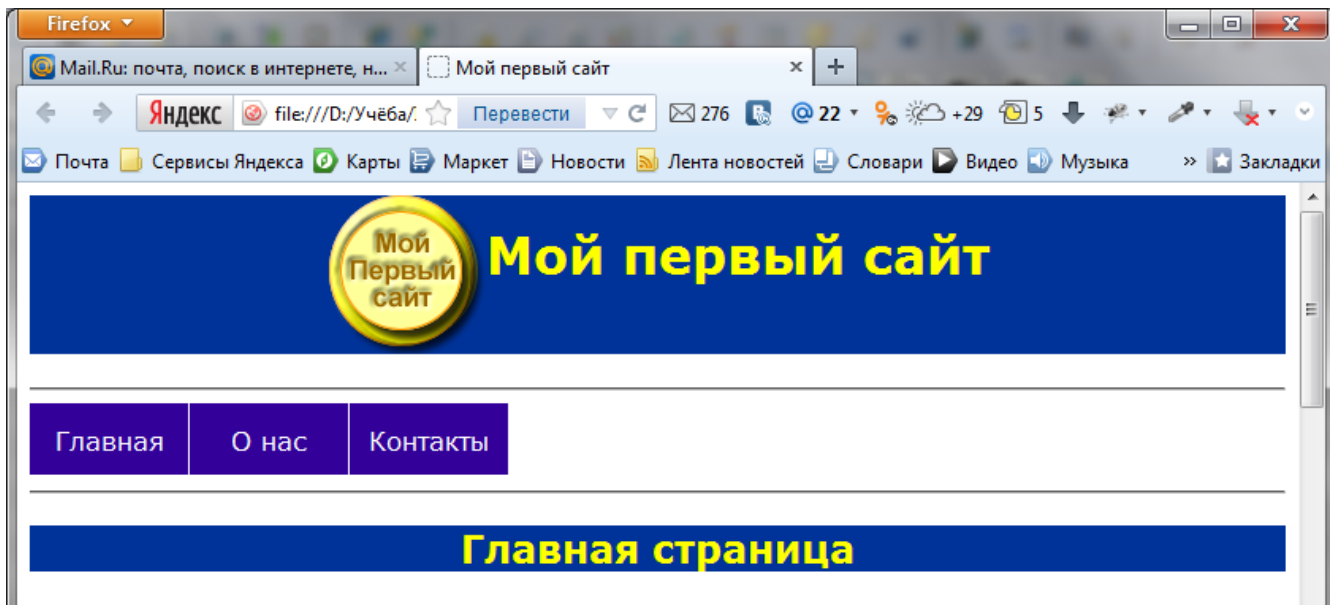
После верхнего меню у нас следует прямая линия `<hr>`. Используем этот тег для очистки рядности.

```
<hr style="clear:both;">
```

Обратите внимание, это как раз тот самый третий вариант описания стилей. Непосредственно в теге, с помощью атрибута style. Декларации CSS пишутся аналогично, как в фигурных скобках, разделяясь точкой с запятой.

В данном случае с помощью оператора clear мы командой both отменили рядность, которая была задана свойством float:left.

Вот что у нас должно получиться:



В результате у нас получились кнопки меню в виде ровненьких прямоугольничков, отстоящие друг от друга на расстоянии 1 px. Это расстояние было задано свойством `margin-right: 1px`.

Свойство `:hover`

Теперь оживим наши кнопки. Сделаем так, чтобы при наведении курсора мыши на кнопку, она меняла цвет фона, цвет текста и к тому же отбрасывала тень, что придаст ей эффект, как будто она приподнимается над страницей.

Для этого создадим отдельный набор CSS инструкций для нашего класса `tophref`, придав ему дополнительное свойство `:hover`.

```
.tophref:hover {  
}
```

Сюда запишем свойства, которые кнопка будет принимать при наведении на нее мышкой.

```
.tophref:hover {  
    color:#CF0; /*Новый цвет текста*/  
    background-color:#06F; /*Новый цвет фона*/  
    /*Тень от кнопки*/  
    box-shadow: 7px 7px 5px rgba(0,0,0,0.6);  
    -moz-box-shadow: 7px 7px 5px rgba(0,0,0,0.6);  
    -ms-box-shadow: 7px 7px 5px rgba(0,0,0,0.6);  
}
```

```
-o-box-shadow: 7px 7px 5px rgba(0,0,0,0.6);  
-webkit-box-shadow: 7px 7px 5px rgba(0,0,0,0.6);  
-khtml-box-shadow: 7px 7px 5px rgba(0,0,0,0.6);  
}
```

Теперь при наведении курсора мыши на кнопку меню, она меняет цвет текста, цвет фона и отбрасывает тень.

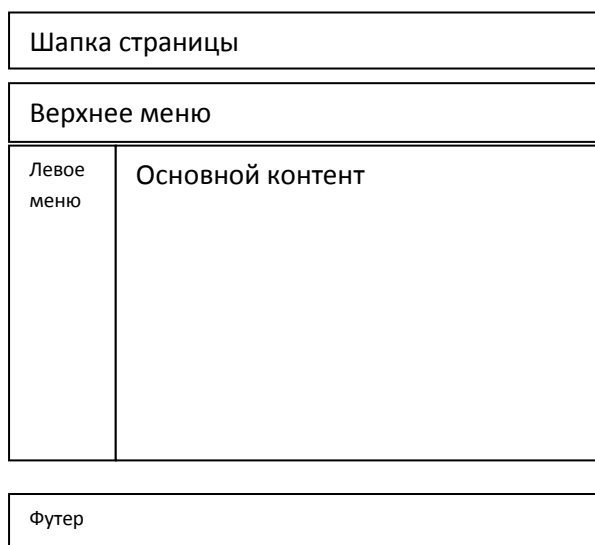
Атрибут id

Атрибуту id также присваиваются параметры (имена), а именам присваиваются свойства CSS. Отличие id от class заключается в том, что имя класса на странице может встречаться многократно, как в примере с меню, а имя id должно встречаться только один раз и быть уникальным.

Часто атрибут id с уникальными параметрами (именами) присваивают элементам блочной верстки, тегам <div>. При этом стилями CSS описывают их размеры, внешний вид и положение на странице. При написании стиля для параметра атрибута id в начале его имени ставят знак #.

```
#header {  
}
```

Теперь рассмотрим применение атрибута id на практике. Нам необходимо сверстать веб-страницу с помощью контейнеров <div>. При этом мы планируем нашу страницу следующим образом:



Каждый элемент нашего плана создается с помощью тега <div> с атрибутом id. Всю эту конструкцию мы поместим в контейнер, чтобы удобнее было позиционировать ее на странице. Создадим новую страницу div.html. В разделе head этой страницы присоединим с помощью тега link файл со стилями и в разделе < body> создадим следующую конструкцию:

```
<!--Контейнер страницы -->  
<div id="container">  
  
<!--Шапка страницы-->  
  <div id="header">  
  </div>  
<!--Верхнее меню -->  
  <div id="topmenu">
```

```
    </div>
<!--Левое меню -->
    <div id="leftmenu">
    </div>
<!--Основной контент -->
    <div id="maincontent">
    </div>
<!--Футер -->
    <div id="footer">
    </div>
</div>
```

Теперь пропишем стили в файле styles.css.

```
body {
    font: 10pt Arial, Helvetica, sans-serif; /* Шрифт на веб-странице */
    background: #e1dfb9; /* Цвет фона */
}
#container {
    width: 800px; /* Ширина слоя */
    margin: 0 auto; /* Выравнивание по центру */
    background: #f0f0f0; /* Цвет фона */
}
#header {

    text-align: center; /* Выравнивание по центру */
    padding: 0; /* Отступы вокруг текста */

}
#topmenu{
    background:#FFFF99;

}
#leftmenu {
    margin-top: 10px;
    width: 110px; /* Ширина слоя */
    padding: 0 10px; /* Отступы вокруг текста */
    float: left; /* Обтекание по правому краю */
}
#maincontent {
    margin-left: 130px; /* Отступ слева */
    padding: 20px; /* Поля вокруг текста */
    background: #fff; /* Цвет фона правой колонки */
}
#footer {
    background: #8fa09b; /* Цвет фона подвала */

    padding: 5px; /* Отступы вокруг текста */
    clear: left; /* Отменяем действие float */
}
#footer p {
    text-align:center;
    font-size:small;
    color: #fff; /* Цвет текста */
}
h1,h2,h3 {
    color:yellow;
    font-family:Verdana, Geneva, sans-serif;
    background:#039;
    text-align:center
}
p {
    color: black;
    font-family:Verdana, Arial, Helvetica, sans-serif;
    font-size:medium;
    text-align:justify;
}
/*p:first-letter {font-size:50px; float:left; color: #000; font-weight:bold}*/

/*Стиль кнопок меню*/
.tophref {
    text-decoration: none;
    display: block;
    float: left;
    width: 100px;
    height: 30px;
    background-color:#309;
```



```

    color: #FFF;

    margin-right: 1px;
    margin-bottom: 10px;
    vertical-align: middle;
    font-family: Verdana, Geneva, sans-serif;

    text-align: center;
    padding-top: 15px;
}
/*Стиль кнопок меню при наведении мышью*/
.tophref:hover {
    color:#CF0; /*Цвет текста*/
    background-color:#06F;

    /*Тень от кнопочки*/
    -moz-box-shadow: 7px 7px 5px rgba(0,0,0,0.6);
    -ms-box-shadow: 7px 7px 5px rgba(0,0,0,0.6);
    -o-box-shadow: 7px 7px 5px rgba(0,0,0,0.6);
    -webkit-box-shadow: 7px 7px 5px rgba(0,0,0,0.6);
    -khtml-box-shadow: 7px 7px 5px rgba(0,0,0,0.6);
    box-shadow: 7px 7px 5px rgba(0,0,0,0.6);
}
/*Класс для отмены действия float*/
.clear {
    clear: left; /* Отменяем действие float */
}

```

Теперь осталось скопировать контент из index.html в соответствующие блоки нашей новой страницы.

