

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Тульский государственный университет»

Институт педагогики, физической культуры, спорта и туризма
Кафедра «Туризм и индустрия гостеприимства»

Утверждено на заседании кафедры
«Туризм и индустрия гостеприимства»
«27» января 2022 г., протокол № 5
Заведующий кафедрой

 И.Ю. Пономарева

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по выполнению лабораторных работ
по дисциплине
«Программные средства разработки WEB-страниц и презентаций»

основной профессиональной образовательной программы
высшего образования – программы бакалавриата

по направлению подготовки **43.03.03 Гостиничное дело**

с направленностью (профилем)
Гостиничная деятельность

Формы обучения: *очная, заочная*

Идентификационный номер образовательной программы: 430303-01-22

Тула 2022 год

Разработчики методических указаний

Танкиева Т.А., канд. техн. наук, доцент ТулГУ
(ФИО, должность, ученая степень, ученое звание)



(подпись)

Содержание

1	Разработка Web-страниц. Структура HTML-документа.....	4
2	Разработка Web-страниц. Форматирование шрифта, таблицы, ссылки.....	12
3	Разработка Web-страниц. Формы.....	24
4	Каскадные таблицы стилей. CSS.....	37
5	Создание элементов страниц. Создание меню.....	46
6	Создание элементов страниц. Базовые разметки страниц.....	61
7	Синтаксис и грамматика JavaScript.....	75
8	Синтаксис и грамматика JavaScript. Основные конструкции.....	91
9	Методы ввода/вывода информации.....	97
10	Объектная модель документа.....	103
11	Серверные сценарии MySQL+PHP.....	111
12	Сценарии двунаправленного обмена данными, Mootools+Ajax+PHP.....	127
13	Создание сайта средствами CMS.....	139
14	Создание презентаций. Интерфейс программы. Основные элементы презентации.....	150
15	Создание презентаций. Работа со слайдами.....	151
16	Создание презентаций. Создание шаблонов.....	151
17	Создание презентаций. Графика. Анимация.....	169
18	Создание презентаций. Работа со звуком, включение видео.....	179
19	Создание презентаций. Разработка презентации, управление показом.....	186
	Библиографический список рекомендуемой литературы.....	192

1 Разработка Web-страниц. Структура HTML-документа

Цель работы – ознакомление со структурой HTML-документы.

Задачи работы – получение навыков создания HTML-документа, получение навыков использования META-инструкций.

Теоретические сведения

Современный HTML-документ должен состоять из двух разделов – заголовка и тела, таким образом, HTML-документ имеет следующую структуру:

Объявление документа как документа содержащего HTML-элементы - `<html>`

Объявление раздела заголовка - `<head>`

Содержание раздела заголовка – это как правило управляющая информация для браузера, например название документа отображаемое в консоли `<title>` и информация о документе META-инструкции `<meta>`

Закрытие раздела заголовка - `</head>`

Объявление раздела тела страницы – это содержимое страницы, которое видит пользователь `<body>`

Содержимое раздела тела страницы – текст, списки, картинки, таблицы, гиперссылки, блочные элементы и пр.

Закрытие раздела тела страницы - `</body>`

Закрытие HTML - документа - `</html>`

Пример простой HTML-странички (см. рисунок 1):

```
<html>
  <head>
    <title>Простая web-страница</title>
    <meta name="description" content="Пример простой web-страницы">
  </head>
  <body bgcolor="#333333" text="#ffff33" link="#ff3333" vlink="#2779c1"
alink="#66ff00">
    <h1 align="center">Простая страничка</h1>
    <p>Чтобы узнать больше <a href="home.html">жми сюда</a>.</p>
  </body>
</html>
```

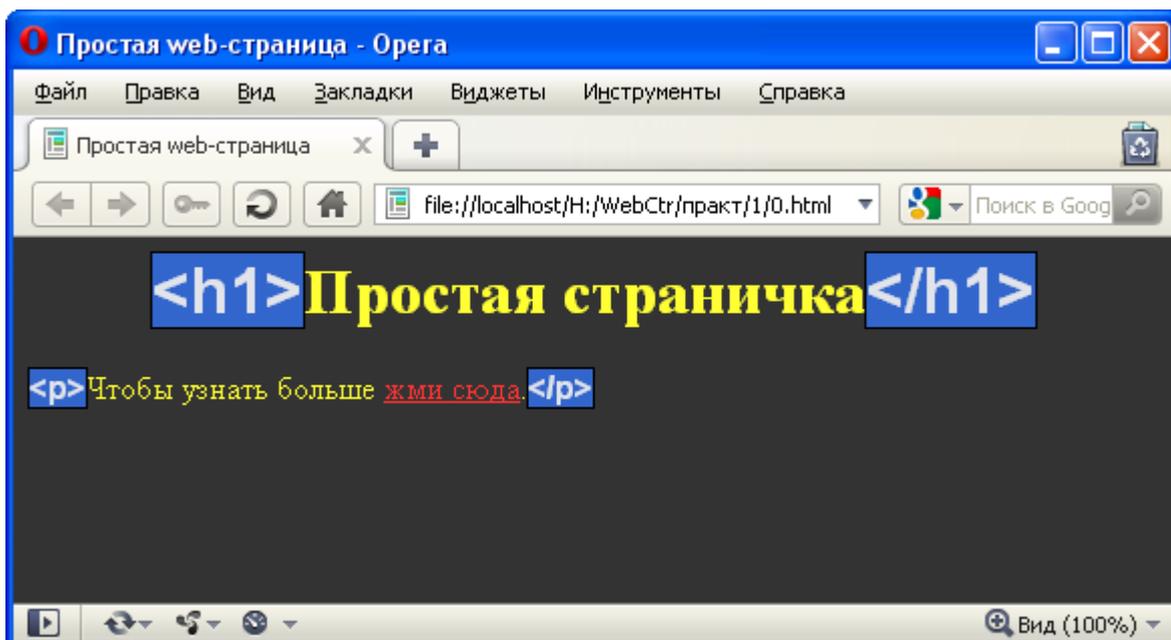


Рисунок 1 – Вид страницы в режиме отображения блочных элементов в браузере Опера

Браузеры игнорируют символы конца строки и множественные пробелы и символы табуляции в HTML-файлах, поэтому их можно использовать для повышения читабельности кода страницы.

Заголовок HTML-документа может включать неограниченное количество так называемых МЕТА-инструкций. МЕТА-инструкция (метатег) – это способ определить некоторую переменную путем указания ее имени (атрибут name) и значения (атрибут content). Наиболее типичные МЕТА-инструкции содержащие информацию для поисковых машин (поисковые машины, сохраняют ее в своих базах данных и демонстрируют ее в ответ на запросы пользователей):

```
<meta name="description" content="Официальный сайт ТулГУ">
```

Переменная description содержит краткое описание документа.

```
<meta name="keywords" content="Высшее профессиональное образование, высшее техническое образование, гуманитарное образование, экономическое образование, ВУЗ, вуз">
```

Переменная keywords, содержит набор ключевых слов, описывающих содержание документа.

Другая группа МЕТА-инструкций определяет эквиваленты команд протокола передачи гипертекстов.

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

Эта МЕТА-инструкция дает браузеру указание интерпретировать загружаемый документ как содержащий HTML-текст в кодировке Windows/1251.

```
<meta http-equiv="Content-Type" content="text/html; charset=koi8-r">
```

Эта МЕТА-инструкция абсолютно аналогична предыдущей, только в качестве кодировки указана КОИ-8.

```
<meta http-equiv="Refresh" " content="[время]; url=[документ]">
```

Через указанное [время]; браузер загрузит страничку указанную в url.

```
<meta http-equiv="Refresh" "content="12; url=http://www.tula.ru/">
```

МЕТА-инструкцию Refresh можно использовать, например, если Вы перенесли некоторый документ с одного сервера на другой. Вместо копии документа на старом

сервере можно оставить короткое сообщение о переносе, включающее МЕТА-инструкцию Refresh и адрес документа на новом сервере. Тогда через указанное время гость переместится на страницу по новому адресу. МЕТА-инструкцию Refresh можно использовать и для создания слайд-фильма. Для слайд-фильма из 5 кадров необходимо создать 5 файлов html связанных с помощью метатегов. Метатег с атрибутом `http-equiv="refresh"` в каждой странице должен перенаправлять на следующую страницу слайда см. рисунок 2.

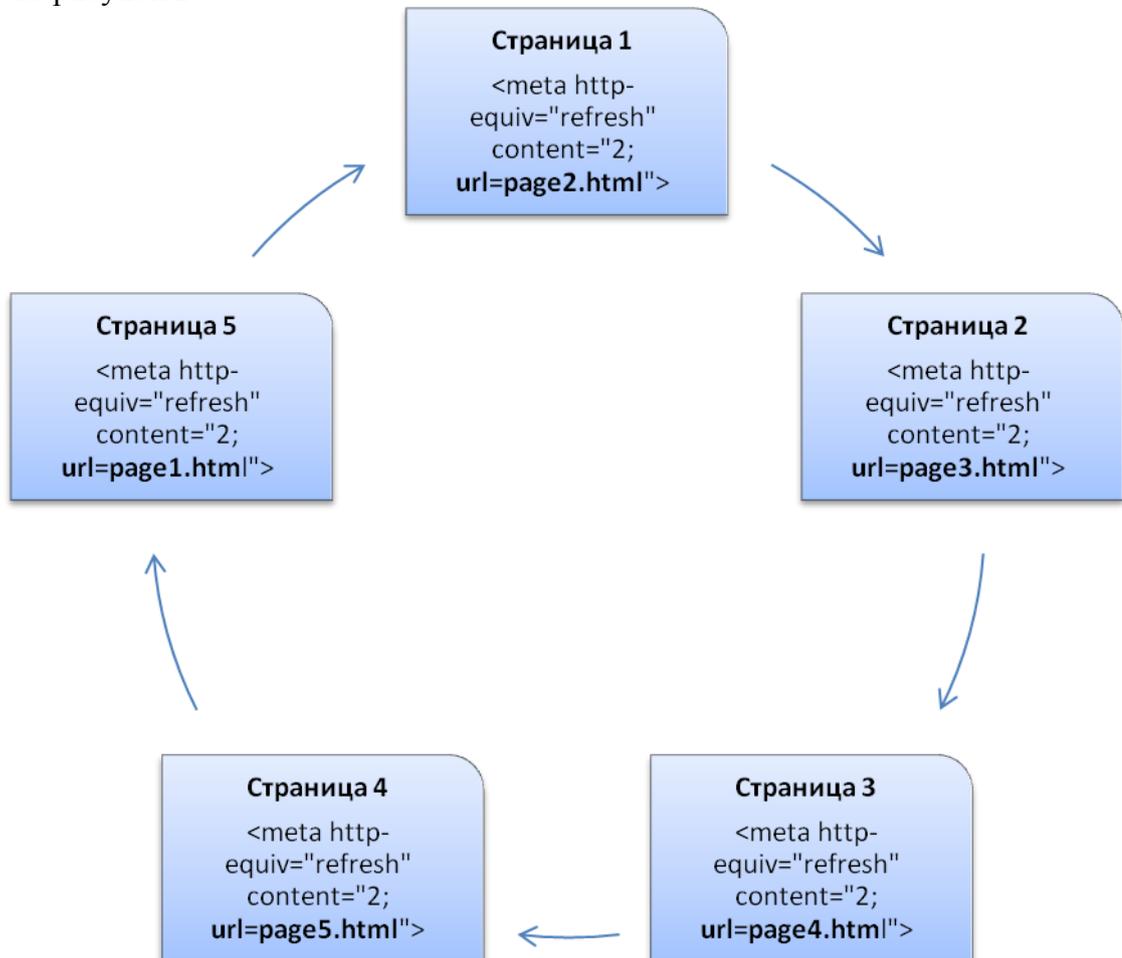


Рисунок 2 – Связывание страничек слайд-фильма метатегом с атрибутом `http-equiv="refresh"`

Таблица – Справка по тегам, используемым в работе

Тэг	Описание	Атрибуты
<html>	Идентифицирует документ как содержащий элементы HTML	необязательные атрибуты lang язык scroll прокрутка (игнорируется браузерами) dir направленность текста xmlns=http://www.w3.org/1999/xhtml адрес документации (для некоторых фильтров EI)
<head>	Элемент-контейнер, определяющий заголовок HTML-документа, внутри	profile, id, name, class, title

	<p>которого задается информация о документе и управляющие инструкции по его обработке. Эти данные задаются посредством элементов заголовка, заключенных между начальным и конечным тегами элемента head</p>	
<meta>	<p>Элемент заголовка, используемый для задания таких сведений о документе, как автор документа, список ключевых слов, кодировка и т.п. Благодаря этому осуществляются некоторые возможности по управлению обработкой HTML-документа</p>	<p>name указывает имя свойства; content задает значение свойства; scheme указывает имя схемы, используемой для обработки значения свойства; http-equiv используется вместо атрибута name для указания имени http-сообщения; lang информация о языке. Необязательный атрибут; dir указывает направленность текста. Необязательный атрибут.</p>
<title>	<p>Название документа, показываемое в заголовке окна браузера</p>	
<body>	<p>Этот элемент является контейнером информационного содержимого документа, при условии, что этот документ имеет не фреймовую структуру. В противном случае вместо элемента body используется элемент frameset. Элемент body определяет тело документа, в содержимом которого задаются все HTML-элементы тела.</p>	<p>Все атрибуты элемента BODY являются необязательными. alink определяет цвет активной (выбранной пользователем) ссылки; link указывает, каким должен быть цвет еще не просмотренной ссылки; vlink задает цвет уже просмотренной ссылки; bgcolor устанавливает, какого цвета должен быть фон документа; background указывает месторасположение (URL-адрес) изображения, которое используется в качестве фонового; bgproperties если установлен в положение fixed, то фоновое изображение не прокручивается. По умолчанию установлено положение nonfixed; text определяет цвет текста документа; leftmargin, rightmargin, topmargin = pixels устанавливают размер левого, правого и верхнего полей в документе соответственно. Задаются в пикселях; scroll логический атрибут: если установлен в положение no, то полоса</p>

		прокрутки документа не отображается. По умолчанию используется в значении yes .
--	--	--

Оборудование, программное обеспечение

Оборудование – ПК с установленной графической операционной системой (Windows или Linux).

Программное обеспечение – текстовый редактор, например, Блокнот; графический Web-браузер, например: Internet Explorer, Mozilla Firefox, Safari, Google Chrome или Opera.

Задание на работу

1. Создать HTML-документ page.html, представленный на рисунке 3. В страницу добавить метатеги с указанием тематики страницы, ключевых слов и кодировки.

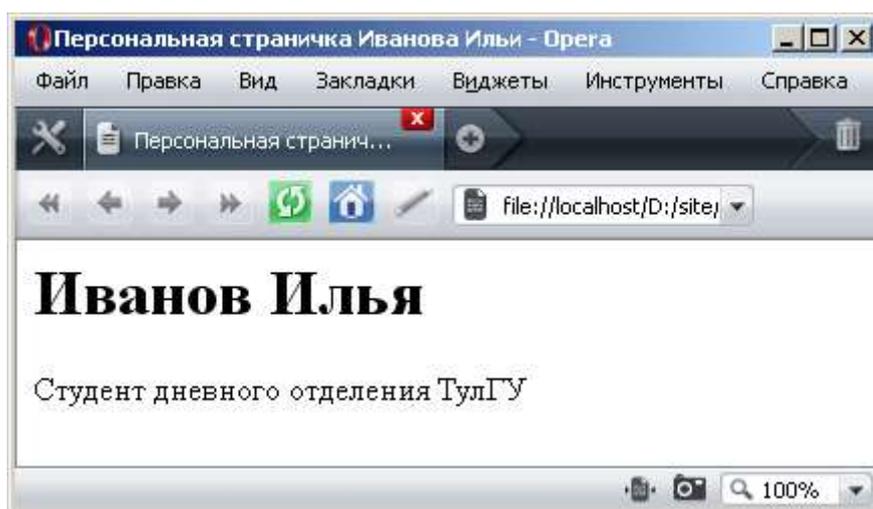


Рисунок 3 – Задание 1

2. Создать показ слайд-фильма из HTML-документов, кадры слайд-фильма показаны на рисунке.