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
<title>Дивная верстка</title>
<link rel="stylesheet" href="css/styles.css">
</head>

<body>
<!--Контейнер страницы -->
<div id="container">

<!--Шапка страницы-->
    <div id="header">
        <h1><a href="index.html"></a>Мой первый сайт
</h1>
    </div>
<!--Верхнее меню -->
    <div id="topmenu">
        <a href="index.html" class="tophref">Главная</a>
        <a href="about.html" class="tophref">    О нас</a>
        <a href="contacts.html" class="tophref"> Контакты</a>
        <div class="clear"></div>
    </div>

<!--Левое меню -->
    <div id="leftmenu">
        <a href="#ch_1">Глава 1</a><br>
        <a href="#ch_2">Глава 2</a><br>
        <a href="#ch_3">Глава 3</a>
    </div>
<!--Основной контент -->
    <div id="maincontent">
        <h2><a name="ch_1"></a>Глава 1</h2>

        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse id dui vel nulla
pellentesque vehicula. Vestibulum tincidunt velit non arcu convallis a sollicitudin mauris
sollicitudin. Curabitur at ipsum justo, sed cursus justo. Proin adipiscing vestibulum justo id
tincidunt. Quisque accumsan sapien aliquet arcu faucibus vitae gravida velit egestas. Aliquam et
sem sem. Donec ut sapien a enim tempus pharetra ut at diam. Maecenas et hendrerit augue. Phasellus

```

pellentesque condimentum mollis. Fusce rhoncus ligula vel lorem pretium sit amet bibendum nibh tempus. Integer eget leo elit, eget rhoncus tellus. Sed suscipit euismod dolor a dictum. Nulla semper tellus quam, vitae laoreet tortor. Aenean eget dui quam. </p>

<p> Vestibulum sodales libero a sem fermentum vitae vehicula mi blandit. Nam convallis est sed felis tempus tincidunt. Quisque iaculis turpis vel urna mattis at interdum metus semper. Aliquam augue dui, ullamcorper vel pharetra quis, aliquam id sem. Praesent sagittis diam a sem hendrerit interdum. Sed fringilla pharetra ornare. In molestie cursus fringilla. Proin dapibus magna iaculis velit pellentesque rutrum. Praesent elementum pharetra odio, quis interdum eros cursus ut. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. Cras vel ipsum dolor. In sit amet nibh sem, sit amet scelerisque urna. Phasellus purus velit, malesuada nec interdum sit amet, volutpat ut ante. Suspendisse consequat consequat velit quis hendrerit. Nulla id massa in enim dictum volutpat a auctor urna. </p>

<p> In imperdiet nunc tellus. Donec quis lorem velit, dignissim vehicula enim. Curabitur ultricies posuere mi at dapibus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Curabitur ut neque libero. Nam sed odio a lorem vulputate ultrices a in risus. Pellentesque est justo, varius sit amet congue in, volutpat gravida arcu. Etiam quis bibendum erat. Etiam tempus sapien eu felis scelerisque fringilla. Nulla malesuada leo nec justo lobortis eget dignissim odio fermentum. Mauris condimentum lacinia neque at vulputate. Nam risus dui, laoreet non imperdiet eu, vehicula eget lectus. Curabitur tristisque nisi et turpis tristisque vel dapibus neque rutrum. Phasellus non turpis a eros euismod ultricies. Pellentesque eu porta sem. Fusce in erat erat. </p>

<h2>Глава 2</h2>

<p> Sed eros metus, fringilla a bibendum eu, porta ut felis. Vivamus placerat, neque et egestas elementum, massa elit sodales est, at feugiat neque metus non elit. Donec eu iaculis erat. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Sed sit amet dapibus lectus. Praesent et nisl purus. Donec sagittis fermentum bibendum. Integer vel lectus magna. Mauris gravida risus vitae justo dictum ut tincidunt risus commodo. Mauris porta vestibulum turpis. Proin ut convallis nunc. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque commodo pharetra libero eu ultricies. Vestibulum venenatis felis non leo tempor dapibus. Integer euismod, massa sed congue eleifend, augue diam pretium nisi, et suscipit purus ante eget massa. Aliquam id nisi quis dui viverra facilisis. </p>

<p> Sed pulvinar iaculis ante quis feugiat. Mauris mollis iaculis convallis. Nam sodales accumsan mi eget malesuada. Etiam molestie facilisis eros et semper. Aliquam interdum faucibus sapien id gravida. Phasellus massa tellus, volutpat tristisque hendrerit dignissim, mattis sit amet tellus. Nunc tempor fermentum risus, a auctor urna ultrices ac. Curabitur ac magna justo, non tristisque dolor. Nulla consectetur nibh sed enim pretium ut sodales mauris tempus. Duis pellentesque ante vitae libero viverra vitae laoreet leo mollis. </p>

<p> Pellentesque dapibus, eros at accumsan consectetur, quam nisl porta turpis, in imperdiet elit mi in leo. Ut libero eros, euismod in placerat in, commodo vel nisi. Vestibulum at sapien enim. Donec id porta odio. Curabitur odio magna, pellentesque in rutrum sit amet, malesuada at dui. Nulla mollis tincidunt leo, sed imperdiet metus dictum at. Donec faucibus convallis est non condimentum. Mauris quis rutrum orci. Mauris varius velit porttitor est ultrices pellentesque. Vivamus accumsan consequat fringilla. Sed eget purus sit amet sapien tempus adipiscing et suscipit eros. </p>

<p> Praesent in libero vel nisl placerat laoreet. Curabitur enim mi, vestibulum in malesuada quis, mollis egestas augue. Phasellus bibendum ultrices nunc nec varius. Nam in nisi tellus. Curabitur tincidunt quam fringilla quam sollicitudin viverra. Nulla a nunc volutpat erat laoreet aliquet id vel ante. Etiam viverra, nunc ut aliquet imperdiet, nunc nisl tristisque lacus, sit amet pulvinar dui leo et leo. Fusce placerat convallis ante a luctus. Nullam at ante felis. </p>

<p> Pellentesque tristique scelerisque tincidunt. Phasellus consectetur, orci ullamcorper convallis lacinia, turpis orci blandit libero, quis congue velit arcu at mi. Ut urna elit, dictum sit amet tincidunt ut, rutrum tincidunt mauris. Nulla venenatis adipiscing justo, nec tincidunt neque pretium ac. Curabitur in mauris neque. Aenean felis diam, fringilla nec facilisis ut, egestas et risus. Aenean dignissim hendrerit iaculis. Suspendisse lacus lectus, dapibus eget aliquam at, luctus ut lorem. Maecenas et arcu orci, ut tristique tortor. Fusce interdum nibh quam. Etiam elit est, sagittis vel facilisis eu, interdum consectetur nisl. Aliquam et eros elit, eleifend mollis metus. </p>

<h2>Глава 3</h2>

<p> Pellentesque pulvinar, dolor non hendrerit tristisque, augue nibh vulputate tortor, eget eleifend lacus magna at lorem. Morbi leo sapien, tempor at accumsan sed, vestibulum eget nisi. Morbi et nisi lorem, a ultricies nisl. Proin est ipsum, luctus et semper et, rutrum id mi. Proin odio nunc, consectetur a faucibus quis, iaculis et diam. Praesent nec nunc non enim tempus faucibus ut et libero. Mauris eget commodo massa. Vivamus sodales laoreet enim, sed porttitor quam scelerisque hendrerit. Nam porta, purus sit amet semper varius, ipsum est consequat augue, a tristisque nibh massa id urna. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Morbi posuere suscipit velit, in ultricies orci euismod et. Cras nec nulla turpis, eget luctus est. Curabitur lacinia consectetur tortor, at vestibulum nisi mollis ac. Sed risus tortor, rutrum eget rhoncus ac, auctor at elit. </p>

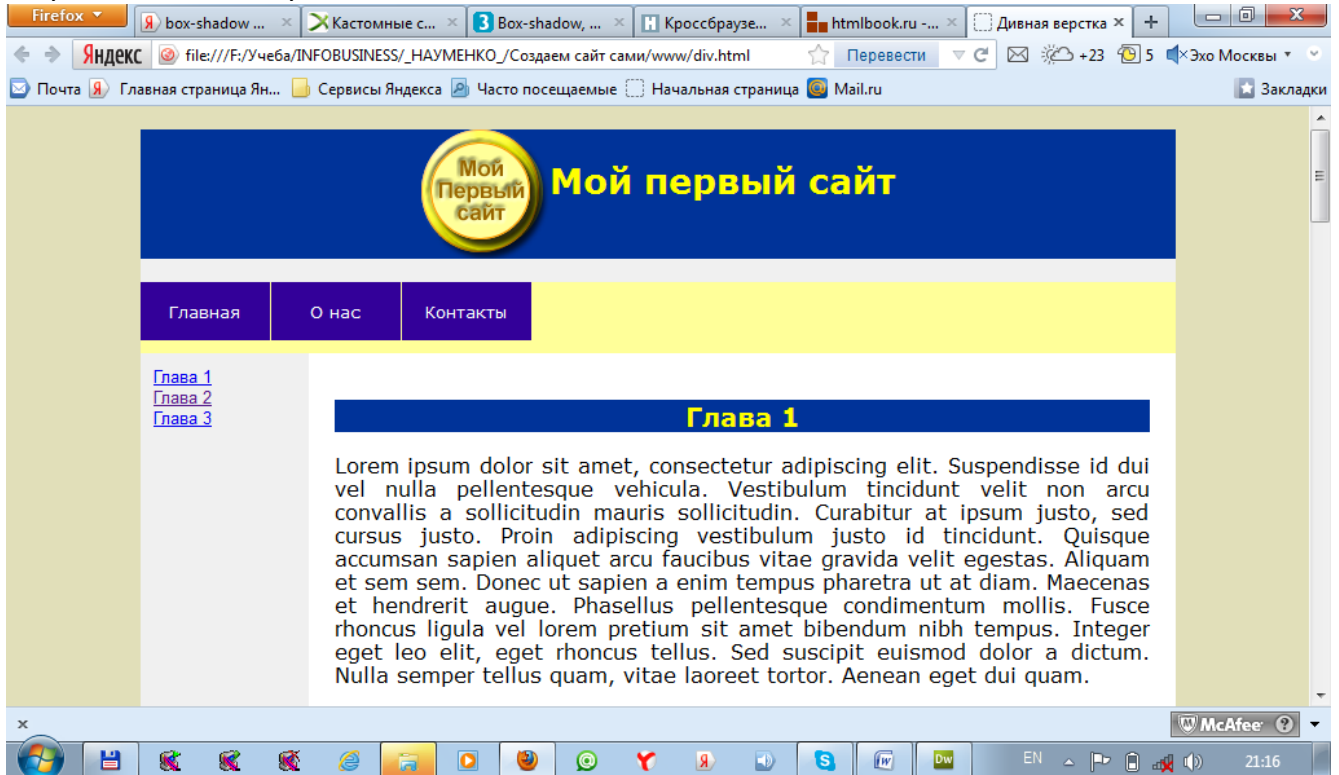
<p> Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Nullam magna metus, consequat ut tempus a, faucibus sit amet orci. Duis in auctor ante. Nulla vehicula sapien elit, quis porta diam. Curabitur vitae dui libero, in semper risus. Donec malesuada risus sed nisi convallis sit amet iaculis nisi sollicitudin. Sed luctus pretium est varius mattis. Nam facilisis dui pulvinar elit hendrerit congue. In purus est, lacinia non ullamcorper non, imperdiet vitae lacus. In hac habitasse platea dictumst. Nunc leo tellus, tincidunt nec eleifend vel, ornare id nisi. Curabitur sit amet neque et magna suscipit lacinia. Vestibulum ipsum tortor, rutrum ac scelerisque at, tristisque a erat. Praesent sit amet ipsum ut metus ultricies auctor non non lacus. </p>

</div>

<!--Футер -->