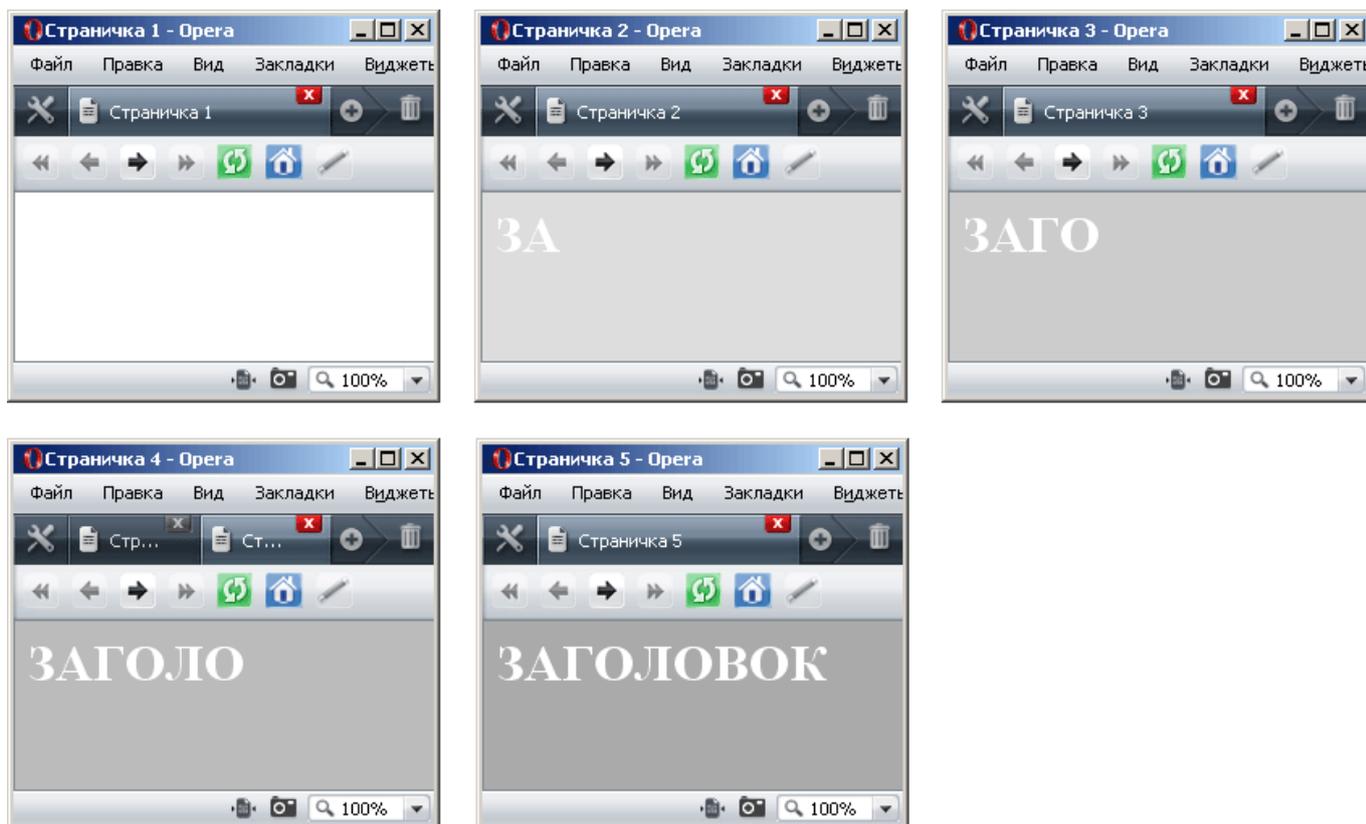


Рисунок 4 – Задание 2

Ход работы

Задание 1

1. В программе Блокнот наберите следующий текст:

```
<html
<head>
  <title>Персональная страничка Иванова Ильи</title>
  <meta name="description" content="Официальный сайт Иванова Ильи">
  <meta name="keywords" content="Иванов Илья, персональная страница, студент">
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
</head>
<body>
  <h1>Иванов Илья</h1>
  <p>Студент дневного отделения ТулГУ</p>
</body>
</html>
```

2. Сохраните документ как page.html, в диалоговом окне при сохранении файла сначала задайте Тип файла: Все файлы, а затем введите Имя файла вместе с расширением файла через html (см. рисунок 5).

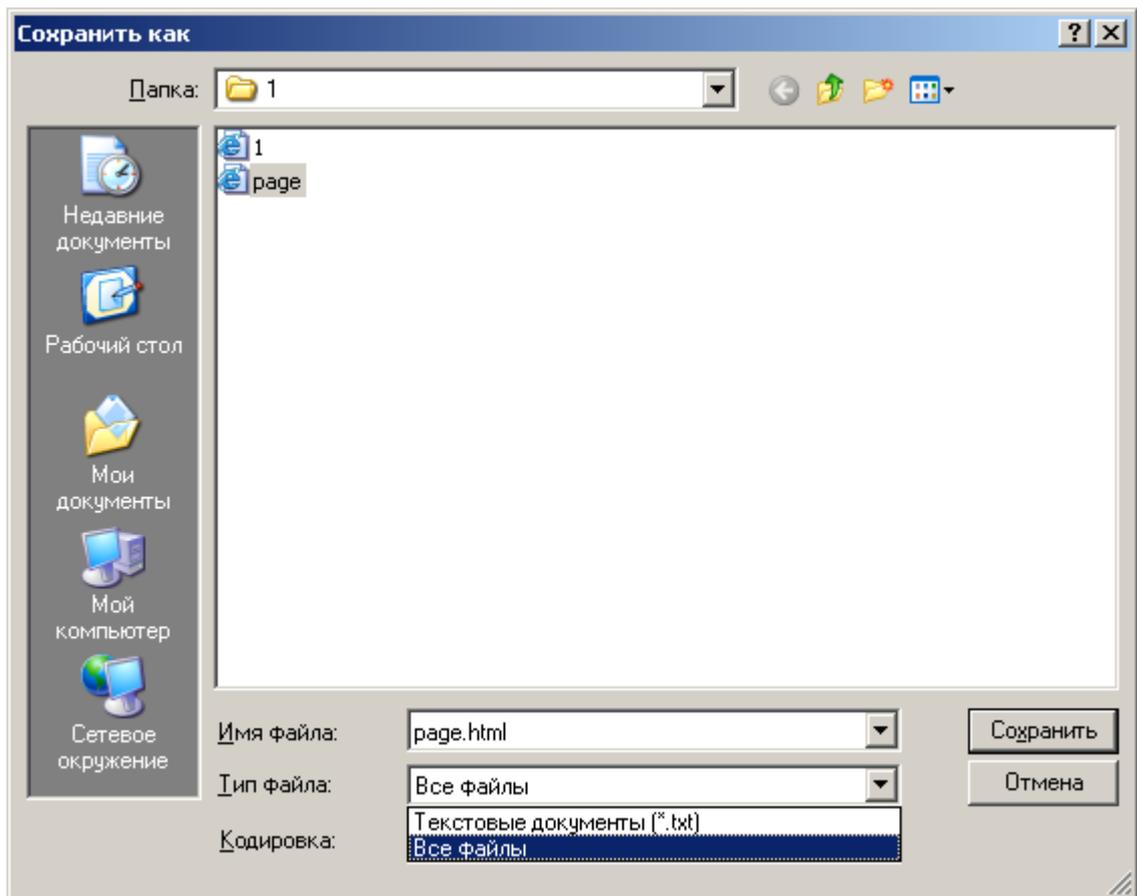


Рисунок 5 – Сохранение файла как HTML-документа

3. Проверьте работу странички в браузере. Исправьте ошибки (если есть).

Задание 2

1. Для слайд фильма создайте 5 веб-страниц. Листинг страничек приведен ниже.

Страница 1. Имя файла page1.html. Листинг файла:

```
<html>
<head>
<meta http-equiv="refresh" content="1; url=page2.html">
<title>Страничка 1</title>
</head>
<body bgcolor="#FFFFFF" text="#FFFFFF">
<h1></h1>
</body>
</html>
```

Страница 2. Имя файла page2.html. Листинг файла:

```
<html>
<head>
<meta http-equiv="refresh" content="2; url=page3.html">
<title>Страничка 2</title>
</head>
<body bgcolor="#DDDDDD" text="#FFFFFF">
<h1>3A</h1>
</body>
```

```
</html>
```

Страница 3. Имя файла page3.html. Листинг файла:

```
<html>
<head>
<meta http-equiv="refresh" content="2; url=page4.html">
<title>Страничка 3</title>
</head>
<body bgcolor="#CCCCCC" text="#FFFFFF">
<h1>ЗАГО</h1>
</body>
</html>
```

Страница 4. Имя файла page4.html. Листинг файла:

```
<html>
<head>
<meta http-equiv="refresh" content="2; url=page5.html">
<title>Страничка 4</title>
</head>
<body bgcolor="#BBBBBB" text="#FFFFFF">
<h1>ЗАГОЛО</h1>
</body>
</html>
```

Страница 5. Имя файла page5.html. Листинг файла:

```
<html>
<head>
<meta http-equiv="refresh" content="2; url=page1.html">
<title>Страничка 5</title>
</head>
<body bgcolor="#AAAAAA" text="#FFFFFF">
<h1>ЗАГОЛОВОК</h1>
</body>
</html>
```

Содержание отчета

Отчет по лабораторной работе должен содержать:
конспект по теоретической части: описание общей структуры документа,
описание МЕТА-инструкций (назначение, конструкция);
отчет об используемых в работе тегах.
К отчетам должны иметься выполненные в ходе работы файлы.

2 Разработка Web-страниц. Форматирование шрифта, таблицы, ссылки

Цель работы – ознакомление со средствами форматирования текста.

Задачи работы – получение навыков работы создания HTML-документа содержащего текст.

Теоретические сведения

Стандарт HTML допускает два подхода к шрифтовому выделению фрагментов текста: физический стиль и логический стиль.

Физический стиль – прямое указание браузеру на модификацию текущего шрифта (что шрифт на некотором участке текста должен быть **жирным** или *наклонным* или моноширинным). Например, все, что находится между метками `` и ``, будет написано **жирным шрифтом**. Текст между метками `<i>` и `</i>` будет написан *наклонным шрифтом*. Текст между `<tt>` и `</tt>` будет написан шрифтом, имитирующим пишущую машинку, то есть имеющим фиксированную ширину символа.

Логический стиль – текст помечается, как имеющий некоторый отличный от нормального **логический стиль**, интерпретацию этого стиля производит браузер. Разные браузеры обрабатывают одни и те же метки логических стилей по-разному, например, текст в тэге `` браузеры под Windows заменяют курсивом, а браузеры под Unix - жирным. Большинство из этих тэгов делают текст или наклонным, или жирным, или моноширинным. Желательно использовать логические стили – упрощает использование **css**, расширяет возможности форматирования текста, делает код странички более понятным.

Таблица 1 – Справка по тегам, используемым в работе

Тэг	Описание	Атрибуты
<code><h1></code>	Заголовок 1-го уровня. Все заголовки выводятся жирным шрифтом.	align выравнивание, может принимать следующие значения: left выравнивание по левому краю (используется по умолчанию); right выравнивание по правому краю; center выравнивание по центру; justify выравнивание по ширине;
<code><h2></code>	Заголовок 2-го уровня	id , class идентификаторы в пределах документа;
<code><h3></code>	Заголовок 3-го уровня	lang , dir информация о языке и направленности текста;
<code><h4></code>	Заголовок 4-го уровня	title заголовок элемента (выводится браузером в качестве комментария при наведении курсора на содержимое элемента);
<code><h5></code>	Заголовок 5-го уровня	style встроенная информация о стиле;
<code><h6></code>	Заголовок 6-го уровня	onclick , ondblclick , onmousedown , onmouseup , onmouseover , onmousemove , onmouseout , onkeypress , onkeydown , onkeyup внутренние события;
<code>
</code>	Вставка конца	clear единственный атрибут, причем необязательный.

	строки, не парный	<p>указывает, где в окне браузера должна появиться перенесенная после использования элемента br строка. этот атрибут учитывает прикрепляемые объекты (таблицы, изображения и т.п.). <i>принимаемые значения:</i></p> <p>none следующая строка текста отображается обычным образом, на ближайшем свободном снизу пространстве. это значение используется по умолчанию;</p> <p>left следующая строка текста отображается на ближайшем свободном пространстве под прикрепленным у левого поля объектом;</p> <p>right следующая строка текста отображается на ближайшем свободном пространстве под прикрепленным у правого поля объектом;</p> <p>all следующая строка текста отображается на ближайшем свободном пространстве под прикрепленным у любого поля объектом.</p>
<center>	Выравнивает заключенные в него компоненты по центру	<p>все необязательные</p> <p>id, class идентификаторы в пределах документа;</p> <p>lang, dir информация о языке и направленности текста;</p> <p>title заголовок элемента (выводится браузером в качестве комментария при наведении курсора на содержимое элемента);</p> <p>style встроенная информация о стиле;</p> <p>onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup</p>
<hr>	Помещает горизонтальную перекладину на страницу, не парный	<p>все необязательные</p> <p>align выравнивание, может принимать следующие значения:</p> <p>left выравнивание по левому краю;</p> <p>right выравнивание по правому краю;</p> <p>center выравнивание по центру (используется по умолчанию);</p> <p>size этим атрибутом задается толщина линии в пикселах;</p> <p>width длина линии в пикселах или процентах от ширины окна браузера;</p> <p>noshade логический атрибут, задание которого отменяет рельефность линии. по умолчанию линия прорисовывается рельефной;</p> <p>color через этот атрибут указывается цвет линии в виде ключевого слова, обозначающего цвет, или в виде rgb-кода.</p>
<p>	Этот элемент отмечает свое содержимое как один абзац. При	<p>align выравнивание внутри абзаца, принимает следующие значения:</p> <p>left выравнивание по левому краю (по умолчанию);</p> <p>right выравнивание по правому краю;</p>

	<p>этом абзац сверху и снизу выделяется пустыми строками.</p>	<p>center выравнивание по центру; justify выравнивание по ширине. id, class идентификаторы в пределах документа; lang, dir информация о языке и направленности текста; title заголовок элемента (выводится браузером в качестве комментария при наведении курсора на содержимое элемента); style встроенная информация о стиле; onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup внутренние события.</p>
--	---	--

Таблица – Справка по тегам, используемым в работе

Тэг	Описание	Атрибуты
<basefont>	Устанавливает шрифт, используемый по умолчанию	face указывает тип шрифта, которым браузер будет выводить текст документа. в качестве значений данного атрибута выступают полные названия шрифтов. может быть задано несколько названий шрифтов через запятую. в этом случае браузер начинает с первого по списку шрифта и, если не удастся его подключить, переходит к следующему. если браузер не смог подключить ни один из указанных атрибутом шрифтов, то он использует шрифт, установленный в о умолчанию;
	Определяет вид, размер и цвет шрифта для текста	<p>size указывает размер шрифта в условных единицах (от 1 до 7). шрифт, используемый браузерами по умолчанию, соответствует значению 3. атрибут size допускает относительное задание, например: size=+2. последнее имеет место обычно в сочетании с использованием элемента basefont, который задает базовый размер шрифта;</p> <p>color задает цвет шрифта. в качестве значения принимается либо название цвета, либо его шестнадцатиричный номер rgb</p> <p>id, class идентификаторы в пределах документа;</p> <p>lang, dir информация о языке и направленности текста;</p> <p>title заголовок элемента (выводится</p>

		браузером в качестве комментария при наведении курсора на содержимое элемента); style встроенная информация о стиле. если при использовании элемента font не задан ни один из атрибутов, то его наличие игнорируется.
	Выделяет текст, обычно курсивом	id, class идентификаторы в пределах документа; lang, dir информация о языке и направленности текста; title заголовок элемента (выводится браузером в качестве комментария при наведении курсора на содержимое элемента); style встроенная информация о стиле; onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup
	Выделяет текст жирным шрифтом	
<sub>	Выделяет текст, переводя его в нижний регистр	
<sup>	Переводит текст в верхний регистр, как степень	
	Списки	
	Преобразует строки текста с тэгами в маркированный список	type - задает информацию о виде используемых маркеров. может принимать следующие значения: circle маркеры отображаются в виде незакрашенных кружков; disc маркеры отображаются в виде закрашенных кружков; square маркеры отображаются в виде закрашенных квадратов; compact присутствие этого атрибута указывает браузеру на то, что данный

		<p>список он должен отображать более компактно;</p> <p>id, class идентификаторы в пределах документа;</p> <p>lang, dir информация о языке и направленности текста;</p> <p>title заголовок элемента (выводится браузером в качестве комментария при наведении курсора на содержимое элемента);</p> <p>style встроенная информация о стиле;</p> <p>onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup внутренние события.</p>
	Преобразует строки текста с тэгами в нумерованный список	<p>type указывает вид нумерации элементов упорядоченного списка. может принимать следующие значения:</p> <p>type =I .задает нумерацию арабскими цифрами (используется браузерами по умолчанию);</p> <p>type =a задает нумерацию прописными латинскими буквами;</p> <p>type =a задает нумерацию строчными латинскими буквами;</p> <p>type =i задает нумерацию большими римскими цифрами;</p> <p>type=i .задает нумерацию маленькими римскими цифрами;</p> <p>start задает начальный номер первого элемента в упорядоченном списке;</p> <p>id, class идентификаторы в пределах документа;</p> <p>lang, dir информация о языке и направленности текста;</p> <p>title заголовок элемента (выводится браузером в качестве комментария при наведении курсора на содержимое элемента);</p> <p>style встроенная информация о стиле;</p> <p>onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup внутренние события.</p>
	Обозначает один элемент в нумерованном или	type атрибут, используемый в упорядоченных списках для смены

	<p>маркированном списке <code><menu></code></p>	<p>способа нумерации; value атрибут, используемый в упорядоченных списках для явного задания номера текущего элемента списка; id, class идентификаторы в пределах документа; lang, dir информация о языке и направленности текста; title заголовок элемента (выводится браузером в качестве комментария при наведении курсора на содержимое элемента); style встроенная информация о стиле; onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup внутренние события.</p>
	<p>Таблицы</p>	
<p><code><table></code></p>	<p>Раздел тэгов <code><tr></code>, <code><td></code> и <code><th></code>, организованных по строкам и столбцам</p>	<p>align задает параметры выравнивания самой таблицы в документе; width задает рекомендуемую ширину таблицы; height атрибут указывающий рекомендуемую высоту таблицы; border управляет видимостью и толщиной рамки вокруг таблицы и видимостью рамки вокруг каждой ячейки; cellspacing .указывает расстояние в пикселах между смежными ячейками (а точнее между их рамками) как по горизонтали, так и по вертикали; cellpadding указывает размер отступа (по горизонтали и по вертикали) в пикселах между рамкой и содержимым ячейки; bordercolor этот атрибут определяет цвет рамок в таблице; bgcolor задает цвет фона на всем пространстве, занимаемом таблицей — на одержимом ячеек и на свободном пространстве между ними; id, class идентификаторы в пределах документа; lang, dir информация о языке и направленности текста; title заголовок элемента (выводится</p>

		браузером в качестве комментария при наведении курсора на содержимое таблицы); style встроенная информация о стиле; onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup внутренние события. имеются еще атрибуты rules и frames , задающие внешний вид рамок в таблице и имеющие фиксированный набор значений. их описание приведено в разделе, посвященном html -таблицам.
<caption>	Заголовок таблицы	id, class идентификаторы в пределах документа; lang, dir информация о языке и направленности текста; title заголовок элемента (выводится браузером в качестве комментария при наведении курсора на содержимое элемента); style встроенная информация о стиле; onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup внутренние события.
<th>	Строку заглавия таблицы; содержимое выровнено по центру и выводится жирным шрифтом <THEAD> Набор строк, используемых как заголовок таблицы	rowspan атрибут, отвечающий за объединение соседних ячеек в столбце; colspan атрибут, отвечающий за объединение соседних ячеек в строке; nowrap логический атрибут, наличие которого запрещает разбивать содержимое ячейки на несколько строк; width атрибут, указывающий браузеру рекомендуемую ширину ячейки в пикселах; height атрибут, указывающий браузеру рекомендуемую высоту ячейки в пикселах; align, char, charoff, valign атрибуты выравнивания; bgcolor атрибут, задающий цвет фона внутри ячейки; id, class идентификаторы в пределах
<td>	Ячейка таблицы	

		<p>документа; lang, dir информация о языке и направленности текста; title заголовок элемента (выводится браузером в качестве комментария при наведении курсора на содержимое элемента); style встроенная информация о стиле; onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup внутренние события.</p>
<code><tr></code>	Строка таблицы	<p>align, char, charoff, valign атрибуты выравнивания; bgcolor атрибут, задающий цвет фона ячеек, содержащей его строки; id, class идентификаторы в пределах документа; lang, dir информация о языке и направленности текста; title заголовок элемента (выводится браузером в качестве комментария при наведении курсора на содержимое документа); style встроенная информация о стиле; onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup внутренние события.</p>

Оборудование, программное обеспечение

Оборудование – ПК с установленной графической операционной системой (Windows или Linux).

Программное обеспечение – текстовый редактор, например, Блокнот; графический Web-браузер, например: Internet Explorer, Mozilla Firefox, Safari, Google Chrome или Opera.

Задание на работу

1. Создайте HTML-документ page.html, представленный на рисунке 1.

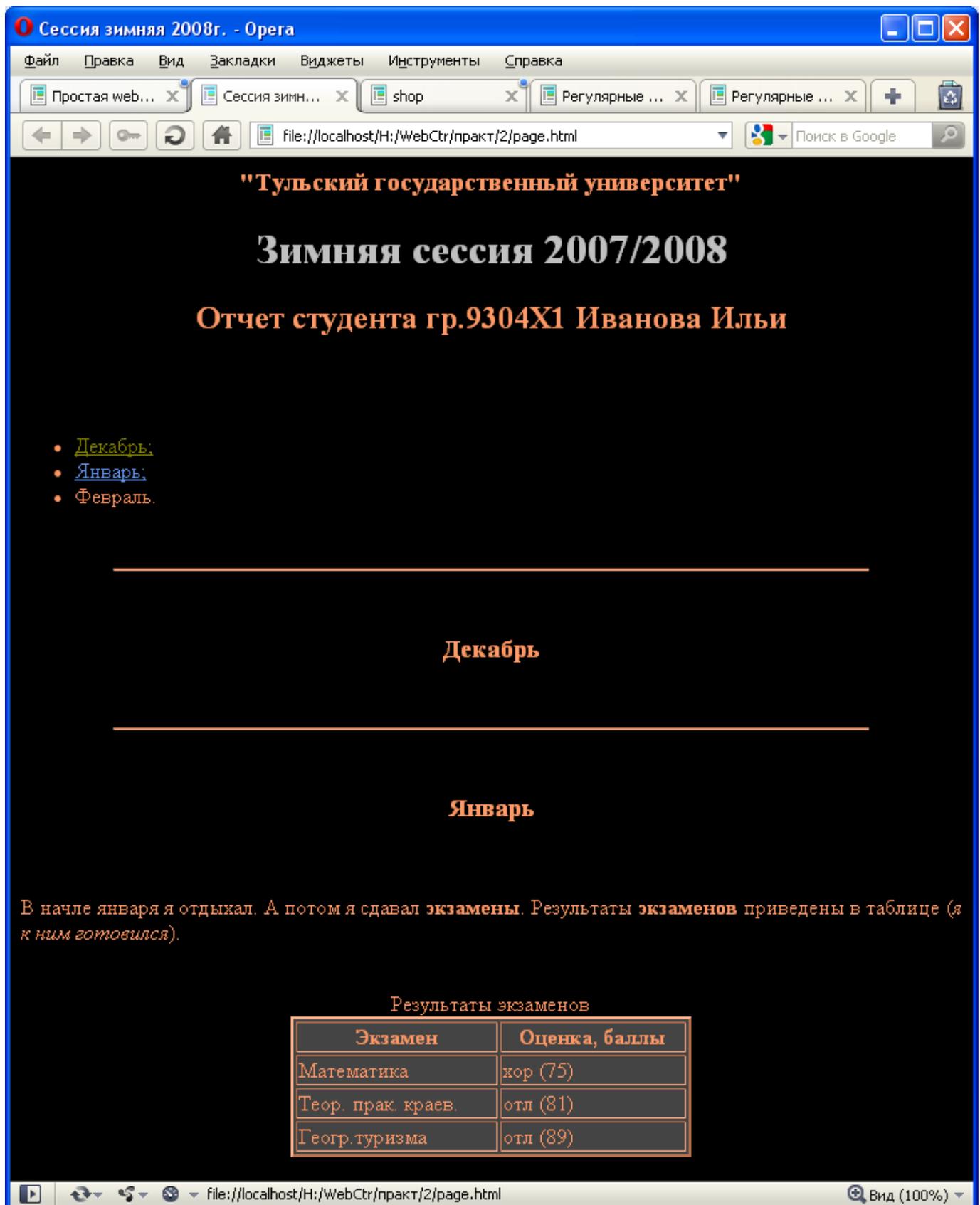


Рисунок 1 –Задание 1

2. Добавьте в созданном документе заголовок Февраль, добавьте абзацы после заголовков Декабрь и Февраль. Сделайте Февраль в списке гиперссылкой на якорь рядом с заголовком Февраль. Добавьте в таблице строку с еще одним экзаменом.

3. Создайте документ с упорядоченными списками вида 1, 2, 3 .. и I, II, III и произвольным маркированным списком.

4. Создайте файл с упорядоченным списком вида a, b, c, d ... и маркированными списками с маркерами вида ■, ○.

5. Создайте документ с использованием индексов.

6. Создайте, используя индексы, файл с формулой $y(x_1, x_2) = (4 + x_1^2)^{x_2+3}$

Ход работы

Задание 1

1. В программе блокнот создайте файл page.html. Листинг файла:

```
<html>
<head>
<title>Сессия зимняя 2008г.</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
</head>
<body bgcolor="#000000" text="#ff9966" link="cornflowerblue" vlink="olive">
<h3 align="center">&quot;Тюльский государственный университет&quot;</h3>
<font color="#c0c0c0"><h1 align="center">Зимняя сессия 2007/2008</h1></font>
<h2 align="center">Отчет студента гр.9304X1 Иванова Ильи</h2>
<br />
<br />
<ul>
<li><a href="#p1">Декабрь;</a></li>
<li><a href="#p2">Январь;</a></li>
<li>Февраль.</li>
</ul>
<br />
<hr size="2" width="80%" />
<br />
<h3 align="center">Декабрь<a name="p1"></a></h3>
<br />
<hr size="2" width="80%" />
<br />
<h3 align="center">Январь<a name="p2"></a></h3>
<br>
<p align="justify">В начале января я отдыхал. А потом я сдавал
<strong>экзамены</strong>. Результаты <strong>экзаменов</strong> приведены в
таблице (<em>я к ним готовился</em>).
</p>
<br />
<table width="300" border="2" bordercolor="#bfe25b" align="center"
bgcolor="#444444">
<caption>Результаты экзаменов</caption>
<tr> <!--Начало первой строки -->
<th><!--Начало первой ячейки-->Экзамен</th>
```

```

<th>Оценка, баллы</th>
</tr>
<tr>
<td>Математика</td>
<td>хор (75)</td>
</tr>
<tr>
<td>Теор. прак. краев.</td>
<td>отл (81)</td>
</tr>
<tr>
<td>Геогр.туризма</td>
<td>отл (89)</td>
</tr>
</table>
</font>
</body>
</html>

```

2. Проверьте работу файла в браузере. Исправьте ошибки.

Задание 2

Выполнить самостоятельно.

Задание 3

В программе блокнот создайте файл Spisok.html. Листинг файла:

```

<html>
<head> <title>Виды списков</title> </head>
<body>
<h2>Списки</h2>
<h3>Это - упорядоченный список. Начальный номер – 4</h3>
<ol type="1" start="4">
  <li>элемент</li>
  <li>элемент</li>
  <li>элемент</li>
  <li>элемент</li>
</ol>
<h3>Это - упорядоченный список. Нумерация римскими цифрами </h3>
<ol type="I">
  <li>элемент</li>
  <li>элемент</li>
  <li>элемент</li>
  <li>элемент</li>
</ol>
</li>
<h3>Это - неупорядоченный список. Здесь используются различные вводные маркеры
</h3>
<ul>
  <li>элемент</li>
  <li>элемент</li>
  <li>элемент</li>
  <li>элемент</li>
</ul>

```

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

Задание 4

Выполнить самостоятельно.

Задание 5

В программе блокнот создайте файл Index.html:

```
<html>
<head><title>Индексы</title> </head>
<body>
<h3>Пример использования индексов
<p> 2+x<sup>2</sup>
</p>
<p>a<sub>1</sub> + a<sub>2</sub>+ a<sub>3</sub>
<p>Подстрочные примечания <sup>2</sup>
</p>
</h3>
</body>
</html>
```

Задание 6

Выполнить самостоятельно.

Содержание отчета

Отчет по лабораторной работе должен содержать: конспект по теоретической части; отчет об используемых в работе тегах.

К отчетам должны иметься выполненные в ходе работы файлы.

3 Разработка Web-страниц. Формы

Цель работы – ознакомление с элементами форм HTML-документы.

Задачи работы – получение навыков создания и использования форм в HTML-документе.

Теоретические сведения

Основное назначение формы в HTML-документе передача данных на Web-сервер, где они обрабатываются CGI-программой (например php-скриптами), либо для пересылки информации по электронной почте. В основном разработчики веб-сайтов используют формы для различных регистраций, опросов и запросов, отправки постов, файлов и пр. Т.е. с помощью форм реализуется диалог между пользователем интернета и сервером. Простейшую форму можно создать с помощью тега `<isindex>` создающего форму поиска, пример:

```
<html>
  <head>
    <isindex>
  </head>
  <body>
```

Форма поиска по базе данных. Чтобы начать поиск, введите ключевое слово, после чего нажмите клавишу Enter.