```
<div id="footer">
  <p> &copy; Сайт своими руками </p>
</div>
</body>
</html>
```

Результат соответствует нашим ожиданиям.



Урок №9

Углубление в CSS

Сегодня мы углубимся в некоторые важные детали, которые необходимо знать при использовании CSS стилей. И начнем мы с единиц длины.

Единицы длины

Мир WEB-дизайна это особый виртуальный мир и поэтому единицы длины, которые используются в WEB-дизайне наряду с традиционными, тоже встречаются необычные.

Когда мы изучали с вами HTML, мы использовали всего две единицы длины это px (пиксель) и % (процент). Мы могли задать, например, ширину таблицы строго в пикселях `<table width="800">` или в процентах относительно родительского контейнера, в который помещена таблица. Если таблица создана самостоятельно, то она может быть задана относительно окна браузера. `<table width="100%">`

В CSS наряду с пикселями и процентами присутствуют и другие единицы длины:

- **in** -- дюйм (примерно 2,54 см);
- **cm** -- сантиметр;
- **mm** -- миллиметр.
- **pt** -- пункт (1/72 дюйма);
- **pc** -- пика (1/6 дюйма или 12 пунктов);
- **ex** -- высота строчной буквы "x" шрифта, эта величина не имеет названия.
- **em** -- единица, вычисляемая относительно размера шрифта элемента;
- **px** -- пиксель; его размер зависит от устройства отображения (монитора, экрана мобильного телефона или карманного компьютера);
- **%** -- процент; вычисляется относительно размера элемента-предка.

Единицы длины, знакомые нам из повседневной жизни, такие, например, как сантиметр не всегда подходят для отображения на мониторе, потому что существуют технические ограничения, связанные с характеристиками различных мониторов. Сантиметр на мониторах с маленьким разрешением будет отображаться невообразимо крупно, а на мониторах с большим разрешением – вроде как нормально. Если при описании стиля web-документа мы будем пользоваться единицами измерения физического мира, например дюймами, то не возможно будет предсказать, какой действительный размер отобразится на экране конкретного пользователя.

Размер пикселя на самом деле тоже зависит от разрешения монитора и может принимать значения от 0,25 до 0,35 мм.

Если мы при верстке веб-страницы задаем жесткий размер контейнера, тогда необходимо пользоваться пикселями. Если мы планируем резиновую верстку, то необходимо пользоваться процентами. Как раз для этого случая для обозначения размера шрифтов подходит единица **em**. Величина **em** незаменима для пропорционального увеличения размера текста, высоты

строки, межсимвольных интервалов и расстояний между словами. Если необходимо на 20% увеличить размер шрифта по сравнению с базовым, то достаточно указать размер 1.2 em. И каким бы ни был базовый размер шрифта, новый размер будет высчитан корректно. Такого нельзя добиться, используя типографские единицы измерения, из-за погрешностей перевода их в пиксели и неопределенности в параметрах устройства вывода текста.

Псевдоэлемент :first letter

Псевдоэлемент :first-letter определяет стиль первого символа в тексте элемента, к которому добавляется. В описании стиля псевдоэлемента можно задавать свойства шрифта, поля, отступы, границы, цвет и фон. Практическое применение – создание «Буквицы» без использования графики.

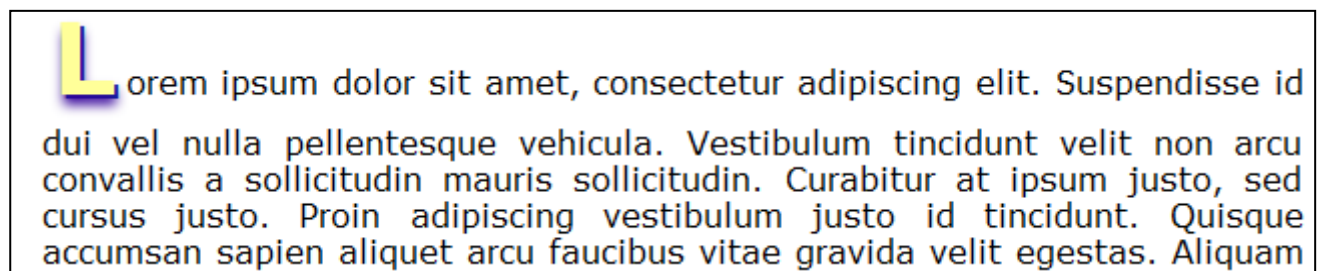
```
p:first-letter {
font-size:50px;
color: #FF9;
font-weight:bold;/*толщина буквицы*/
text-shadow: 1px 5px 5px #309, 2px 2px #039; /*Тень буквицы*/
}
```

При таком задании стилей каждый параграф будет начинаться с Буквицы, что не всегда оправдано замыслом. Для того чтобы с буквицы начинался только желаемый параграф, можно создать «пустой» класс и присоединить ему псевдоэлемент :first-letter.

```
.first:first-letter {
font-size:50px;
padding:5px;
color: #FF9;
font-weight:bold;
text-shadow: 1px 5px 5px #309, 2px 2px #039;
}
```

В данном примере мы создали класс .first и присоединили к нему псевдоэлемент :first-letter. Теперь мы можем присваивать класс .first к нужному нам параграфу, который будет начинаться с Буквицы.

```
<p class="first">Lorem ipsum...</p>
```



Псевдоэлемент :first-line

Псевдоэлемент :first-line задает стиль первой строки форматированного текста. В описании стиля используются свойства отображения шрифтов, изменения цвета текста и фона.

```
.first:first-line {
background-color:#FF0;
font-variant:small-caps;/*Отображение строчными буквами*/
}
```

В данном примере первая строка параграфа с классом first будет отображаться строчными буквами на желтом фоне.

LOREM IPSUM DOLOR SIT AMET, CONSECTETUR ADIPISCING ELIT. SUSPENDISSE ID DUI
vel nulla pellentesque vehicula. Vestibulum tincidunt velit non arcu
convallis a sollicitudin mauris sollicitudin. Curabitur at ipsum justo, sed
cursus justo. Proin adipiscing vestibulum justo id tincidunt. Quisque
accumsan sapien aliquet arcu faucibus vitae gravida velit egestas. Aliquam

Свойство `font-family`

Одним из технических ограничений современного web-дизайна является разнообразие установленных шрифтов на компьютерах пользователей и возможность установки шрифтов по умолчанию в браузерах пользователей. Это ограничение не дает гарантии точного отображения шрифта, который задумали WEB-дизайнеры. Особенно это касается всяких экзотических шрифтов. Однако во всех компьютерах всегда присутствуют стандартные шрифты, которые и рекомендуется всегда использовать WEB-дизайнерам.

- serif -- шрифты с засечками (антиквенные), типа Times;
- sans-serif -- рубленые шрифты (шрифты без засечек или гротески), типичный представитель -- Arial;
- cursive -- курсивные шрифты;
- fantasy -- декоративные шрифты;
- monospace -- моноширинные шрифты, ширина каждого символа в таком семействе одинакова (шрифт Courier).

В CSS стилях можно устанавливать семейства шрифтов, сходных по начертанию, например:

```
font-family: Verdana, Geneva, sans-serif;
```

При этом самый «экзотический» шрифт ставится первым, а стандартный – последним. Браузер начинает отыскивать шрифты на компьютере в том же порядке и если не окажется нужного шрифта из семейства, устанавливает стандартный.

Свойство `font-size`

Определяет размер шрифта элемента. Размер шрифта можно устанавливать с помощью единиц длины.

```
font-size: 14px;
```

Размер шрифта еще можно устанавливать с помощью абсолютных и относительных констант, а также с помощью абсолютных и относительных значений единиц размеров. Абсолютные константы:

- xx-small,
- x-small,
- small,
- medium,

- large,
- x-large,
- xx-large

Не смотря на название «Абсолютные», размер этих констант сильно зависит от свойств и настроек браузера и операционной системы.

Относительные константы:

- larger,
- smaller

Относительные константы наследуют размер от родительского элемента, поэтому размер шрифта сильно зависит от значения свойства font-size у родителя элемента.

Свойство font-style

Определяет начертание шрифта - обычное, курсивное или наклонное. Когда для текста установлено курсивное или наклонное начертание, браузер обращается к системе для поиска подходящего шрифта. Если заданный шрифт не найден, браузер использует специальный алгоритм для имитации нужного вида текста. Результат и качество при этом могут получиться непредсказуемыми.

- normal (Обычное начертание текста)
- italic (Курсивное начертание)
- oblique (Наклонное начертание)
- inherit (Наследует значение родителя)

Курсив и наклонный шрифт это не одно и то же. Курсив это специальный шрифт, который имитирует рукописный. Наклонный шрифт формируется путем наклона буквенных знаков вправо.

```
font-style: italic;
```

Свойство font-variant

Определяет, как нужно представлять строчные буквы - оставить их без модификаций или делать их все прописными уменьшенного размера. Такой способ изменения символов называется капителью.

- normal оставляет регистр символов исходным, заданным по умолчанию;
- small-caps модифицирует все строчные символы как заглавные уменьшенного размера;
- inherit наследует значение родителя

```
.first:first-line {  
    background-color:#FF0;  
    font-variant:small-caps;  
}
```




ESTIBULUM SODALES LIBERO A SEM FERMENTUM VITAE VEHICULA MI BLANDIT. NAM

convallis est sed felis tempus tincidunt. Quisque iaculis turpis vel urna mattis at interdum metus semper. Aliquam augue dui, ullamcorper vel pharetra quis, aliquam id sem. Praesent sagittis diam a sem hendrerit

На этом примере первая строка абзаца с классом `.first` отображается в виде модифицированных строчных символов в заглавные уменьшенного размера.

Работа с блоками

Современный дизайн сайта, как мы с вами видели на предыдущем уроке, в основном состоит из блоков, которые мы размещаем на странице так, как нам нужно. В качестве блоков могут выступать все блочные теги, такие как `h`, `p`, `div`, `ul`, `li` и другие. Они называются блочными, потому что каждая конструкция, состоящая из открывающего и закрывающего тега, отображается одна под другой. И все они вложены в глобальный блочный элемент под названием `body`.

С помощью CSS мы можем задавать размеры блоков, располагать блоки не один под другим, а в один ряд, задавать форму блоков, скругляя углы, задавать цвет фона блока и фоновое изображение, устанавливать отступы от краев блока до его содержимого, создавать рамки блоков и даже тени.

Все эти действия с блоками имеют практическое применение при построении сайтов.

Фоновое изображение

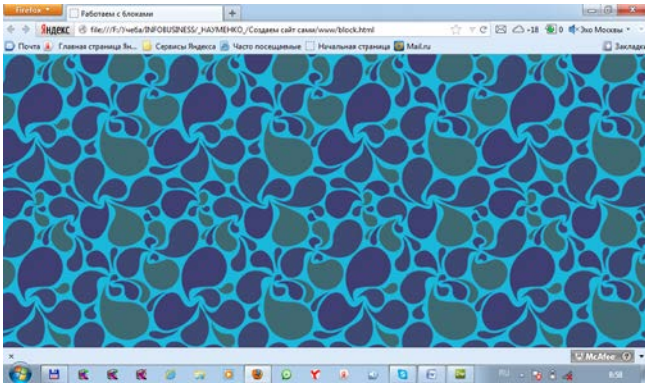
Применение фоновых изображений позволяет придавать web-страницам практически любой вид, задуманный автором.

Вот пример фона страниц, которые можно применить к тегу `body`.



```
body {  
    background-color: #FFFFFF;  
    background-image: url(images/bg_blue.png);  
}
```

Результат будет следующим:

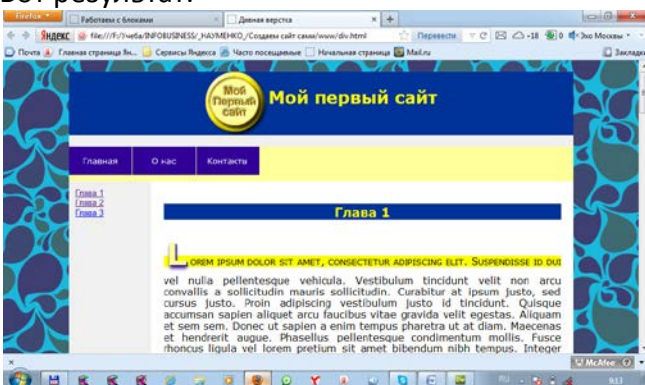


Весь блок body заполнился повторяющимся рисунком `bg_blue.png` из папки `images`. Используем этот прием для украшения нашего сайта. Для этого в файл `styles.css` дескриптору `body` добавим аналогичное свойство:

```
body {  
    font: 10pt Arial, Helvetica, sans-serif; /* Шрифт на веб-  
    странице */  
    background: #e1dfb9; /* Цвет фона */  
    background-image: url(../images/bg_blue.png); /*Фоновое  
    изображение*/  
}
```

Обратите внимание, в файле со стилями путь к изображению фона нужно определять относительно расположения файла со стилями, а не файла самой страницы, поэтому мы добавляем вначале путь выше папки со стилями: «../».

Вот результат:



Фон можно сделать неподвижным, тогда он не будет прокручиваться вместе с содержимым страницы:

```
body {  
    font: 10pt Arial, Helvetica, sans-serif;  
    background-color: #e1dfb9; /* Цвет фона */  
    background-image: url(../images/bg_blue.png); /*Фоновое изображение*/  
    background-attachment: fixed; /*Фиксация фона*/  
}
```

Округление вершин блоков

Все блочные элементы по умолчанию являются прямоугольниками. С помощью стилей CSS с недавних пор стало возможным скруглять прямые углы вершин блоков, не прибегая к

использованию фоновых изображений. Раньше, когда такой возможности не существовало, приходилось рисовать скругленный блок в графическом редакторе, а потом делать его фоном для блокового элемента.

Часто поступали так: Рисовались все скругленные уголки, строилась таблица, и эти уголки вставлялись в ячейки, расположенные по углам, в результате получалась цельная картина скругленного блока.



Сейчас необходимость использования графики отпала. Все можно сделать с помощью CSS.