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

Сложные формы создаются в HTML-документе при помощи следующих тэгов и элементов:

```
<form>
<textarea>
<input>
<select>
<option>
```

Form. Элемент, который собственно обозначает наличие формы в HTML-документе. Между его начальным и конечным тэгами содержатся все теги и элементы, участвующие в построение формы. Тэг `<form>` имеет следующие параметры:

action - единственный обязательный параметр. Задаёт URL-адрес электронной почты или CGI-программы, по которому отправляется содержимое формы.

method - задаёт способ отправки данных. Возможные значения: `get` (по умолчанию) и `post` (метод `post` обязательно используется если форма содержит поле загрузки файла, или для передачи объемных данных, в современных браузерах этот параметр используется по умолчанию) .

name - присваивает форме имя. Используется при наличии нескольких форм.

enctype - задаёт способ кодировки содержимого формы. Возможные значения: `application/x-www-form-urlencoded` (по умолчанию) и `multipart/form-data` (обязательно надо использовать если форма содержит поле загрузки файла).

Например: `<form action="http://www.real.com/form.php" method="post" name="main_form"></form>`

Textarea – создаёт область в виде прямоугольника для ввода нескольких строк текста. Имеет следующие параметры:

name - задаёт название области.

cols - задаёт ширину области в количестве печатных символов.

rows - задаёт высоту области в количестве строк.

Пример:

```
<form action="http://www.real.com/form.php" method="post" name="main_form">
```

Дополнительная информация:


```
<textarea name="additional" cols="40" rows="5">
```

Много всякой информации

```
</textarea>
```

```
</form>
```

Дополнительная информация:

Много всякой информации

Input - используется для создания полей ввода информации, а также различных кнопок и переключателей. Имеет следующие параметры:

name - задаёт полю или кнопке идентификационное имя.

type - задаёт тип поля (по умолчанию - text). Тэг <type> имеет следующие значения:

text - создаёт поле для ввода одной строки текста. Используется совместно с параметрами: maxlength - задаёт максимальное количество символов, вводимых в строку; size - задаёт ширину поля в символах; value - задаёт начальное значение поля, которое может изменить пользователь. Пример: <input type="text" name="login" maxlength="40" value="Фамилия">

Фамилия

password - создаёт поле для ввода пароля, в котором вводимая пользователем информация отображается в виде звёздочек. Используется совместно с параметрами: maxlength - задаёт максимальное количество символов, вводимых в строку; size - задаёт ширину поля в символах. Пример: <input name="pass" type="password" maxlength="8">

file - создаёт поле для ввода имени файла, который пользователь может приобщить к данной форме. Пример: <input name="t3" type="file" maxlength="40" size="20">

Обзор...

hidden - задаёт поле, которое не отображается на экране монитора, но содержит информацию, которую не может изменить ни пользователь, ни браузер.

checkbox - создаёт поле для односложных ответов (да/нет, вкл/выкл, истина/ложь) в виде установки флажка или галочки. Используется совместно с параметрами: value - задаёт значение для выбранной кнопки; checked - определяет обозначенный вариант выбранным по умолчанию; name - задаёт полю идентификационное имя. Пример:

```
<input type="checkbox" name="t4" value="Англия" checked="checked">Англия  
<input type="checkbox" name="t4" value="Франция">Франция
```

Англия Франция

radio - создаёт переключатель в группе. На экране отображается в виде круглой кнопки. Все кнопки в группе имеют одинаковый параметр name. Используется совместно с параметром value - задаёт значение для выбранной кнопки. Пример:

```
<input type="radio" name="sex" value="мужской">мужской  
<input type="radio" name="sex" value="женский">женский
```

мужской женский

submit - создаёт кнопку, при нажатии которой происходит пересылка содержимого формы. Используется совместно с параметром value - значение которого определяет надпись на кнопке. Пример:

```
<input type="submit" value="отправить">
```

image - создаёт кнопку отправки данных в виде изображения (заменяет кнопку submit). Адрес изображения задаётся параметром src. Параметр align выравнивает изображение по вертикали. Пример:

```
<input type="image" src="button.gif">
```

reset - создаёт кнопку сброса всех введённых значений формы. В момент нажатия данной кнопки форма приобретает первоначальный вид. Пример:

```
<input type="reset" value="сброс">
```

Select - создаёт раскрывающийся список или прокручиваемое меню, из которого можно выбрать как одно, так и несколько значений. Имеет следующие параметры:

name - задаёт имя списка.

size - задаёт количество видимых пунктов списка. При значении size равным единице, отображается раскрывающийся список, а при значении больше единицы - прокручиваемое меню.

multiple - задаёт возможность выбора нескольких пунктов из списка.

Пункты меню внутри элемента `<select>` задаются при помощи тэга `<option>`.

Пример:

```
<select size="1">  
  <option value=1>синий  
  <option value=2>красный  
  <option value=3>зеленый  
</select>
```

<p><form></p>	<p>описывает форму на странице, которая может содержать другие компоненты и элементы управления</p>	<p>action единственный обязательный атрибут. В качестве его значения указывается url-адрес запрашиваемой CGI-программы, которая будет обрабатывать данные, содержащиеся в форме;</p> <p>method определяет метод http, используемый для пересылки данных формы от браузера к серверу. Атрибут method, как правило, может принимать два значения: get и post;</p> <p>enctype указывает тип содержимого формы, используемый для определения формата кодирования при ее пересылке. в html определены два возможных значения для атрибута enctype: <i>application/x-www-form-urlencoded</i> (используется по умолчанию); <i>multipart/form-data</i>.</p> <p>id, class идентификаторы в пределах документа;</p> <p>lang, dir информация о языке и направленности текста;</p> <p>title заголовок элемента (выводится браузером в качестве комментария при наведении курсора на содержимое элемента);</p> <p>style встроенная информация о стиле;</p> <p>onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup, onsubmit, onreset внутренние события.</p>
<p><fieldset></p>	<p>рисует прямоугольник вокруг содержащихся в нем элементов для обозначения связанных объектов</p>	<p>accesskey задает «горячие клавиши» доступа к выделенной группе управляющих элементов;</p> <p>id, class идентификаторы в пределах документа;</p> <p>lang, dir информация о языке и направленности текста;</p> <p>title заголовок элемента (выводится браузером в качестве комментария при наведении курсора на содержимое элемента);</p> <p>style встроенная информация о стиле;</p> <p>onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup внутренние события.</p>
<p><legend></p>	<p>определяет текст, помещаемый в прямоугольник, созданный тэгом <fieldset></p>	<p>align атрибут, задающий месторасположение заголовка;</p> <p>left заголовок выводится слева на верхнем сегменте рамки;</p> <p>right заголовок выводится справа на верхнем сегменте рамки;</p> <p>accesskey задает «горячие клавиши» доступа к выделенной группе управляющих элементов;</p> <p>id, class идентификаторы в пределах документа;</p> <p>lang, dir информация о языке и направленности текста;</p> <p>title заголовок элемента (выводится браузером в качестве комментария при наведении курсора на содержимое элемента);</p> <p>style встроенная информация о стиле;</p> <p>onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress,</p>

		onkeydown, onkeyup внутренние события.
<button>	создает на странице кнопку	<p>name в качестве его значения указывается имя данного управляющего элемента;</p> <p>value через этот атрибут задается значение, которое будет передано сценарию после нажатии на кнопку;</p> <p>type этот атрибут указывает тип данной кнопки button: reset кнопка сброса;</p> <p>submit... кнопка отправки;</p> <p>button ...уникальная кнопка, обработка которой задается разработчиком.</p> <p>tabindex порядковый номер в последовательности перехода по клавише «tab»;</p> <p>disabled лог. атрибут, наличие которого «отключает» данный элемент формы;</p> <p>accesskey «горячие клавиши» доступа к данному управляющему элементу;</p> <p>onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup внутренние события.</p>
<input>	задает параметры элемента управления формы	<p>name значением этого атрибута является имя управляющего элемента, образованного данным элементом input. Указание атрибута name является обязательным, так как без него невозможно будет переслать значение управляющего элемента;</p> <p>type обязательный атрибут, определяющий тип управляющего элемента. принимает следующие значения:</p> <p>text создается поле ввода текста;</p> <p>password создается поле ввода пароля;</p> <p>checkbox создает флажок «вкл\выкл» (on\off);</p> <p>radio создает радиокнопку (кнопку с зависимой фиксацией);</p> <p>submit создает кнопку отправки. нажатие на эту кнопку отправляет все содержимое формы на сервер;</p> <p>image создает кнопку отправки в виде графического изображения;</p> <p>reset создает кнопку сброса;</p> <p>file создает управляющий элемент выбора файла;</p> <p>hidden создает скрытый элемент, не отображаемый браузером;</p> <p>size задает начальную ширину управляющего элемента;</p> <p>align выравнивание;</p> <p>id, class идентификаторы в пределах документа;</p> <p>lang, dir информация о языке и направленности текста;</p> <p>title заголовок элемента (выводится браузером в качестве комментария при наведении курсора на содержимое элемента);</p> <p>style встроенная информация о стиле;</p> <p>alt в качестве значения этого атрибута указывается</p>

		<p>краткая поясняющая информация для данного управляющего элемента;</p> <p>usemap используется для подключения клиентской навигационной карты. В качестве значения указывается имя этой карты;</p> <p>readonly логический атрибут, наличие которого запрещает редактирование;</p> <p>tabindex порядковый номер в последовательности перехода по клавише «tab»;</p> <p>disabled логический атрибут, наличие которого «отключает» данный элемент формы;</p> <p>accesskey «горячие клавиши» доступа к данному управляющему элементу;</p> <p>onfocus, onblur, onselect, onchange, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup внутренние события.</p> <p>элементы каждого типа (text, image и др) имеют свои дополнительные атрибуты</p>
<label>	определяет текст метки или заголовка для блока элементов управления	<p>for обязательный атрибут, связывающий метку с управляющим элементом. в качестве значения этого атрибута указывается заключенное в кавычки id-имя управляющего элемента, находящегося в пределах одного документа с меткой;</p> <p>id, class идентификаторы в пределах документа;</p> <p>lang, dir информация о языке и направленности текста;</p> <p>title заголовок элемента (выводится браузером в качестве комментария при наведении курсора на содержимое элемента);</p> <p>style встроенная информация о стиле;</p> <p>accesskey в качестве значения этого атрибута указываются «горячие клавиши» для быстрого доступа к данной кнопке;</p> <p>tabindex этот атрибут определяет положение данного управляющего элемента в последовательности перехода по клавише tab для текущего документа;</p> <p>onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup внутренние события.</p>
<option>	обозначает одну из альтернатив в элементе select	<p>value указывает значение, которое пересылается серверу, если из списка предлагаемых значений выбран содержащий его элемент. В том случае, если атрибут value не задан, при отправке формы в качестве результата выбора серверу будет отсылаться содержимое элемента option;</p> <p>selected логический атрибут, наличие которого делает содержащий его элемент option выбранным по умолчанию;</p> <p>disabled логический атрибут, наличие которого отключает данный элемент;</p>

		<p>id, class идентификаторы в пределах документа;</p> <p>lang, dir информация о языке и направленности текста;</p> <p>title заголовок элемента (выводится браузером в качестве комментария при наведении курсора на содержимое элемента);</p> <p>style встроенная информация о стиле;</p> <p>onfocus, onblur, onchange, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup внутренние события.</p>
<select>	определяет поле списка или выпадающий список	<p>name присваивает имя содержащему его элементу;</p> <p>size задает число одновременно видимых элементов выбора. в качестве значений принимает натуральные числа. если атрибут size установлен в значении 1, то список предлагаемых значений выбора изображается в виде ниспадающего меню. в том случае, если значение атрибута size меньше количества элементов списка выбора, то при его отображении используется вертикальная полоса прокрутки;</p> <p>multiple логический атрибут, указание которого позволяет осуществлять множественный выбор сразу нескольких предлагаемых вариантов. если этот атрибут отсутствует, то, по умолчанию, множественный выбор в данном списке значений невозможен;</p> <p>tabindex порядковый номер в последовательности перехода по клавише «tab»;</p> <p>disabled логический атрибут, наличие которого «отключает» данный элемент формы;</p> <p>onfocus, onblur, onchange внутренние события.</p>
<textarea>	обозначает поле ввода, состоящее из нескольких строк	<p>name присваивает имя элементу;</p> <p>rows задает число видимых текстовых строк. В качестве значений принимает натуральные числа;</p> <p>cols задает видимую ширину в виде количества символов средней ширины;</p> <p>id, class идентификаторы в пределах документа;</p> <p>lang, dir информация о языке и направленности текста;</p> <p>title заголовок элемента (выводится браузером в качестве комментария при наведении курсора на содержимое элемента);</p> <p>style встроенная информация о стиле;</p> <p>readonly логический атрибут, наличие которого;</p> <p>tabindex порядковый номер в последовательности перехода по клавише «tab»;</p> <p>disabled логический атрибут, наличие которого «отключает» данный элемент формы;</p> <p>onfocus, onblur, onchange, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup</p>

внутренние события.

Оборудование, программное обеспечение

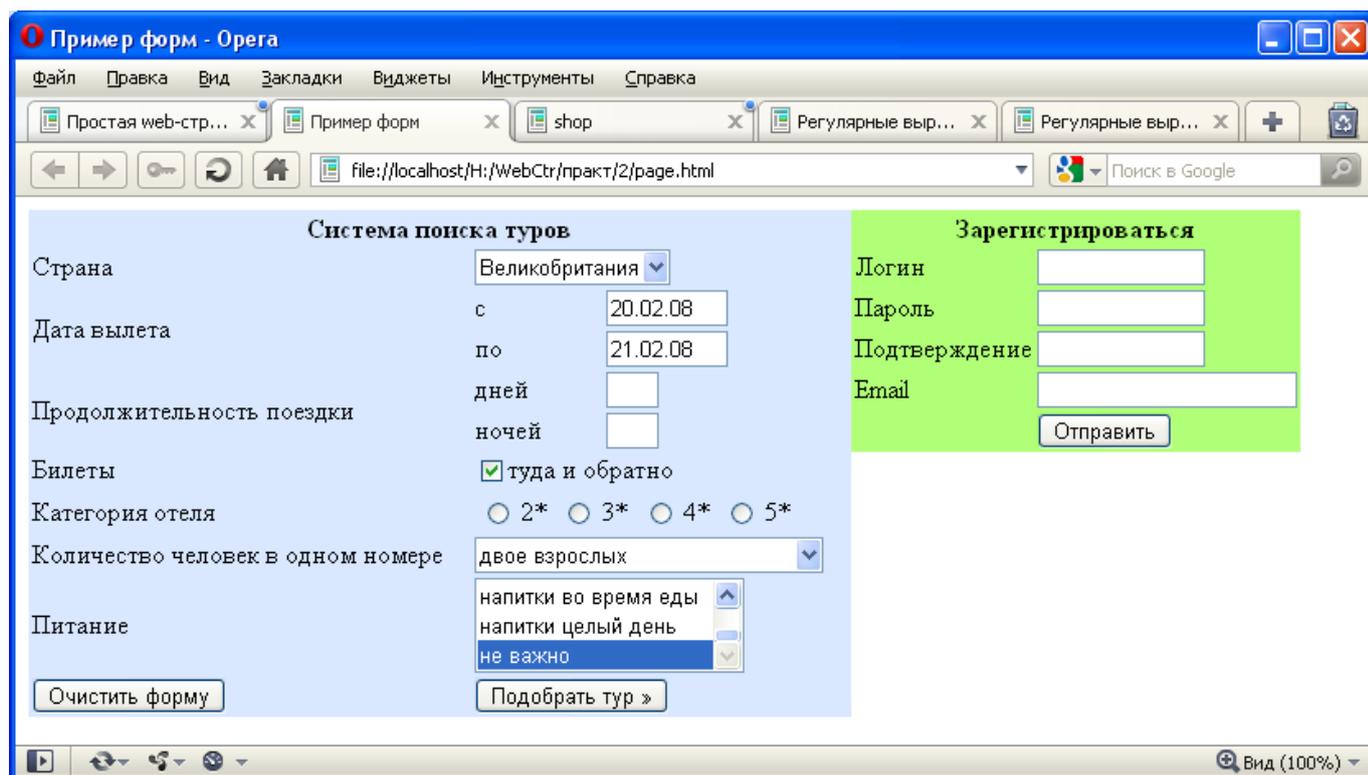
Оборудование – ПК с установленной графической операционной системой (Windows или Linux).