```
.skrugl {
  color: #930;
  background-color: #FFF;
  height: 100px;
  width: 200px;
  position: relative; /*выдергиваем блок из потока*/
  display: table; /*блок будет отображаться как табличный*/
  /*Закругление*/
  border-radius: 20px ;
  -moz-border-radius:20px;
  -ms-border-radius: 20px ;
  -o-border-radius: 20px ;
  -webkit-border-radius: 20px;
  -khtml-border-radius: 20px;
}
.skrugl p {
  display: table-cell; /*абзац будет отображаться как ячейка таблицы (TD)*/
  vertical-align: middle; /*выравниваем абзац по вертикали*/
  text-align: center; /*выравниваем абзац по горизонтали*/
}
```

HTML код будет выглядеть так:

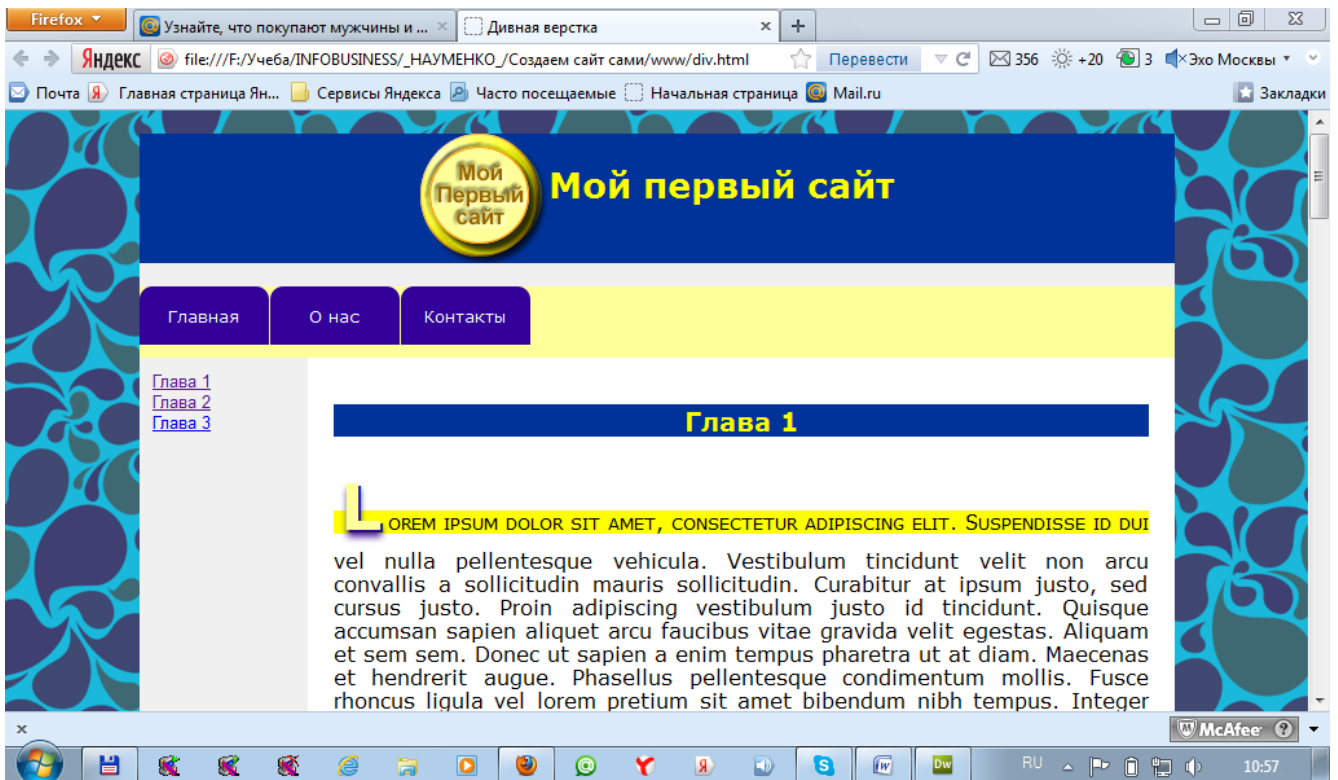
```
<div class="skrugl">
  <p>Скругленный блок</p>
</div>
```



Применим это к кнопочкам меню для нашей WEB-страницы. Для этого в файле styles.css для класса .tophref допишем свойства для закругления:

```
.tophref {
    text-decoration: none;
    display: block;
    float: left;
    width: 100px;
    height: 30px;
    color: #FFF;
    margin-right: 1px;
    margin-bottom: 10px;
    vertical-align: middle;
    font-family: Verdana, Geneva, sans-serif;
    text-align: center;
    padding-top: 15px;
    /*Закругление*/
    border-radius: 10px 10px 0 0;
    -moz-border-radius:10px 10px 0 0;
    -ms-border-radius: 10px 10px 0 0;
    -o-border-radius: 10px 10px 0 0;
    -webkit-border-radius: 10px 10px 0 0;
    -khtml-border-radius: 10px 10px 0 0;
}
```

Вот результат:



При желании свойство округления можно применить и к другим блокам.

Урок №10

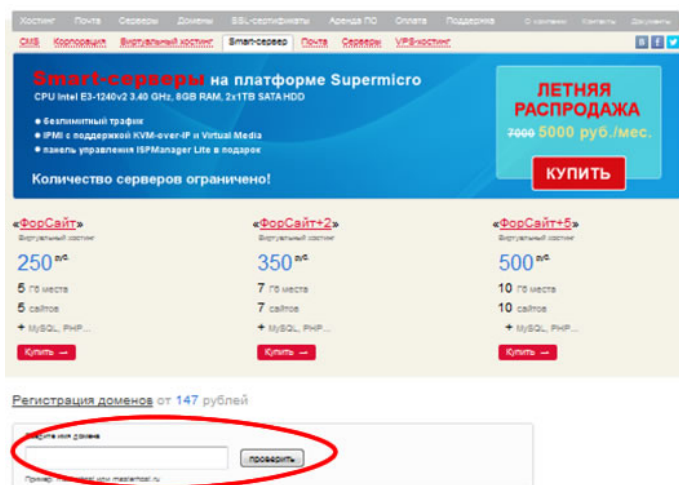
Публикация сайта в интернете

В жизни каждого сайтостроителя возникает счастливый момент, когда сайт уже готов и его необходимо опубликовать в интернете.

Для этого целесообразно воспользоваться услугами хостинговой компании. Компанией номер один в России, которая предоставляет услуги хостинга, а также регистрирует доменные имена второго уровня, безусловно, является компания Мастерхост <http://masterhost.ru>. Я пользуюсь ее услугами постоянно на протяжении десяти лет. Регистрирую домены для себя и своих клиентов, а также регистрирую хостинговые площадки для своих сайтов и сайтов моих клиентов. Мастерхост – надежный партнер. У них платный качественный хостинг. За 10 лет работы у меня к ним не было никаких претензий.

Но давайте все по порядку.

Первое, что необходимо сделать, это зарегистрировать домен. Для этого сначала нужно придумать его имя и проверить, свободно ли оно. Чтобы проверить свободно ли доменное имя, можно зайти на сайт Мастерхоста по ссылке <http://masterhost.ru>, ввести имя домена в соответствующее окошко и нажать кнопку.



Если домен свободен, скорее регистрируйте его.

Для регистрации домена вам необходимо стать клиентом компании Мастерхост. Пройдите по ссылке «регистрация» и зарегистрируйтесь как физическое лицо, заполнив анкету.

The screenshot shows the Masterhost website interface. At the top left is the Masterhost logo with the tagline "PROFESSIONAL HOSTING". To its right are navigation links: "Магазин", "Регистрация" (circled in red), and "Панель управления". A search box is visible on the far right. Below the navigation is a horizontal menu with categories: "Хостинг", "Почта", "Серверы", "Домены", "SSL-сертификаты", "Аренда ПО", "Оплата", "Поддержка", and "О ком". A secondary menu lists services: "CMS", "Корпорация", "Виртуальный хостинг", "Smart-сервер", "Почта", "Серверы", and "VPS-хостинг". The main content area features a blue banner for "Smart-серверы на платформе Supermicro" with specifications: "CPU Intel E3-1240v2 3.40 GHz, 8GB RAM, 2x1TB SATA HDD". It lists features: "Безлимитный трафик", "IPMI с поддержкой KVM-over-IP и Virtual Media", and "панель управления ISPManager Lite в подарок". A note states "Количество серверов ограничено!". Below the banner are three pricing cards for virtual hosting: "«ФорСайт»" (250 руб., 5 GB space, 5 sites), "«ФорСайт+2»" (350 руб., 7 GB space, 7 sites), and "«ФорСайт+2»" (500 руб., 10 GB space, 10 sites).