Программное обеспечение – текстовый редактор, например, Блокнот; графический Web-браузер, например: Internet Explorer, Mozilla Firefox, Safari, Google Chrome или Opera.

Материалы – для выполнения второго задания понадобятся графические файлы формата GIF, JPEG. Если используется браузер не Internet Explorer то, графические файлы надо будет скопировать в ту же папку, где и файл с выполняемым заданием.

Задание на работу

1. Создать HTML-документ содержащий форму подбора тура по нескольким параметрам и форму регистрации с полями логина, пароля и пр. смотри рисунок 1.



The screenshot shows a web browser window titled "Пример форм - Орега". The browser's address bar displays "file:///localhost/H:/WebCtr/практ/2/page.html". The page content is divided into two main sections:

- Система поиска туров** (Tour Search System):
 - Страна: Великобритания (dropdown)
 - Дата вылета: с 20.02.08 по 21.02.08 (date pickers)
 - Продолжительность поездки: дни (input), ночей (input)
 - Билеты: туда и обратно
 - Категория отеля: 2* 3* 4* 5*
 - Количество человек в одном номере: двое взрослых (dropdown)
 - Питание: напитки во время еды, напитки целый день, не важно (dropdown menu)
 - Buttons: "Очистить форму" and "Подобрать тур »"
- Зарегистрироваться** (Register):
 - Fields: Login, Password, Confirmation, Email (text inputs)
 - Button: "Отправить" (Submit)

Рисунок 1 – Задание 1

2. Создать файл с полем загрузки графического файла формата GIF, и полями редактирования, смотри рисунок 2.

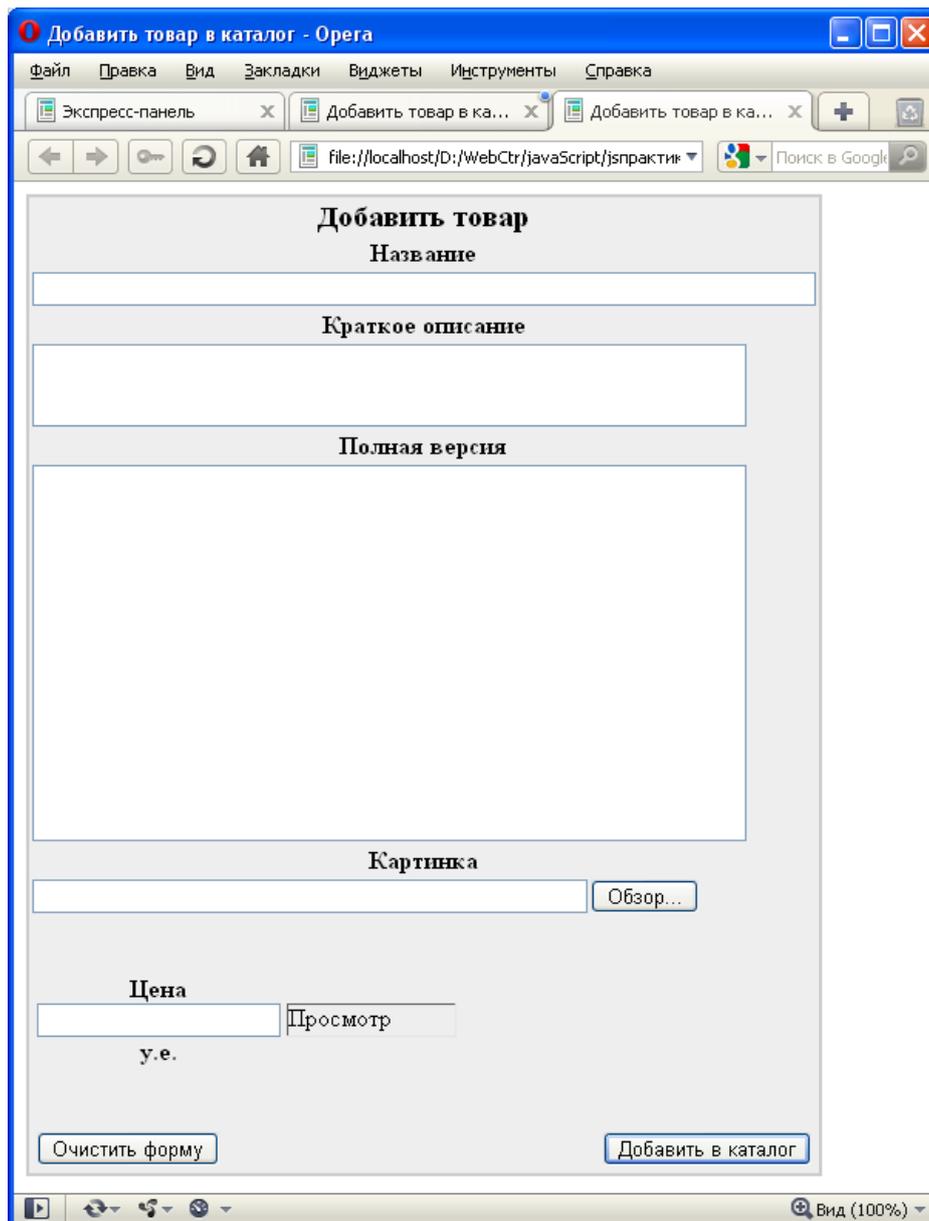


Рисунок 2 – Задание 2

Ход работы

Задание 1

1. Создайте файл form.html:

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />
<title>Пример форм</title>
</head>
<body>
<table width="500" border="0" align="left" cellpadding="1" cellspacing="1"
bgcolor="#D9E8FF">
<form action="form.php" name=formA id=formA method="post">
<tr>
```

```
<th colspan="3">Система поиска туров</th>
</tr>
<tr>
<td><label for="strana">Страна</label></td>
<td colspan="2">
<select name="strana" id="strana" tabindex="9">
<option value=1>Австрия</option>
<option value=2>Антигуа</option>
<option value=3>Багамы</option>
<option value=4>Барбадос</option>
<option value=5>Болгария</option>
<option value=6>Бразилия</option>
<option value=7 selected>Великобритания</option>
<option value=8>Венгрия</option>
<option value=9>Вьетнам</option>
<option value=10>Греция</option>
<option value=11>Доминикана</option>
<option value=12>Египет</option>
<option value=13>Израиль</option>
<option value=14>Индия</option>
<option value=15>Индонезия</option>
<option value=16>Иордания</option>
<option value=17>Испания</option>
<option value=18>Италия</option>
<option value=19>Кипр</option>
</select> </td>
</tr>
<tr>
<td rowspan="2">Дата вылета</td>
<td>c </td>
<td><input type="text" name="first" id="first" accesskey="f" tabindex="1"
value="20.02.08" size="8" maxlength="8"></td>
</tr>
<tr>
<td>по </td>
<td><input type="text" name="end" id="end" tabindex="2" value="21.02.08" size="8"
maxlength="8"></td>
</tr>
<tr>
<td rowspan="2">Продолжительность поездки</td>
<td>дней </td>
<td><input type="text" name="first" id="first" tabindex="3" size="2"
maxlength="2"></td>
</tr>
<tr>
<td>ночей</td>
<td><input type="text" name="end" id="end" tabindex="4" size="2"
maxlength="2"></td>
</tr>
<tr>
```

```

<td>Билеты</td>
<td colspan="2"><input name="chek_b" type="checkbox" value="" checked
tabindex="12">туда и обратно</td>
</tr>
<tr>
<td>Категория отеля</td>
<td colspan="2">
<table width="200">
<tr>
<td><label>
<input type="radio" name="RadioGroup1" value="radio" id="RadioGroup1_0"
tabindex="8">
2*</label></td>
<td><label>
<input type="radio" name="RadioGroup1" value="radio" id="RadioGroup1_1"
tabindex="7">
3*</label></td>
<td><label>
<input type="radio" name="RadioGroup1" value="radio" id="RadioGroup1_2"
tabindex="6">
4*</label></td>
<td><label>
<input type="radio" name="RadioGroup1" value="radio" id="RadioGroup1_3"
tabindex="5">
5*</label></td>
</tr>
</table> </td>
</tr>
<tr>
<td>Количество человек в одном номере</td>
<td colspan="2"><select name="numb" id="numb" tabindex="10">
<option value=1 selected>двое взрослых</option>
<option value=2>двое взрослых с ребенком</option>
<option value=3>двое взрослых с двумя детьми</option>
<option value=4>один взрослый</option>
<option value=5>взрослый с ребенком</option>
<option value=6>взрослый с двумя детьми</option>
<option value=7>трое взрослых</option>
<option value=8>трое взрослых с ребенком</option>
<option value=9>трое взрослых с двумя детьми</option>
</select></td>
</tr>
<tr>
<td>Питание</td>
<td colspan="2"><select name="eat[]" id="eat" multiple size=3 tabindex="11">
<option value="0">завтрак</option>
<option value="1">обед</option>
<option value="2">ужин</option>
<option value="3">дополнительное питание</option>
<option value="4">напитки во время еды</option>

```

```

    <option value="5">напитки целый день</option>
    <option value="6" selected>не важно</option>
  </select> </td>
</tr>
<tr>
  <td><input name="" type="reset" value="Очистить форму"></td>
  <td colspan="2"><input name="goto" type="submit" value="Подобрать тур
&raquo;"></td>
</tr></form>
</table>
<table width="230" border="0" align="left" cellpadding="1" cellspacing="1"
bgcolor="#B0FF77"><form action="pass.php" name=form_pas id=form_pas
method="post">
  <tr>
    <th colspan="2">Зарегистрироваться</th>
  </tr>
  <tr>
    <td>Логин</td><td><input type="text" name="login" id="login" size="12"
maxlength="11"></td>
  </tr>
  <tr>
    <td>Пароль</td><td><input type="password" name="pass_1" id="pass_1" size="12"
maxlength="11"></td>
  </tr>
  <tr>
    <td>Подтверждение</td><td><input type="password" name="pass_2" id="pass_2"
size="12" maxlength="11"></td>
  </tr>
  <tr>
    <td>Email</td><td><input type="text" name="login" id="login" size="20"></td>
  </tr>
  <tr>
    <td><input type="hidden" name="ses" id="ses" value="session"></td>
    <td><input type="button" name="pas_go" id="pas_go" onClick="submit()"
value="Отправить">
  </td>
  </tr>
</form>
</table>
</body>
</html>

```

2. Проверьте работу странички в браузере.

Задание 2

Указание к выполнению задания

1. Скопируйте к себе в папку несколько картинок из папки img_m.
2. В своей папке создайте документ add_new.html

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
<title>Добавить товар в каталог</title>

```

```

<script>
function show_img(){
  if(document.formN.file.value!="")
  document.images[0].src=document.formN.file.value
}
</script>
</head>
<body>
<div style="width:480px; float:left; padding:0px; margin:0px">
<form name="formN" action="add_pr.php" id="formN" method="post" cc
style="padding:0px; margin:0px">
  <table width="98%" border="0" cellspacing="1" cellpadding="1" style="font-size:12pt;
border:2px solid #CCC; background-color:#EEE">
    <tr><th><h3>Добавить товар</h3></th></tr>
    <tr><th>Название</th></tr>
    <tr><td><input type="text" size="70" name="title" id="title" maxlength="70" /></td></tr>
    <tr><th>Краткое описание</th></tr>
    <tr><td><textarea rows="3" cols="55" name="short_story"
id="short_story"></textarea></td></tr>
    <tr><th>Полная версия</th></tr>
    <tr><td><textarea rows="15" cols="55" name="full_story"
id="full_story"></textarea></td></tr>
    <tr><th>Картинка</th></tr>
    <tr><td><input type="file" accept="image/gif" name="file" size="50"
onchange="show_img()"/></td></tr>
    <tr><td><table width="100%">
    <tr><th width="100%">Цена <input type="text" size="20" name="cast" id="cast"
maxlength="20"/> y.e.</th>
    <td height="130"><img src="" alt="Просмотр" width="110" /></td>
    </tr></table>
    </td>
    <tr>
    <td><table width="100%"><tr>
    <td><input type="reset" name="button" id="button" value="Очистить форму" /></td>
    <td style="text-align:right"><input name="goto" type="submit" value='Добавить в
каталог' /></td></tr>
    </table></td>
    </tr>
  </table>
</div>
</body>
</html>

```

3. Проверьте работу странички в браузере.

Содержание отчета

Отчет по лабораторной работе должен содержать: конспект по теоретической части, отчет об используемых в работе тегах.

К отчетам должны иметься выполненные в ходе работы файлы.

4 Каскадные таблицы стилей. CSS

Цель работы – ознакомление с синтаксисом и способами включения каскадных таблиц стилей.

Задачи работы – получение навыков форматирования HTML-документа каскадными таблицами стилей.

Теоретические сведения

Язык CSS предназначен для описания способа отображения содержимого документов с помощью устройств вывода. Существует возможность создавать такие описания с учётом специфики этих устройств.

В настоящее время существует три версии стандарта CSS: CSS1, CSS2 и CSS3.

Способы связывания файлов в формате CSS с HTML-файлами и способы встраивания фрагментов CSS-кода в HTML-документы

1. У всех тэгов, кроме <base>, <basefont>, <head>, <html>, <meta>, <param>, <script>, <style> и <title>, можно указать атрибут style.

Пример: <p style="color:red"> задаёт абзац, в котором символы будут иметь красный цвет.

2. В заголовке HTML-документа (внутри элемента head) можно вставить фрагмент CSS-кода внутри тэга <style>. Его атрибуты:

type задаёт тип языка (это всегда text/css).

media задаёт тип устройства. Возможные значения:

all (по умолчанию) соответствует любому устройству; tty соответствует устройствам с моноширинным шрифтом (некоторые принтеры, факсы и т. п.);

tv соответствует телевизору (маленькое разрешение, низкое количество цветов, проблемы скроллинга)

projection соответствует проекторам

hldheld соответствует карманным устройствам (малое разрешение, экран, цвета и т. п.)

print соответствует режиму preview (для печати)

braille соответствует устройствам для слепых

aural соответствует синтезаторам речи

Пример:

```
<style type="text/css" media="all">
```

```
<!--
```

```
p {color:red}
```

```
-->
```

```
</style>
```

HTML-комментарий здесь используется для того, чтобы фрагмент CSS-кода был проигнорирован в браузерах, не поддерживающих тэг <style>.

3. Можно связать файл с текстом в формате CSS с текущим HTML-документом с помощью тэга <link>. С помощью него можно связывать HTML-документ не только с CSS-файлами, но и файлами любой природы. Приведём список его атрибутов: charset, href, hreflang, type, rel, rev, target предназначены для тех же целей, что и одноименные атрибуты тэга <a>; атрибут media задаёт тип устройства (используется для text/css,

например). Для случая стилевых таблиц используют `rel="stylesheet"` или `rel="alternate stylesheet"`. Указывая значение атрибута `title`, можно его использовать для задания предпочитаемого стиля в HTTP-заголовке или тэге `<meta>`.

Пример:

```
<meta http-equiv="Default-Style" content="Compact" />
<link href="style.css" rel="stylesheet" title="Compact" />
```

Среди приведённых способов задания информации следует отдавать предпочтение описаниям во внешних CSS-файлах, так как это будет облегчать задание общего стиля документам всего сайта.

Синтаксис языка CSS

Комментарии в языке CSS начинаются с `/*` и заканчиваются `*/`.

Текст на языке CSS состоит из последовательности конструкций вида:

```
селектор {
    свойство1:значение;
    свойство2:значение;
    ...
    свойствоN:значение
}
```

Сначала приведём способы задания селекторов в языке CSS.

Селекторы

В качестве селектора можно указать тип элемента (тэга) HTML. В этом случае описание будет относиться ко всем вхождениям этого тэга. Можно указать, что описание относится ко всем тэгам с помощью знака `*`. Пример:

```
h1 {text-align:center}
```

Здесь задаётся выравнивание по центру для всех вхождений тэга `<h1>`

В одном селекторе можно через запятую указать несколько элементов:

```
h1,h2,h3,h4,h5,h6 {text-align:center}
```

В качестве селектора можно указать название класса элемента (он указывается после точки). Именно, в любом тэге, где можно указать атрибут `style`, можно указать класс с помощью атрибута `class`. Название класса зависит от регистра символов. Можно указать несколько классов через пробел (как значение атрибута `class`). Несколько названий классов в селекторе можно указать, каждый раз отделяя их точкой. Например:

```
.test {text-align:right}
```

Перед точкой можно указать название элемента.

Селектор можно дополнить значением атрибута HTML в квадратных скобках.

Например:

```
h1[lang=en] {text-align:left}
```

Если указать только название атрибута, то в тэге должен быть использован этот атрибут (с любым значением или без него), если указано вместо равно `~=`, то ищется одно из значений, указанных через пробел (внутри описания атрибута тэга). Если указано `|=`, то ищется значение до дефиса (используется в `lang`).

В качестве селектора можно указать значение атрибута `id` после знака диэза.

Например:

```
*#copy {text-align:right}
```

Селектор может указывать на вложенный тэг. В этом случае указываются через пробел элементы. Описание будет относиться к вложенному тэгу. Пример:

```
td div {color: blue}
```

Указание дочернего элемента (непосредственно подчиненного тэга). Пример:

```
td> p {color: blue}
```

Сначала указывается родительский элемент, далее знак > (без пробела), за ним идёт один или несколько пробелов, далее имя дочернего элемента. Кроме того после двоеточия можно указать first-child (для элемента являющегося первым дочерним).

Можно указать элемент, который непосредственно следует за другим элементом с помощью плюса. Пример:

h1+h2 {color: blue}

Для гиперссылок можно указать следующие псевдоклассы (их названия указывают после двоеточия):

link обычная гиперссылка

visited, гиперссылка, по которой происходил переход

active - активная гиперссылка (выделена щелчком)

hover - гиперссылка, над которой находится курсор

focus - гиперссылка, находящаяся в фокусе

Указание псевдоэлементов (также после двоеточия):

first-line для тэга <p> выбирает первую строку

first-letter для тэга <p> выбирает первый символ

before указывает текст, который будет вставлен до элемента

after указывает текст, который будет вставлен после элемента

Дополнительно можно указать язык после двоеточия, например: h1:lang(ru)

@-команды

@charset указывает используемую таблицу символов (например, @charset "windows-1251")

@import задаёт загрузку документа CSS (и последующую обработку). Далее идёт либо просто url документа, либо внутри конструкции url. Пример:

@import url("my.css")

В конце можно указать тип устройства. Можно использовать только одну конструкцию import для одного устройства.

Указание типа носителя (@media). Далее идёт тип носителя (или несколько через запятую). Затем в фигурных скобках идёт текст, относящийся к такому носителю.

Способы задания констант

Для указания размеров используют следующие единицы: in (дюймы), cm (сантиметры), mm (миллиметры), pt (1/72 дюйма), pc (12pt), em (размер шрифта), ex (размер буквы "x"), px (размер пикселя). Собственно размер задаётся в виде десятичного числа с разделителем в виде точки. Перед числом может стоять знак + или -.

Единицы измерения

Обозначения	Название	Значение
in	дюйм	25,4 mm
mm		
cm		
pc	пика	1/6 in
pt	пункт	1/72 in
px	пиксель	0,24 0,35 mm
em	m ширина	длина шрифта
ex	x высота	высота шрифта

URL задаются в виде конструкции url(). Внутри скобок в кавычках задаётся собственно адрес.

Цвет задаётся либо названием, либо числовым кодом (#шестнадцатиричный код), либо в виде тройки (rgb(255,255,255)). Возможно также использовать системные значения: ActiveBorder - рамка активного окна, ActiveCaption - заголовок активного

окна, AppWorkspace - цвет фона многодокументного интерфейса, Background - фон рабочего стола, ButtonFace - цвет переднего плана 3-мерных элементов, ButtonHighlight - тёмная тень для 3-мерных элементов (для краёв, выступающих из светлой основы), ButtonShadow - цвет тени для 3-мерных элементов, ButtonText - текст кнопки. CaptionText - текст заголовка, бокса и в боксе прокручиваемого списка, GrayText - серый ("недоступен") текст. Это цвет установлен в #000 на тот случай, если текущий дисплей не поддерживает сплошной серый цвет, Highlight - объект(ы), выделенный в элементе управления, HighlightText - текст объекта(ов), выделенного в элементе управления, InactiveBorder - рамка неактивного окна, InactiveCaption - заголовок неактивного окна. InactiveCaptionText - цвет текста неактивного заголовка, InfoBackground - цвет фона элементов подсказки, InfoText - цвет текста элементов подсказки, Menu - фон меню, MenuItemText - текст меню, Scrollbar - серая область прокрутки, ThreeDDarkShadow - тёмная тень для 3-мерных элементов дисплея, ThreeDFace - цвет переднего плана для 3-мерных элементов дисплея, ThreeDHighlight - цвет подсветки для 3-мерных элементов дисплея, ThreeDLightShadow - светлый цвет для 3-мерных элементов дисплея (для краёв, выступающих из светлой основы), ThreeDShadow - тёмная тень для 3-мерных элементов дисплея, Window - фон окна, WindowFrame - кадр окна, WindowText - текст в окнах.

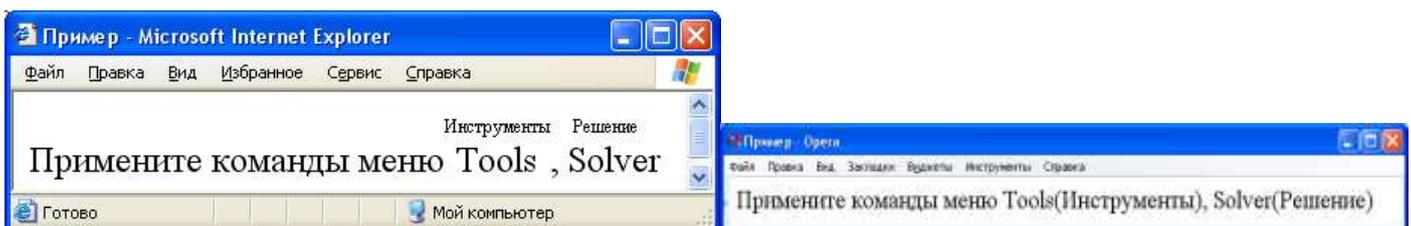
Оборудование, программное обеспечение

Оборудование – ПК с установленной графической операционной системой (Windows или Linux).

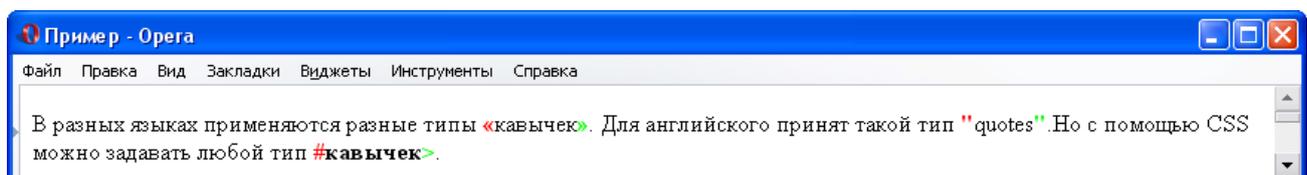
Программное обеспечение – текстовый редактор, например, Блокнот или MS SharePoint Designer ; Web-браузеры Internet Explorer и Opera.

Задание на работу

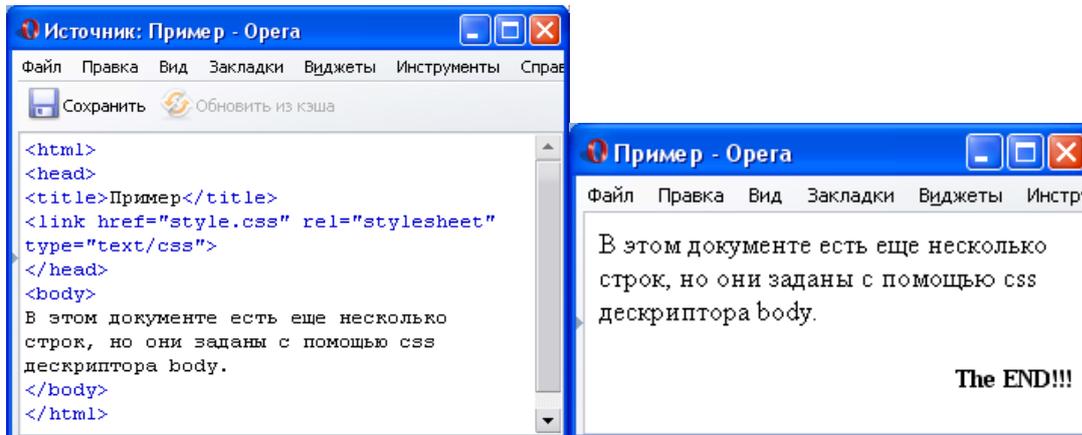
1. На рисунке приведен вид одного и того же документа в браузерах Opera и IE. Приведите исходный код тела документа (использовались только тэги).



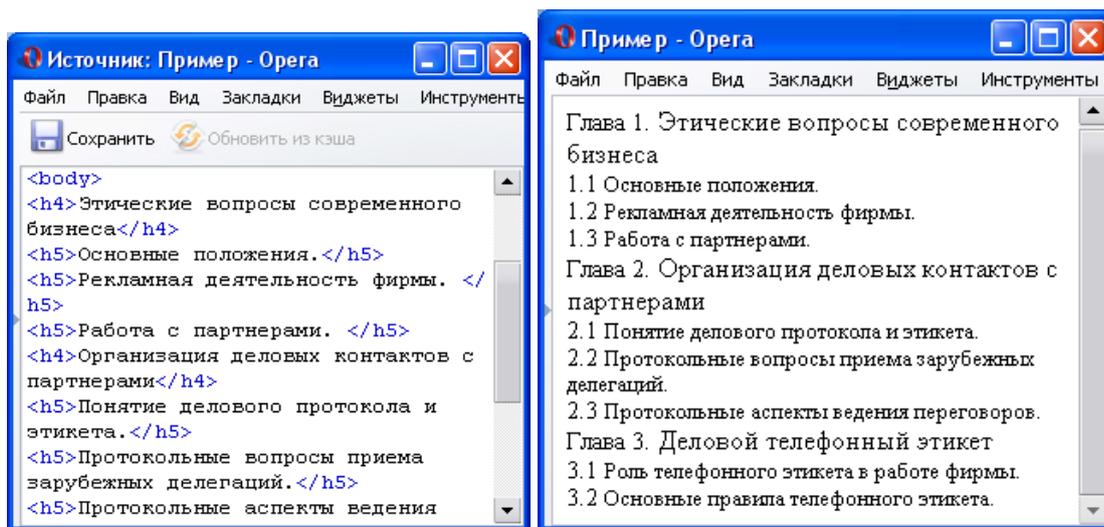
2. Приведите код css для дескриптора q. Код строки: `<p lang="ru">В разных языках применяются разные типы <q>кавычек</q>. Для английского принят такой тип <q>quotes</q>.Но с помощью CSS можно задавать любой тип <q>кавычек</q>.</p>`



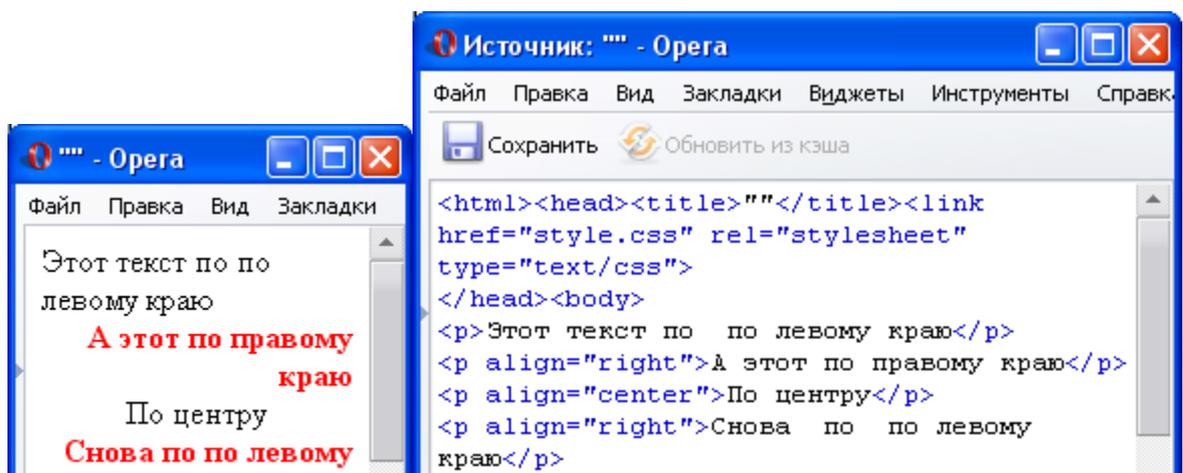
3. В конце документа в браузере показывается строка “The END!!!”. Приведите код заданный в style.css.



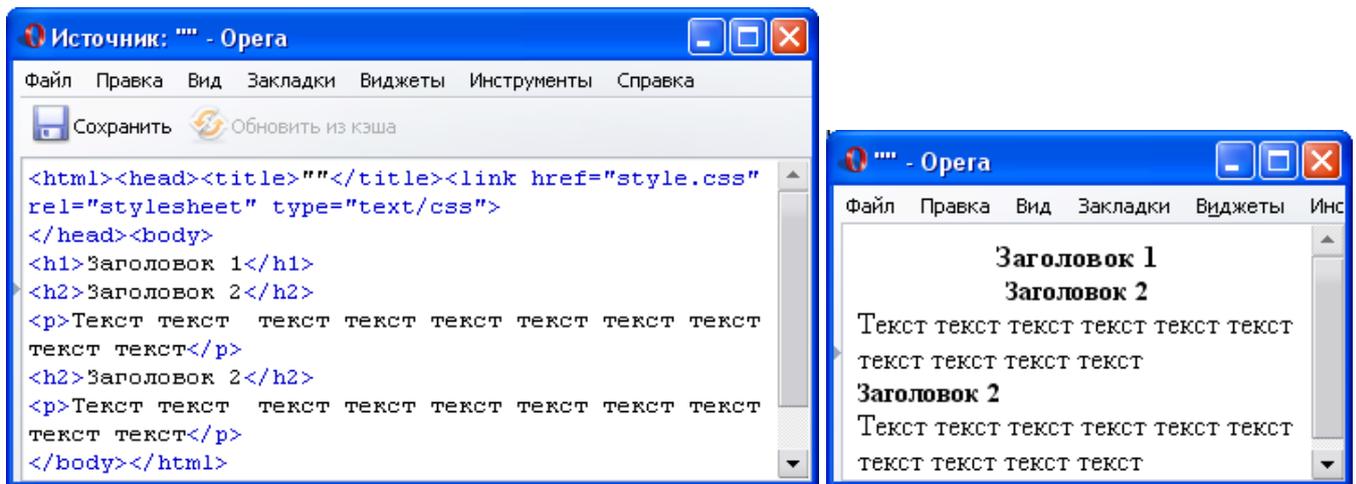
4. Заголовки h4, h5 документа автоматически нумеруются как сложный список. В качестве разделителя используется точка, для заголовка h4 добавляется название “Глава”. Приведите код для h4, h5 заданный в style.css.