The screenshot shows the "Регистрация клиентов" (Client Registration) form. It features the Masterhost logo and a progress indicator with three steps: "1. Юридический статус" (highlighted in red), "2. Информация о клиенте", and "3. Подтверждение". Below the progress bar, a text block explains: "Регистрация необходима для заказа услуг и доступа в панель управления услугами. Процесс регистрации разделён на несколько шагов." The form asks for the user's legal status with three radio button options: "физическое лицо" (selected), "юридическое лицо", and "индивидуальный предприниматель". At the bottom left, there is a link for "Помощь online-консультанта". At the bottom right is a green "Далее" button. The footer contains contact information: "© 1999–2013 ЗАО «Мастерхост»" and phone numbers for Moscow and other regions.

При регистрации используйте реальные данные, чтобы избежать проблем с приобретением услуг и осуществлением платежей. Следуйте инструкциям. Вам придется ввести свои фамилию, имя, отчество, адрес, электронный адрес и паспортные данные. В конце регистрации вам на электронную почту придет письмо с ссылкой, логином и паролем для доступа в контрольную панель управления вашими услугами.

Здравствуйте, ██████████

Вы успешно зарегистрировались в качестве клиента компании .masterhost - <http://masterhost.ru/>.

Login: ██████████
Password: ██████████

В случае изменения контактной информации, необходимо отредактировать ваши данные в разделе "Информация о пользователе" в панели управления URL: <https://cp.masterhost.ru/client>

Ваш персональный rip-код ██████████

Он является по умолчанию ответом на контрольный вопрос, который можно изменить в разделе "Общие услуги" - "параметры голосовой авторизации".

Если во время звонка в службу поддержки ваше обращение будет невозможно обработать оперативно, то после успешной проверки rip-кода, менеджер компании зафиксирует его письменно. Номер, присвоенный запросу, будет отправлен вам по указанному при регистрации электронному адресу.

Ответ будет также направлен вам по электронной почте либо, при желании, сообщен по указанному вами номеру телефона.

Для получения возможности создания виртуальных площадок и регистрации доменов необходимо дождаться решения...

Для того, чтобы зарегистрировать домен и электронную площадку для размещения на ней вашего сайта, вам необходимо будет положить на ваш личный счет необходимую сумму денег.

Вот стоимость регистрации доменов сроком на 1 год:

ЦЕНЫ НА РЕГИСТРАЦИЮ ДОМЕНОВ

Доменная зона	Стоимость регистрации доменов, руб./год	Стоимость продления доменов, руб./год
.ru	147,00	147,00
.ru (поддержка «WebNames.ru», «RU-CENTER»)	нет	413,00
.su	602,00	602,00
русскоязычные домены .rf	147,00	147,00
.rf (регистратор RU-CENTER)	нет	392,00
.tel	686,00	686,00
.com .net .org .info .biz	784,00	784,00
.cc	1568,00	1568,00
цены на все зоны		

Рекомендую регистрировать домен в зоне ru. Крайне не рекомендую регистрировать русскоязычный домен в зоне РФ.

Вот цены на покупку хостинга

Основные тарифы виртуального хостинга

Unix Windows

ФорСайт 5 Гб места на диске 5 сайтов MySQL, PHP, CGI 250 руб. в месяц Купить →	ФорСайт+2 7 Гб места на диске 7 сайтов MySQL, PHP, CGI 350 руб. в месяц Купить →	ФорСайт+5 10 Гб места на диске 10 сайтов MySQL, PHP, CGI 500 руб. в месяц Купить →