5. Текст абзаца, выровненного по правому краю, имеет красный шрифт и полужирное начертание. Приведите код заданный в style.css для абзаца, выровненного по правому

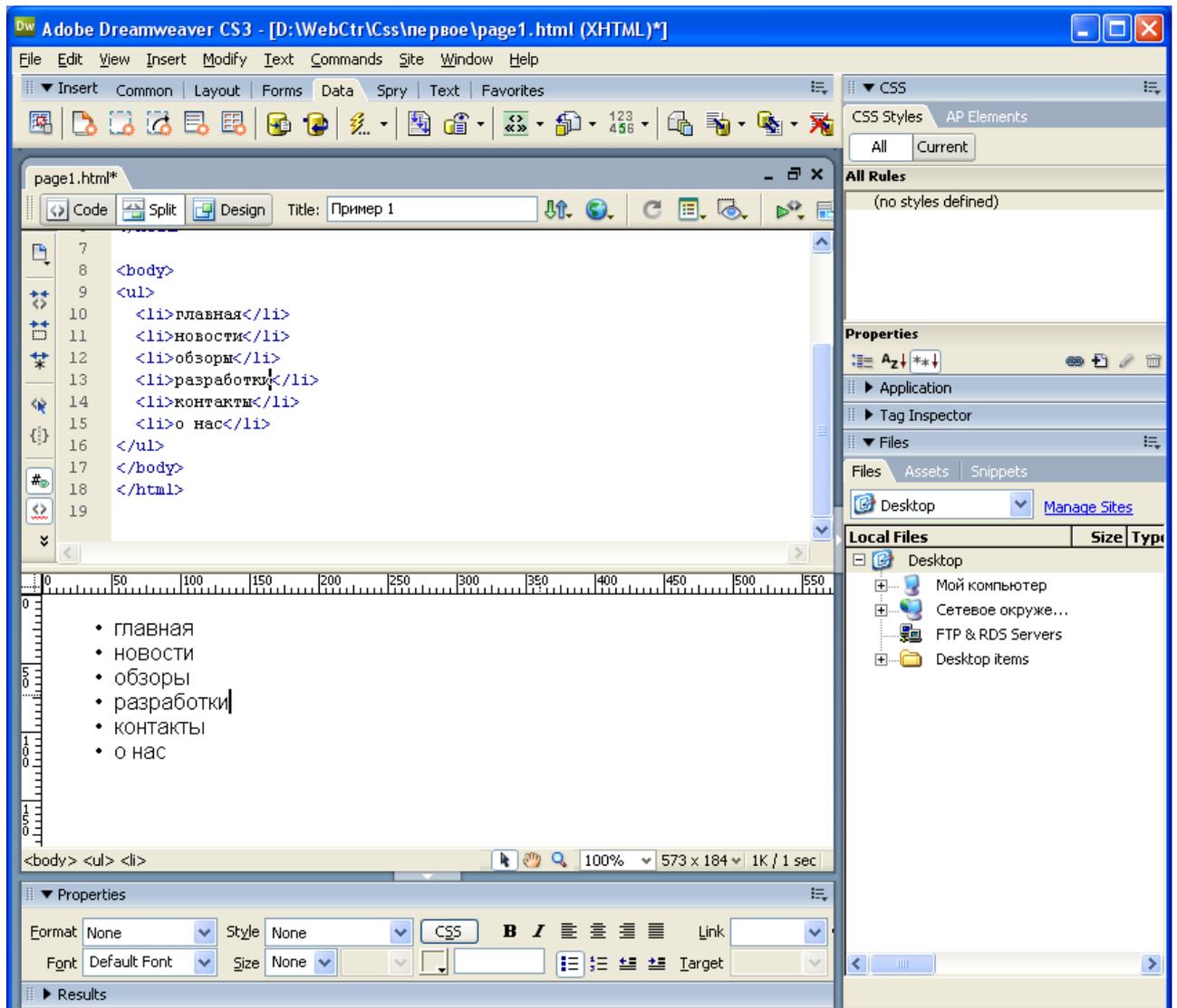


6. Заголовок h2, если он следует за заголовком h1, выравнивается по центру, в остальных случаях используется значение по умолчанию. Приведите код заданный в style.css

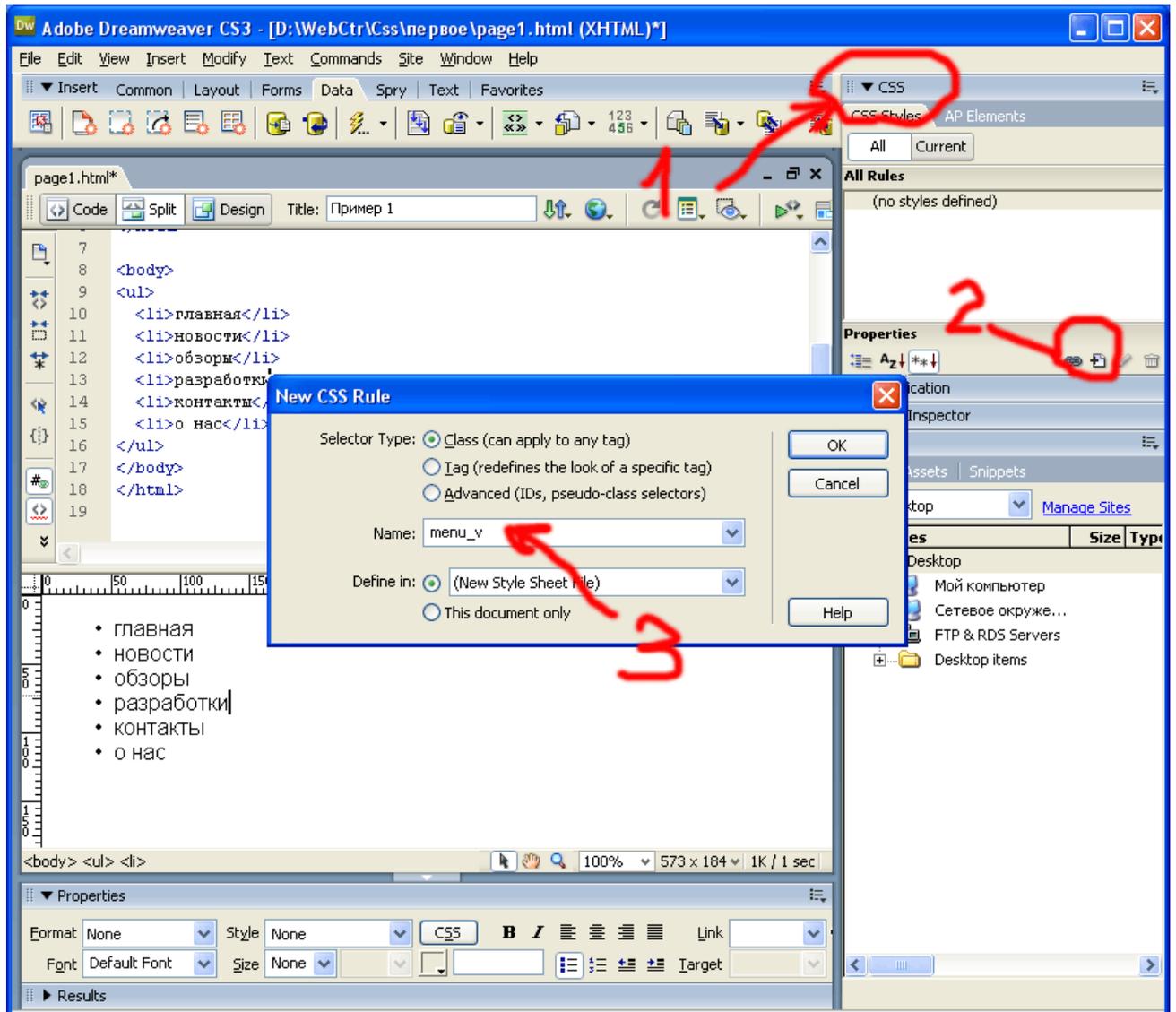


1. Ознакомьтесь с теоретической частью. Наберите код html-страницы.
2. Создайте в MS SharePoint Designer новый HTML документ. Сохраните, например, как page_1.html.

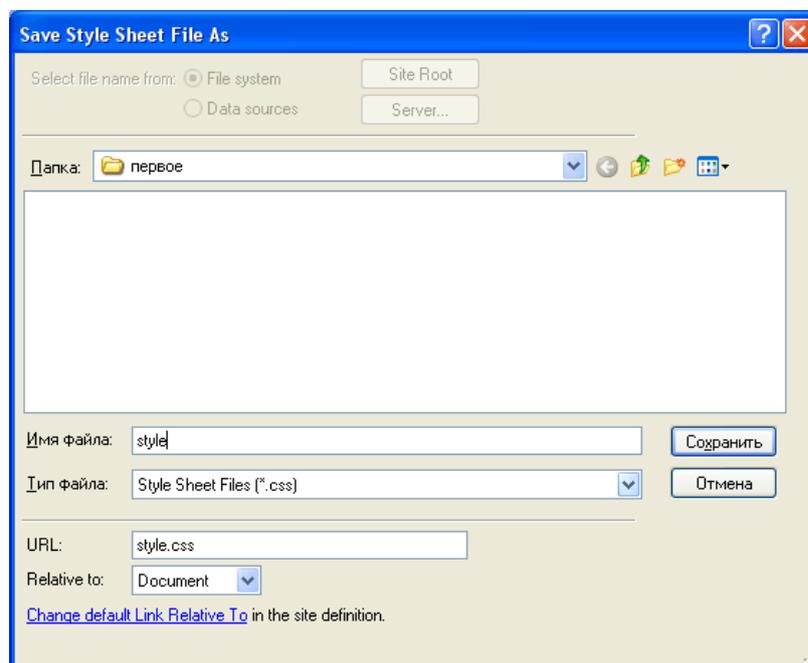
3. Задайте режим отображения Split. Наберите, если надо, html код см. рисунок ниже.



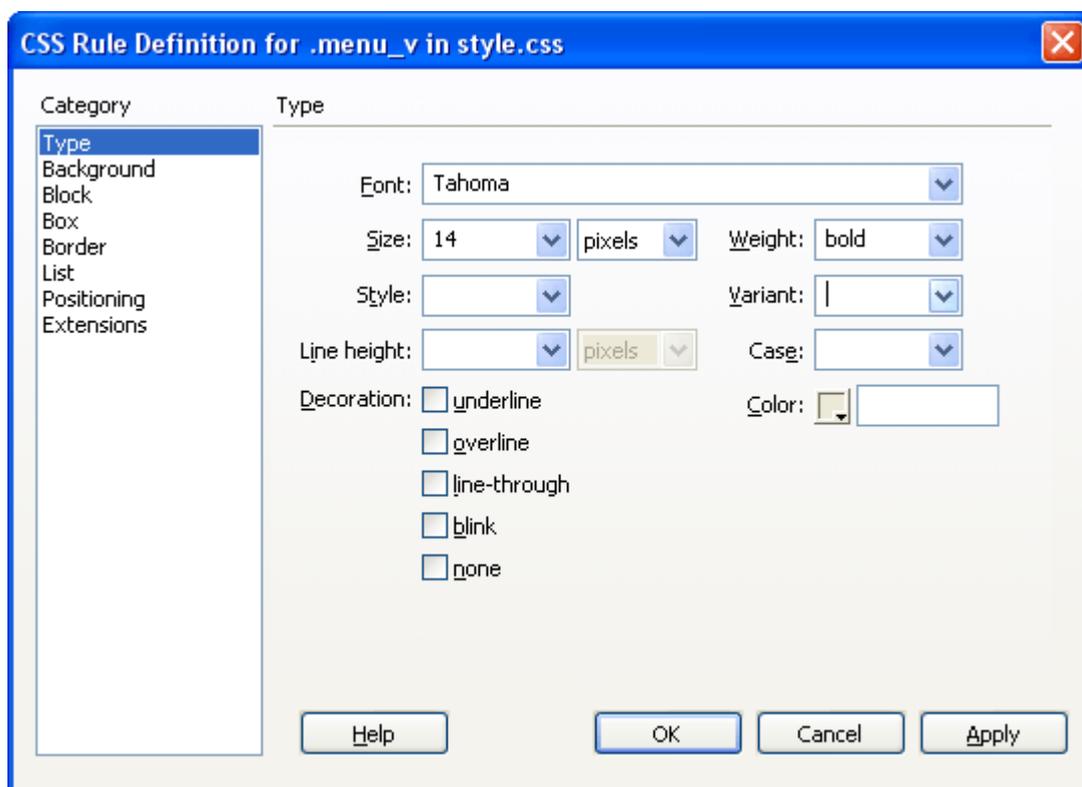
4. На панели CSS выберите New CSS Rule, задайте например имя класса Selector type/class и Name: menu_v (см. рисунок ниже), или выберите нужный тэг, например Tag: body.



5. Сохраните как style: style.css (расширение .css добавлять не надо, MS SharePoint Designer сам добавит).



6. Далее задайте необходимые параметры.



Содержание отчета

Выполненные в ходе работы файлы.

5 Создание элементов страниц. Создание меню

Цель работы – ознакомление с принципами создания элементов страниц.

Задачи работы – получение навыков создания базовых элементов страниц.

Теоретические сведения

Горизонтальное меню навигации частый элемент дизайна многостраничных сайтов (см. рисунок 1). Консорциум Wide Web Consortium (W3C), разрабатывающий стандарты для всемирной паутины, рекомендует использовать семантическую верстку при создании элементов страниц размещаемых в сети. Требования, предъявляемые к семантическому, коду обязывают делать код понятным для человека. Согласно этим рекомендациям, меню оформляется в виде списка. Такое оформление меню в виде списка гиперссылок позволит человеку пользоваться им, в случае если css и выполнение скриптов недоступно, а также лучше индексируется поисковыми системами. Для придания меню необходимого вида используются каскадные таблицы стилей.