---	---	---

Для начала выбираем тариф ФорСайт. Его вам вполне хватит. Кроме того он вам пригодится, когда вы научитесь программировать и пользоваться базами данных. Расчитываем сумму на год: $147+250*12=3147$ рублей. Кладите 3500р. с запасом. Я так всегда делаю.

Для входа в контрольную панель переходим по ссылке <https://cp.masterhost.ru/>

.masterhost

Вход в контрольную панель

логин

пароль [напомнить](#)

Вхожу с чужого компьютера

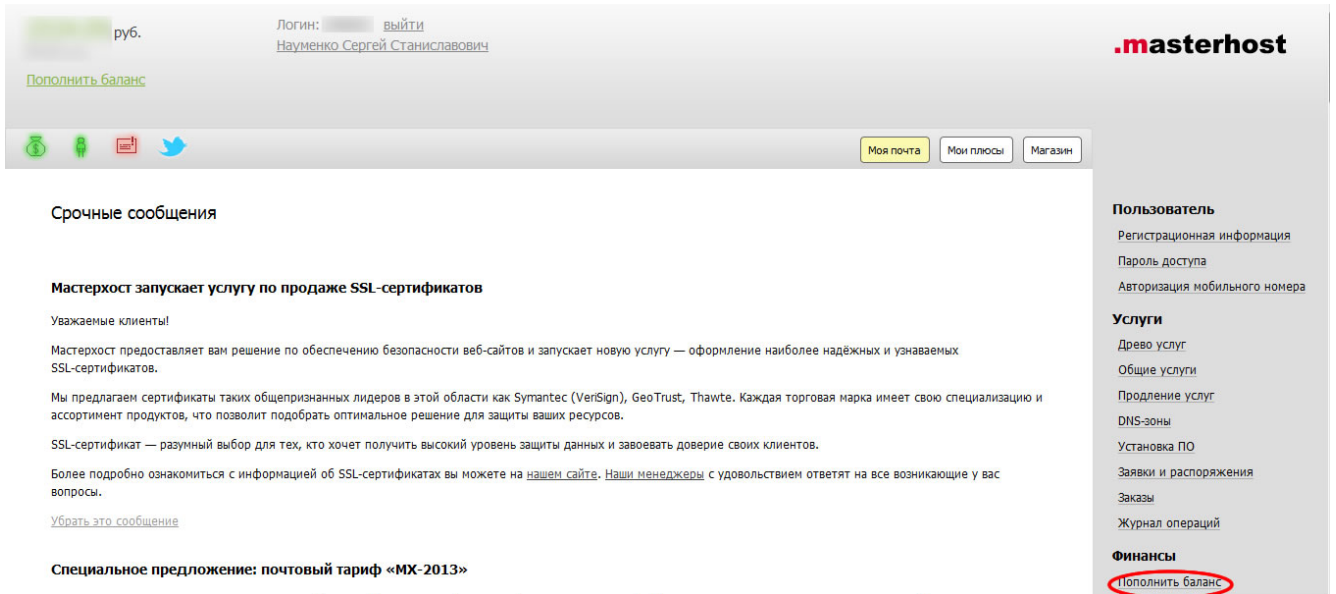
[Зарегистрироваться](#)

© 1999—2013 .masterhost

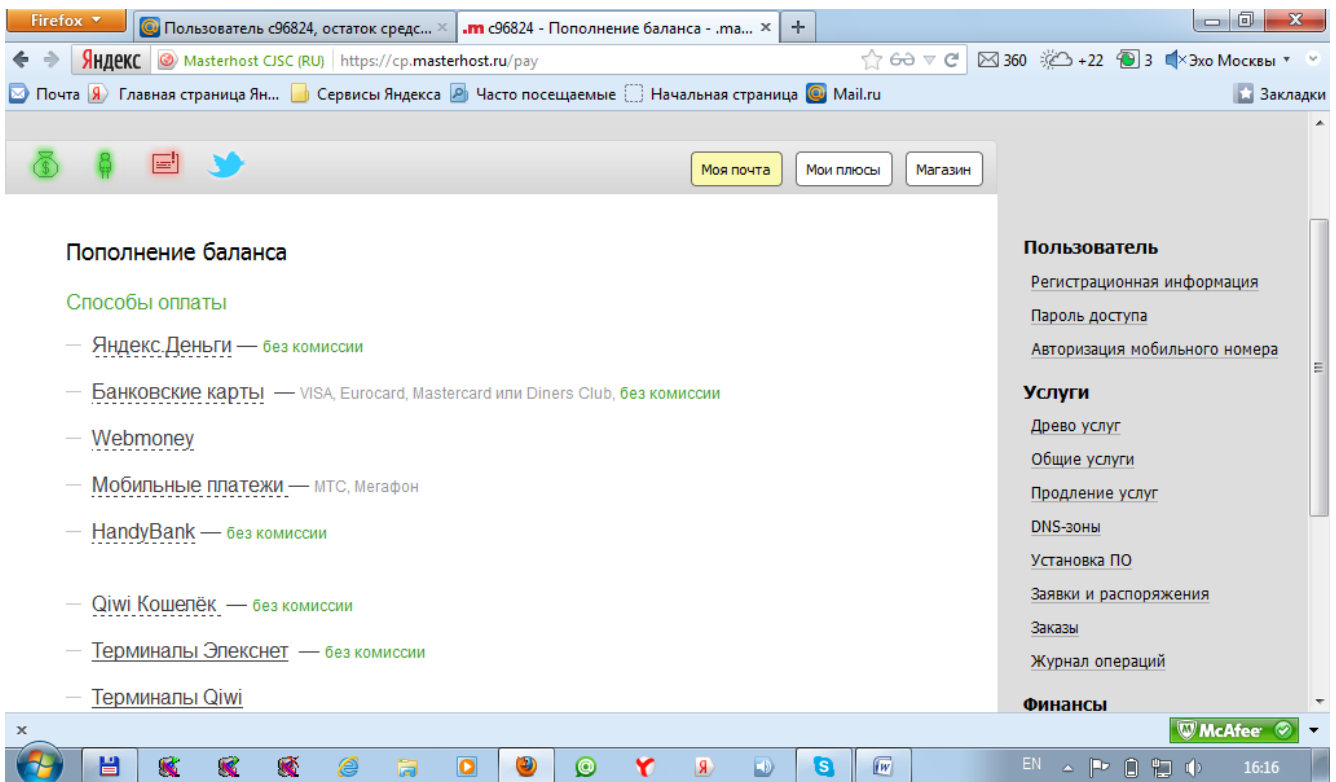
[Контактные телефоны, схема проезда](#)

Техническая помощь: support@masterhost.ru

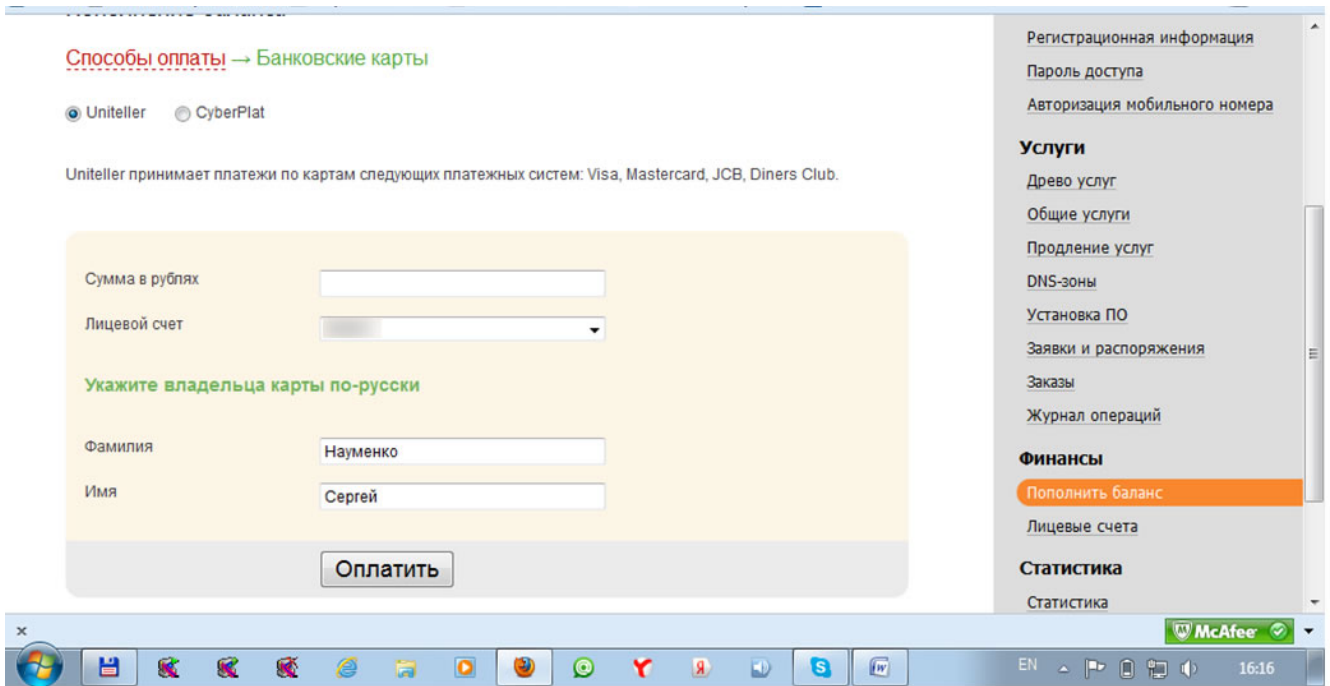
Вводим логин и пароль, которые пришли в письме и нажимаем кнопку «Войти».



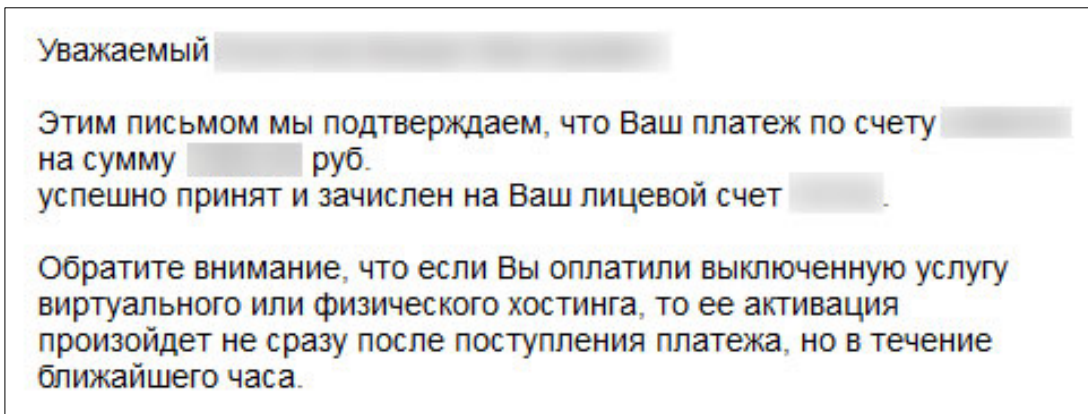
В правом меню выбираем «Пополнить баланс»



Выбираете способ оплаты и кладете деньги на свой лицевой счет.



После зачисления платежа на почту придет информационное письмо.



Далее заходите в магазин и покупаете домен и хостинг.



shop.masterhost.ru

Виртуальный хостинг

Виртуальный хостинг от 75 рублей в месяц

Регистрация доменов

Unix Windows

VPS-хостинг

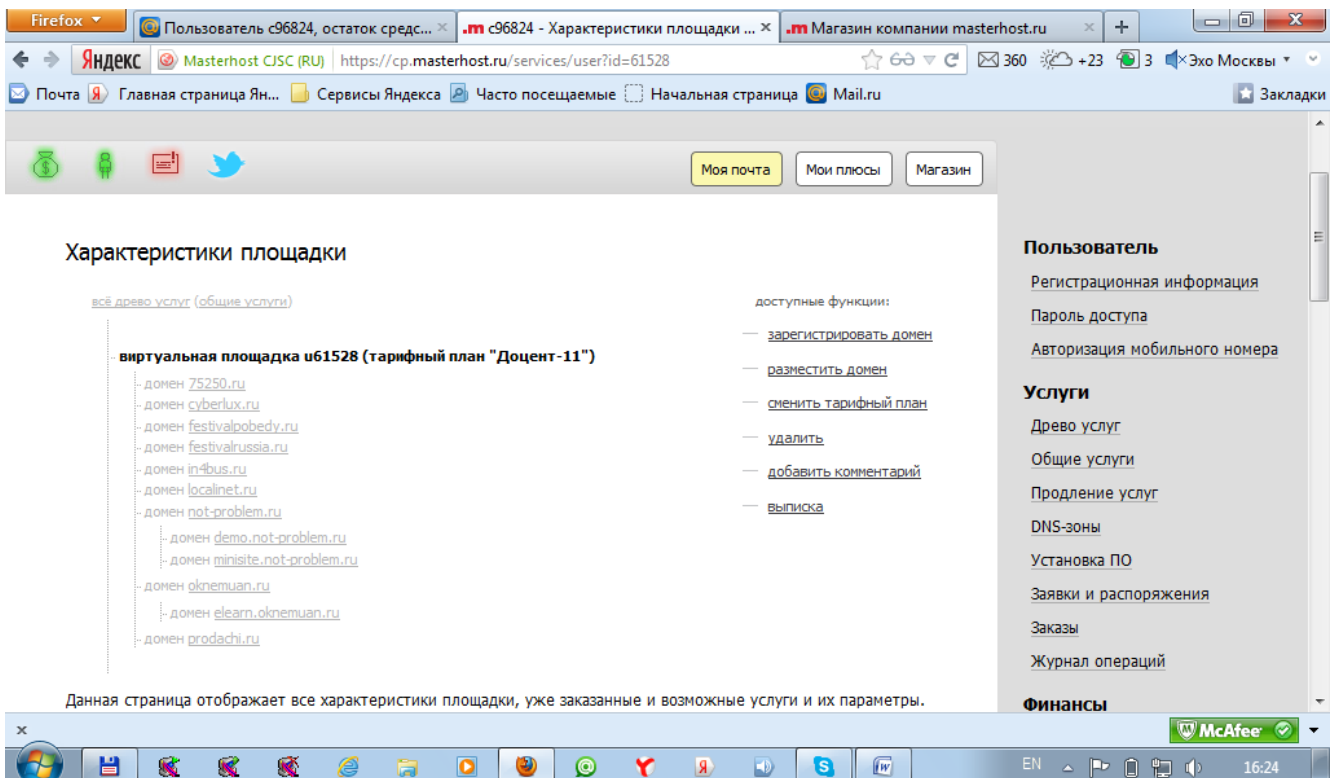


Задать вопрос online

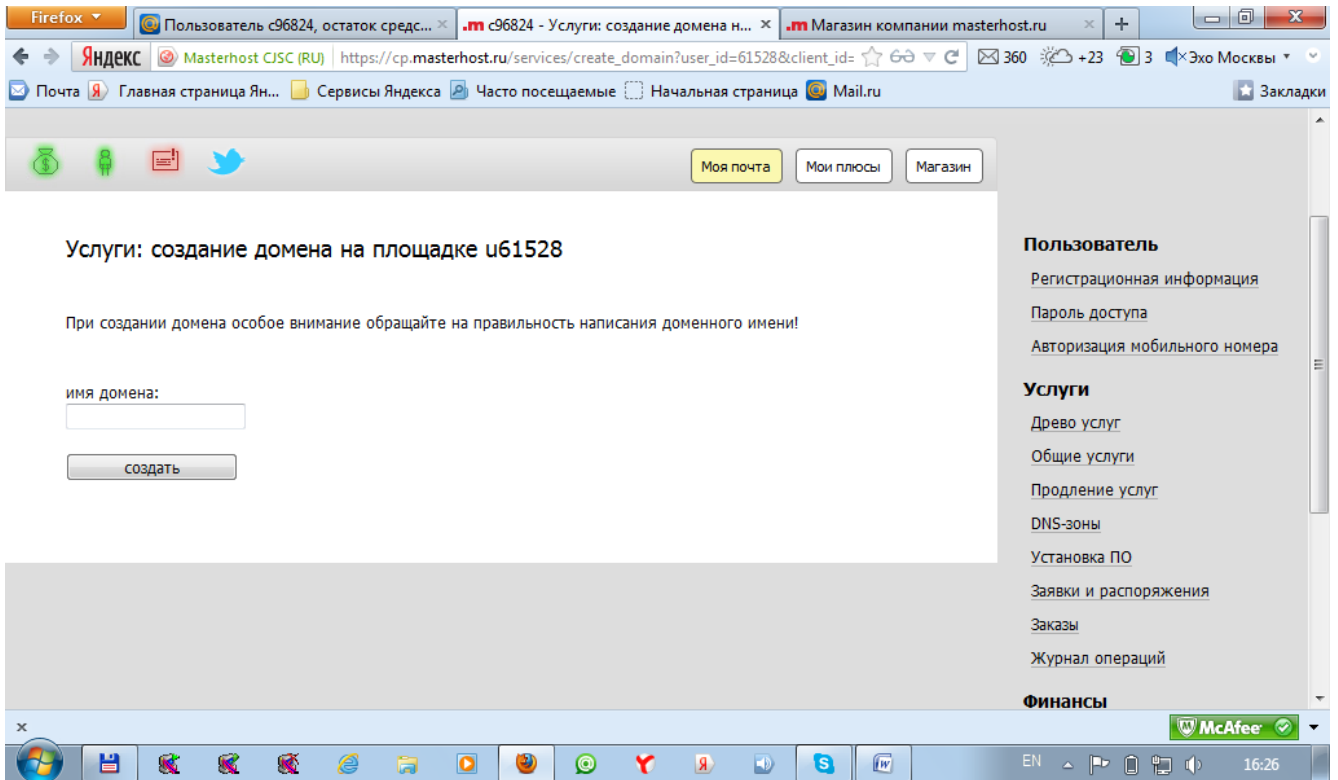
Визитка	ФорСайт	ФорСайт+2	ФорСайт+5
1 Гб места на диске	5 Гб места на диске	7 Гб места на диске	10 Гб места на диске
5 сайтов	5 сайтов	7 сайтов	10 сайтов
	MySQL, PHP, CGI	MySQL, PHP, CGI	MySQL, PHP, CGI
75 руб. в месяц	250 руб. в месяц	350 руб. в месяц	500 руб. в месяц
Купить →	Купить →	Купить →	Купить →

После покупки домена вас могут попросить подтвердить данные, путем отправки по электронной почте скан-копию вашего паспорта.

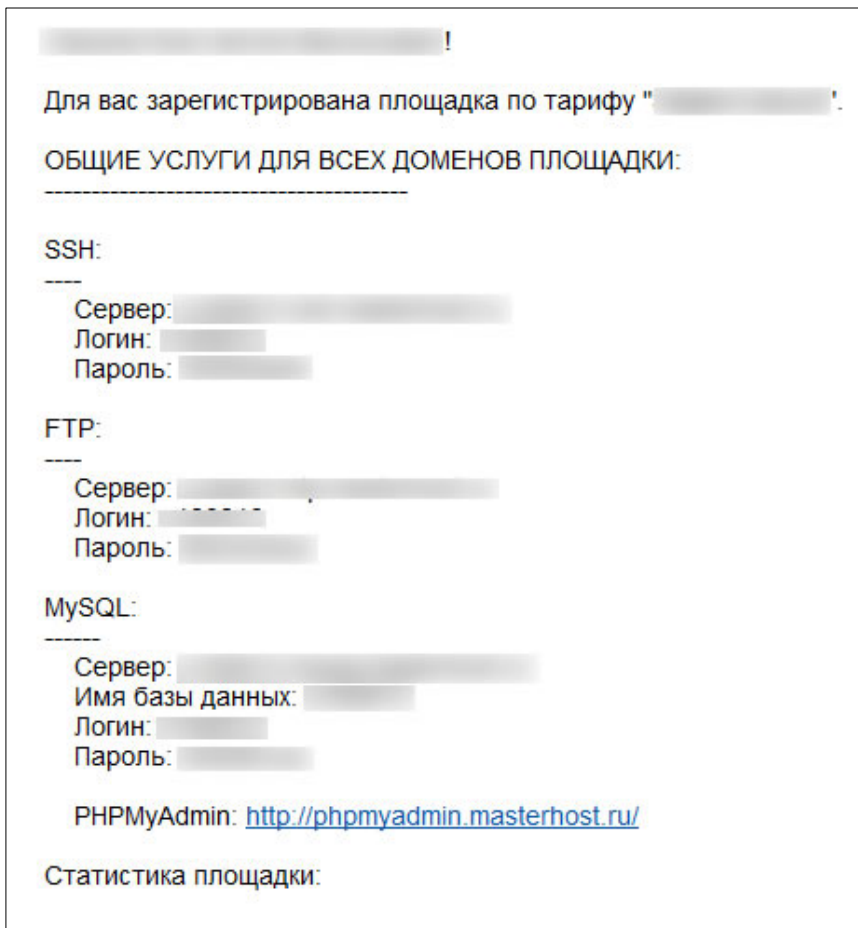
Последнее, что вам необходимо будет сделать, это присоединить домен к вашей площадке. Кликаете справа в меню Дерево услуг, выбираете свою площадку и кликаете «Разместить домен»



В окошко вводите имя домена и нажимаете кнопку создать.



После присоединения домена к площадке начнется его делегирование. Вам необходимо просматривать вашу почту, куда будут приходить информационные письма от Мастерхост. Среди них будет письмо с параметрами входа в файловую систему сервера вашей площадки. Его я рекомендую распечатать и сохранить в надежном месте.



Также распечатайте и сохраните письма, которые касаются регистрации доменов.

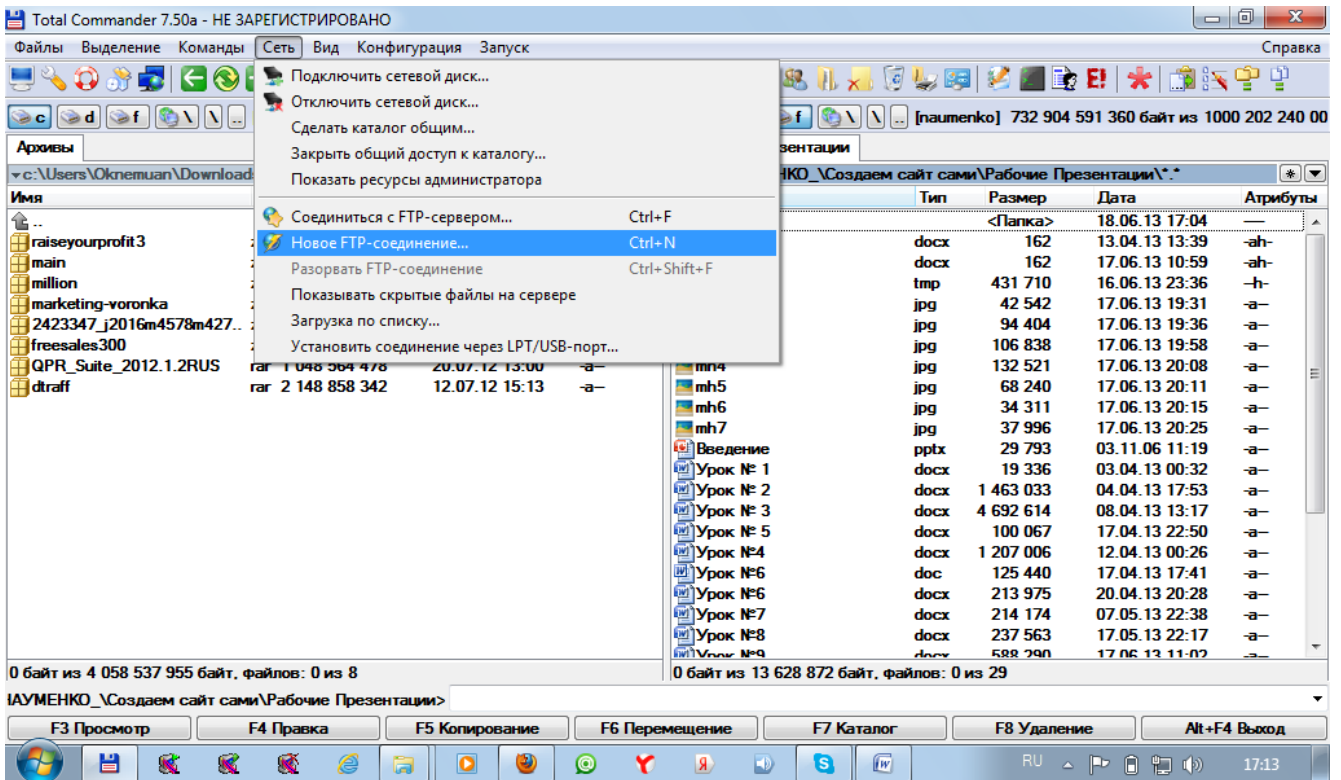
Теперь, когда вы станете счастливым обладателем домена и хостинга, где-то через сутки домен проделегирован и у вас появится физический доступ к вашей площадке.

Чтобы воспользоваться вашей площадкой на удаленном сервере, вам понадобится программа, помогающая осуществлять взаимодействие с удаленным веб-сервером по протоколу ftp.

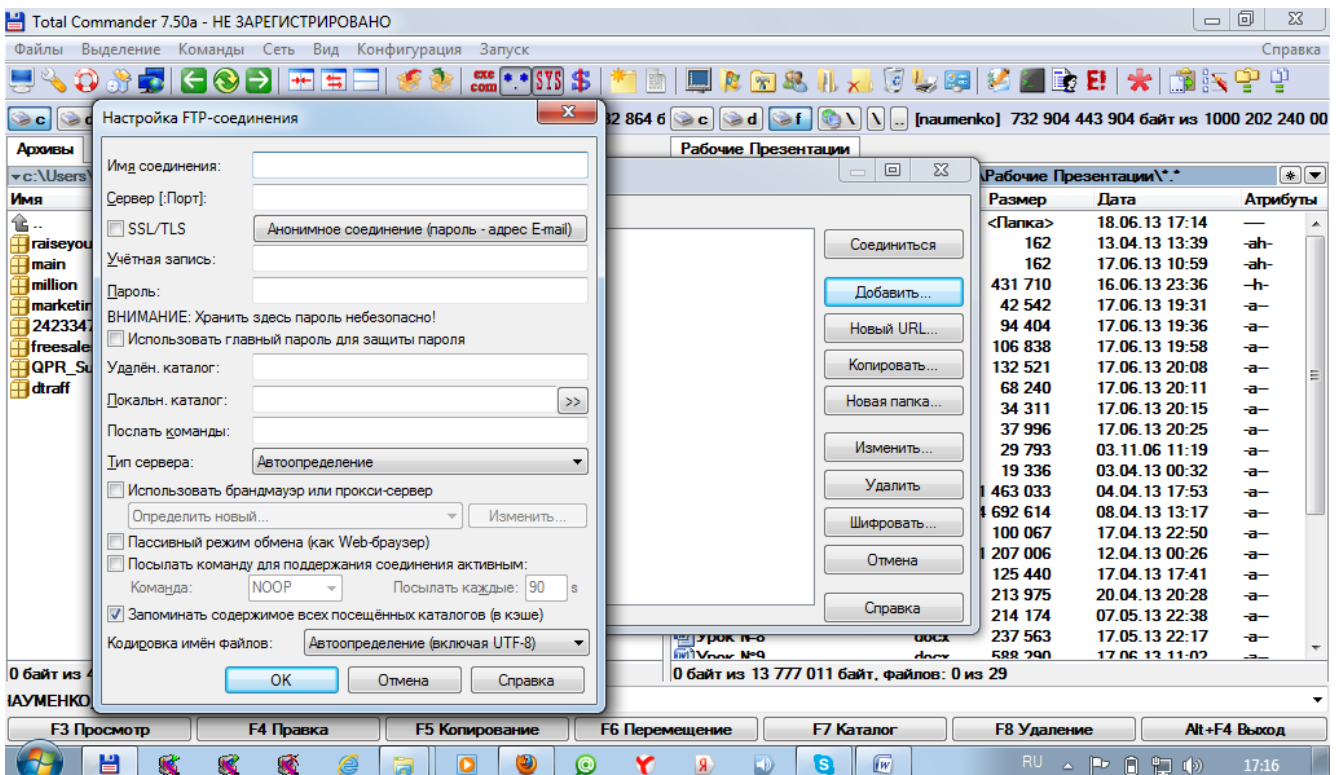
На мой взгляд, самой удобной программой для этих целей является total commander. Скачать его можно по этой ссылке: http://downloads.hotdownloads.ru/total_commander/tcmd750a.exe.

После установки необходимо настроить ftp соединение с удаленным web-сервером:

Сеть-Новое FTP-соединение



Или Сеть-Соединиться с FTP-сервером-Добавить



Вводите имя соединения, например MyFirstSite.

Сервер[Порт]-сюда вводите имя ftp сервера, которое пришло в электронном письме при покупке площадки.

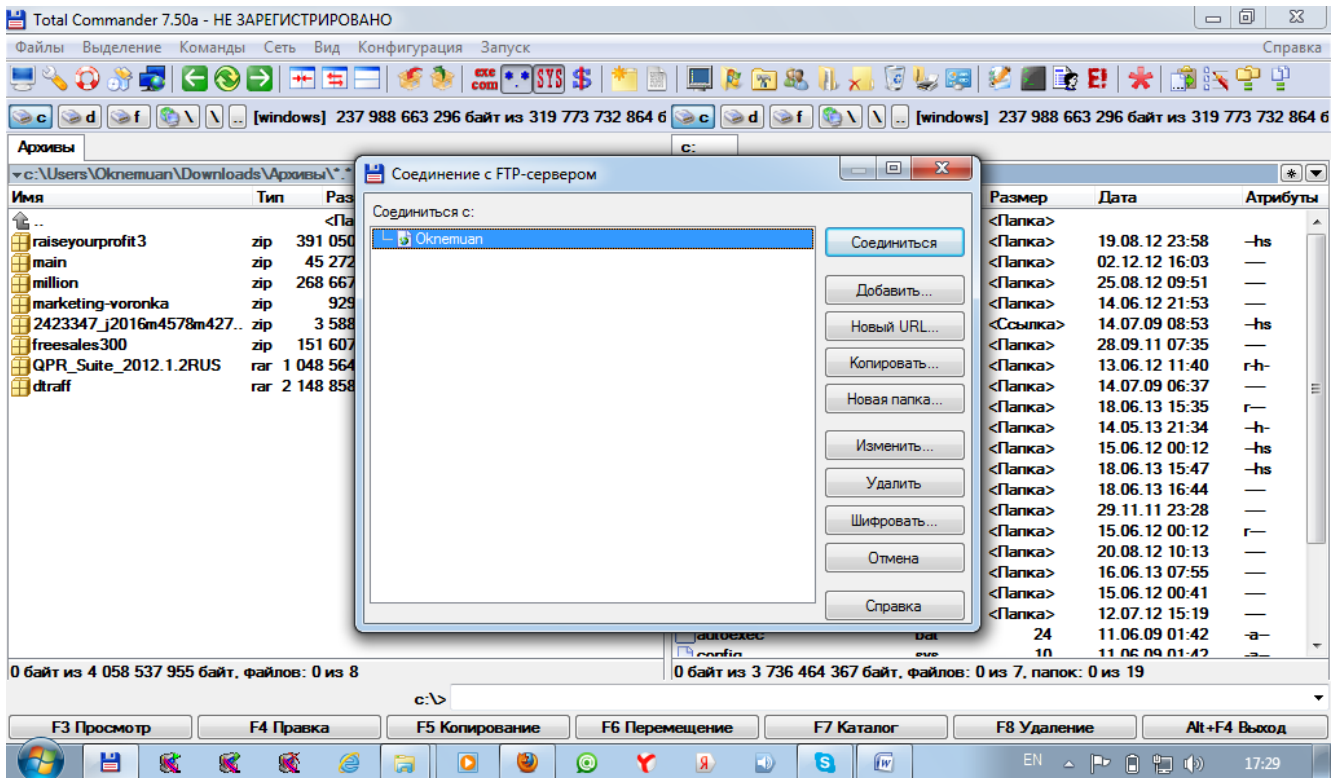
Учетная запись- сюда вводите свой логин, который также возьмете из письма.

Пароль вводить и сохранять не рекомендуется, лучше его вводить в процессе соединения. Так безопаснее.

Установите галочку Пассивный режим обмена (как Web-браузер).

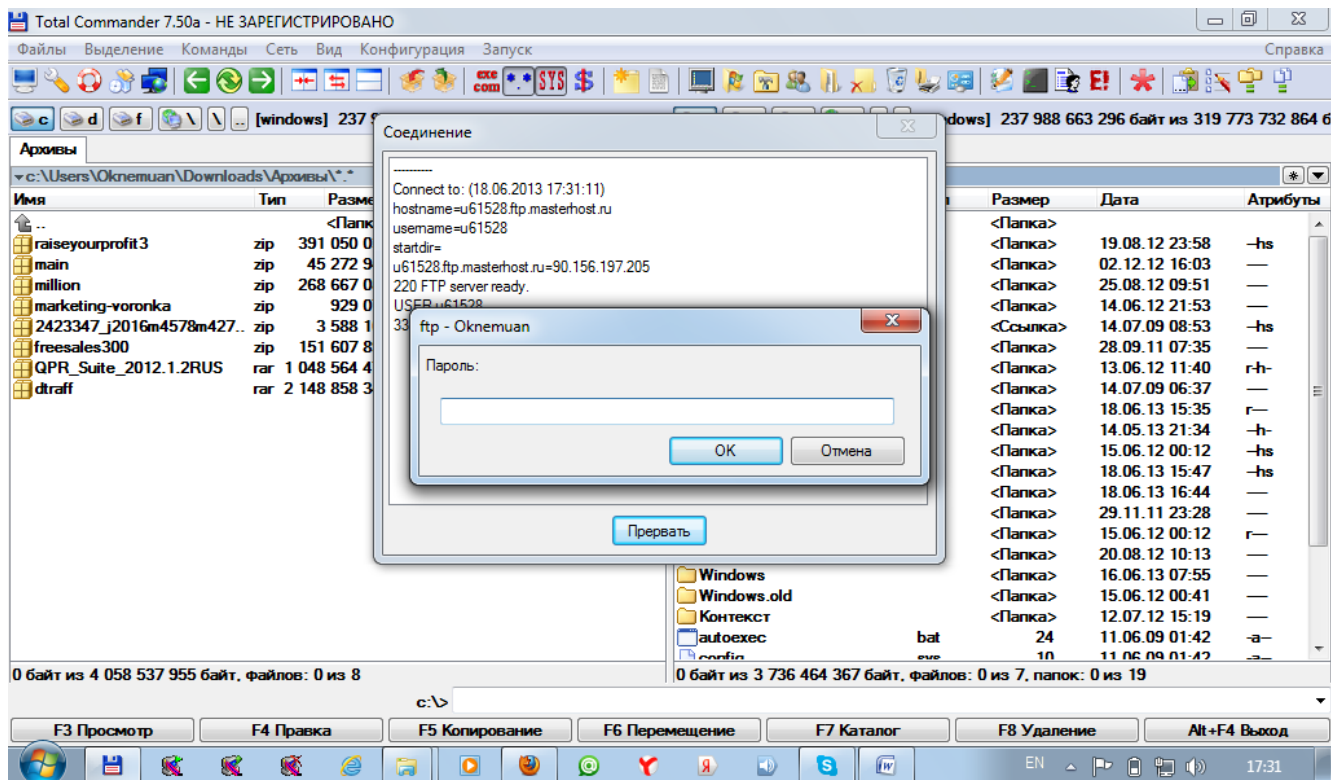
Нажмите ОК.

Ваше соединение окажется в списке

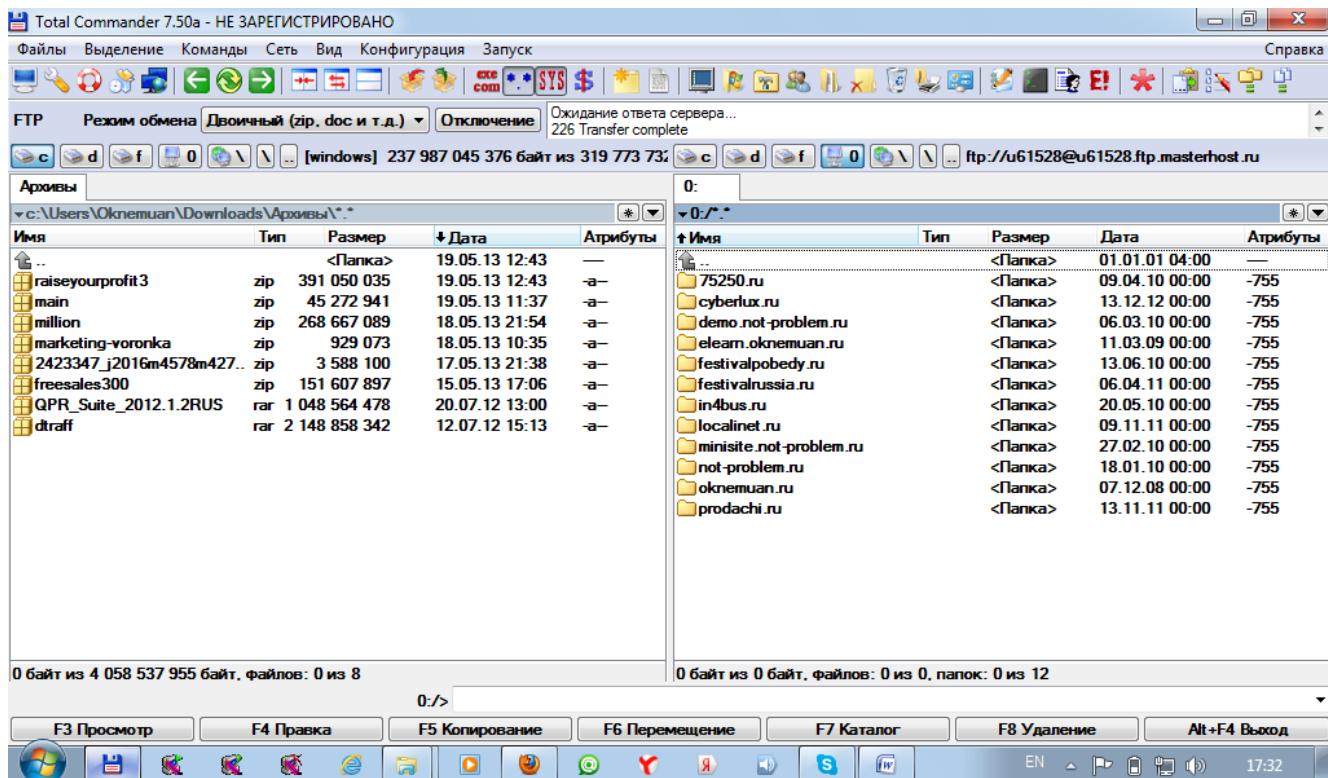


Для соединения с удаленным сервером, выделите соединение и кликните по кнопочке «Соединиться».

В окно для пароля вбейте пароль, который пришел в письме.

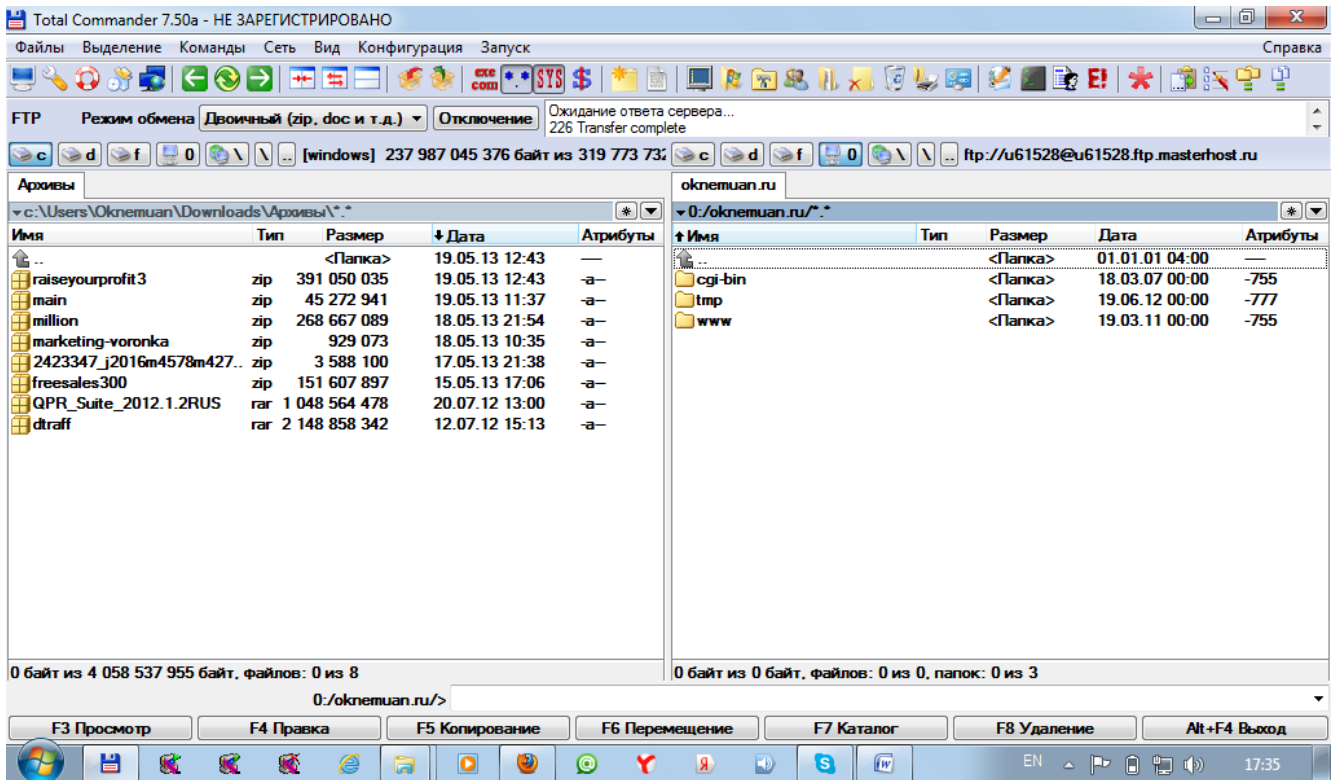


Введите пароль и нажмите OK



Появится раздел 0 и в нем папочка с именем вашего домена. Если у вас к площадке присоединено несколько доменов, как у меня, то все они будут выглядеть как папочки с именами доменов.

Кликайте по папочке



Вы увидите среди прочих папочку с именем www. Это и будет ваша папочка, куда нужно будет скопировать файлы вашего сайта. Для этого нужно ее открыть, а в левой части программы найти папку www с вашим сайтом на вашем компьютере и перетащить все файлы из вашей папки www в папку www на сервере.

После того, как вы загрузите все файлы сайта, можно открывать браузер и ввести в строку адреса доменное имя. Браузер загрузит и отобразит файл с именем index.html.

Заключение

Благодарю вас, уважаемый и терпеливый читатель, за то, что вы прочитали эту книгу!

Создание сайтов это одно из увлекательных, полезных и прибыльных занятий. Изучение вопросов сайтостроения начинается с освоения основ, которые вы изучили в только что прочитанной книге.

Как действовать дальше?

Статичные сайты пригодятся вам для создания продающих одностраничных или небольших промо-сайтов. Немного практики, задатки которой вы получили после прочтения этой книги, а также любознательности и вы станете экспертом в области создания статичных WEB-сайтов.

Создание более сложных многостраничных управляемых корпоративных сайтов или блогов, а тем-более интернет-магазинов, требуют несколько иных, расширенных знаний. Потому что сайты, которыми можно управлять создаются на основе технологий WEB-программирования. Именно поэтому следующим шагом в освоении сайтостроения у вас должен стать изучение технологии WEB-программирования. Сегодня существует множество платформ для программирования сайтов. Наиболее простой в освоении и самой массовой на сегодняшний день является технология PHP+MySQL. Это серверная технология, в основе которой лежит язык серверного WEB-программирования под названием PHP и база данных под названием MySQL.

Освоение данной технологии выведет вас на новый уровень построения сайтов. Станут доступны возможности для создания управляемых сайтов и интернет-магазинов. Их вы сможете сделать своими руками. Освоив работу базы данных MySQL, в совокупности с языком PHP, вы научитесь создавать динамичные сайты.

Динамичные сайты, в отличие от статичных, формируют свой контент в результате работы серверных программ, написанных на языке PHP.