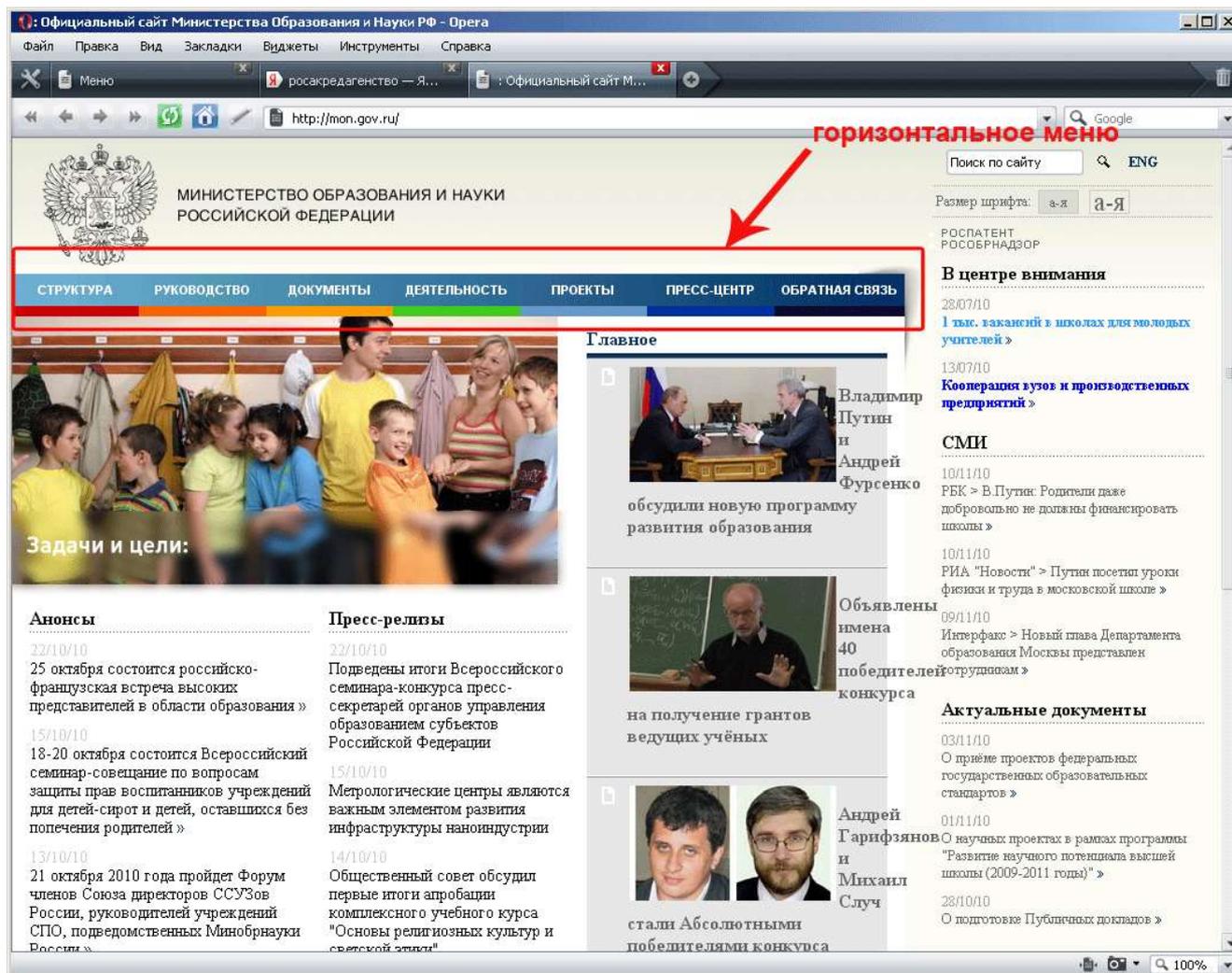


Рисунок 1 - Пример сайта с горизонтальным меню
Оборудование, программное обеспечение

Оборудование – ПК с установленной графической операционной системой (Windows или Linux).

Программное обеспечение – текстовый редактор, например, Блокнот, MS SharePoint Designer ; графический Web-браузер, например: Internet Explorer, Mozilla Firefox, Safari, Google Chrome или Opera.

Задание на работу

Создать горизонтальное меню для списка:
 На главную
 Каталог
 Регистрация
 О компании.

Ход работы

1.1 На первом этапе надо создать список ссылок на страницы сайта. Например: создайте файл menu.html. Наберите между <body> и </body> следующий код:

```
<ul id="main_menu">
  <li><a href="#">На главную</a></li>
  <li><a href="catalog.html">Каталог</a></li>
  <li><a href="registration.html">Регистрация</a></li>
  <li><a href="about.html">О компании</a></li>
</ul>
```

Обратите внимание меню присвоен уникальный идентификатор main_menu. В браузере код будет выглядеть следующим образом см. рисунок 2.

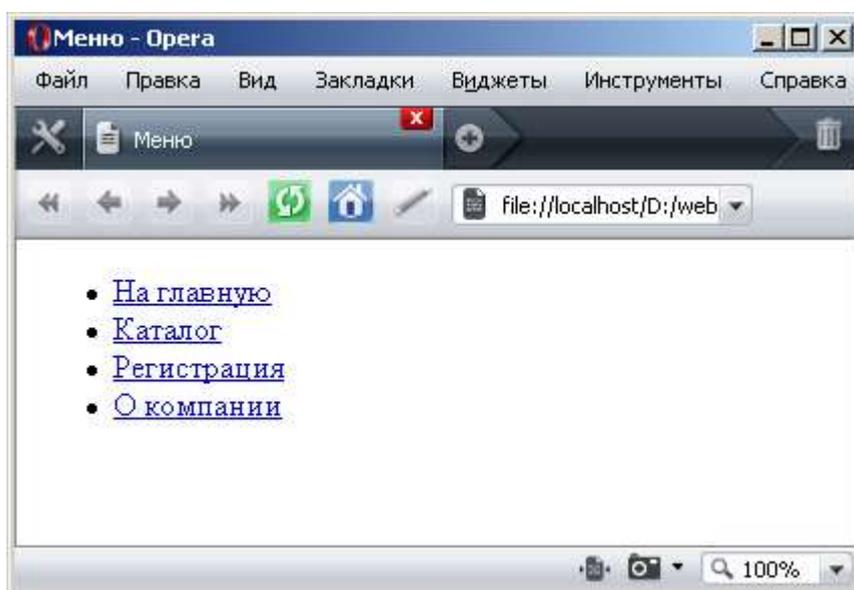


Рисунок 2

1.2 Добавляем каскадные таблицы стилей. Для сайта удобно и правильно (!) использовать таблицы стилей в отдельном файле css. Присоединяем каскадную таблицу стилей к menu.html: ставим курсор в main_menu и на панели CSS выбираем команду New CSS Rule (Новое CSS правило) см. рисунок 3.

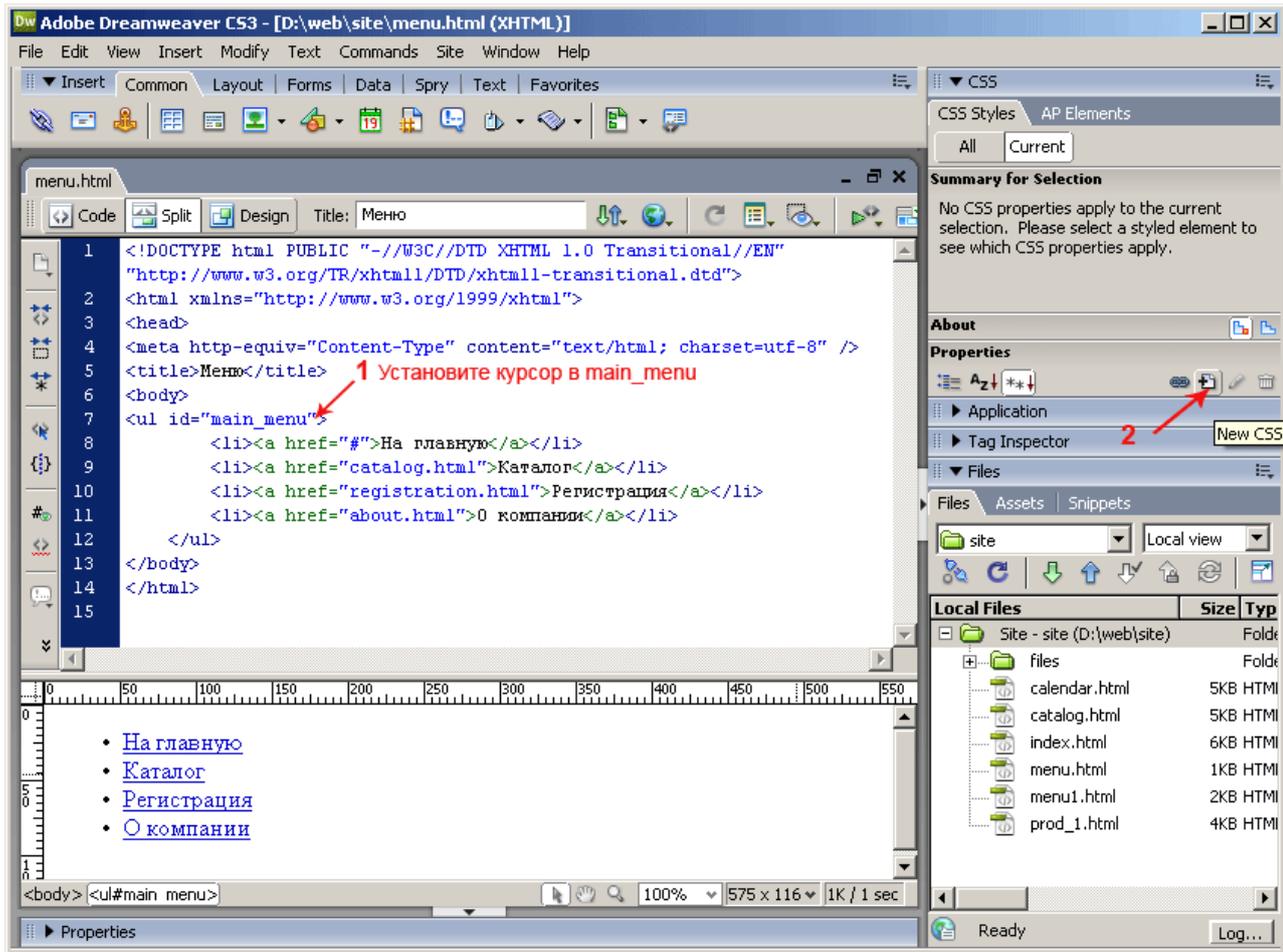


Рисунок 3

В появившемся диалоговом окне (рисунок 4) будет уже набрано имя селектора #main_menu (MS SharePoint Designer вписывает имя селектора или тега где находился курсор момент вызова команды New CSS Rule). Выберите New Style Sheet File если не выбрано, дальше ОК.

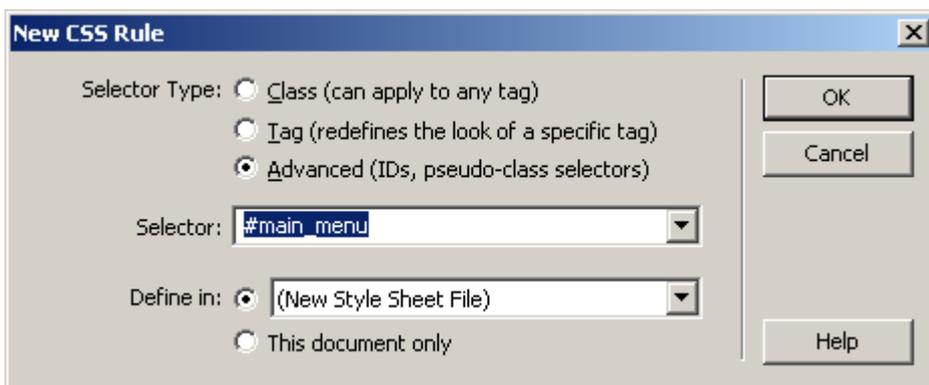


Рисунок 4

Назовите файл style_menu.css.

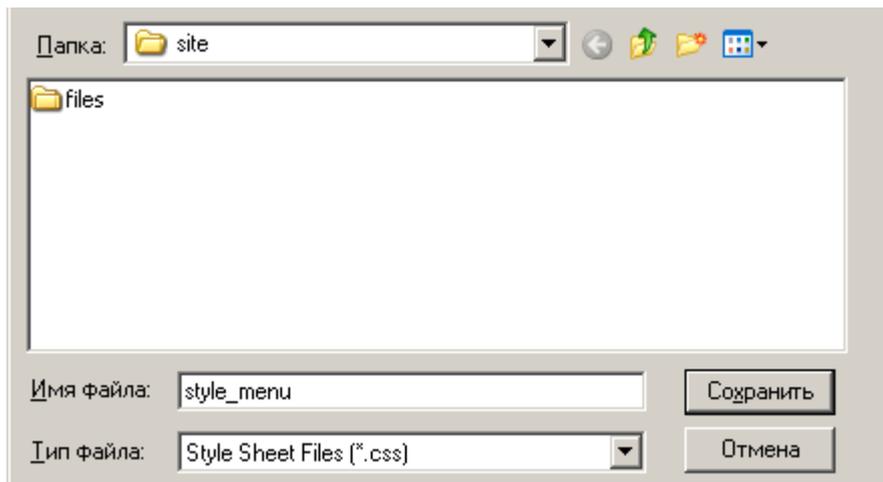


Рисунок 5

Задайте параметры на вкладке Box Float:left Margin:0 (см. рисунок 6). На вкладке List Type:none (см. рисунок 7).

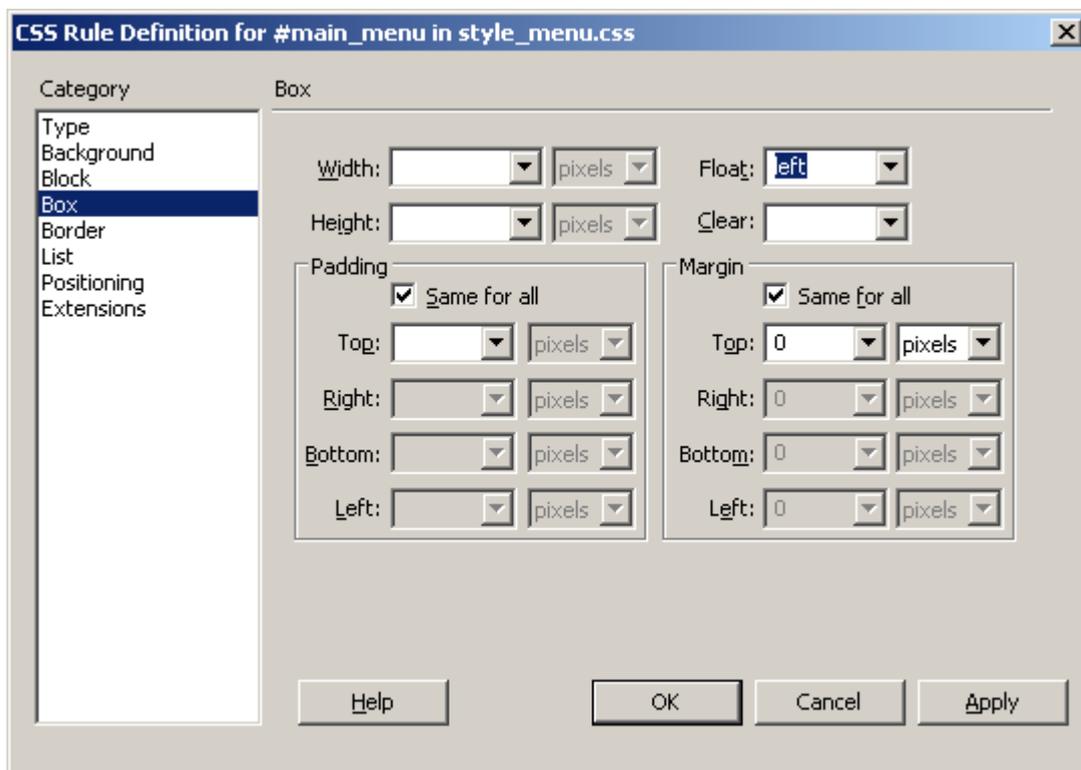


Рисунок 6

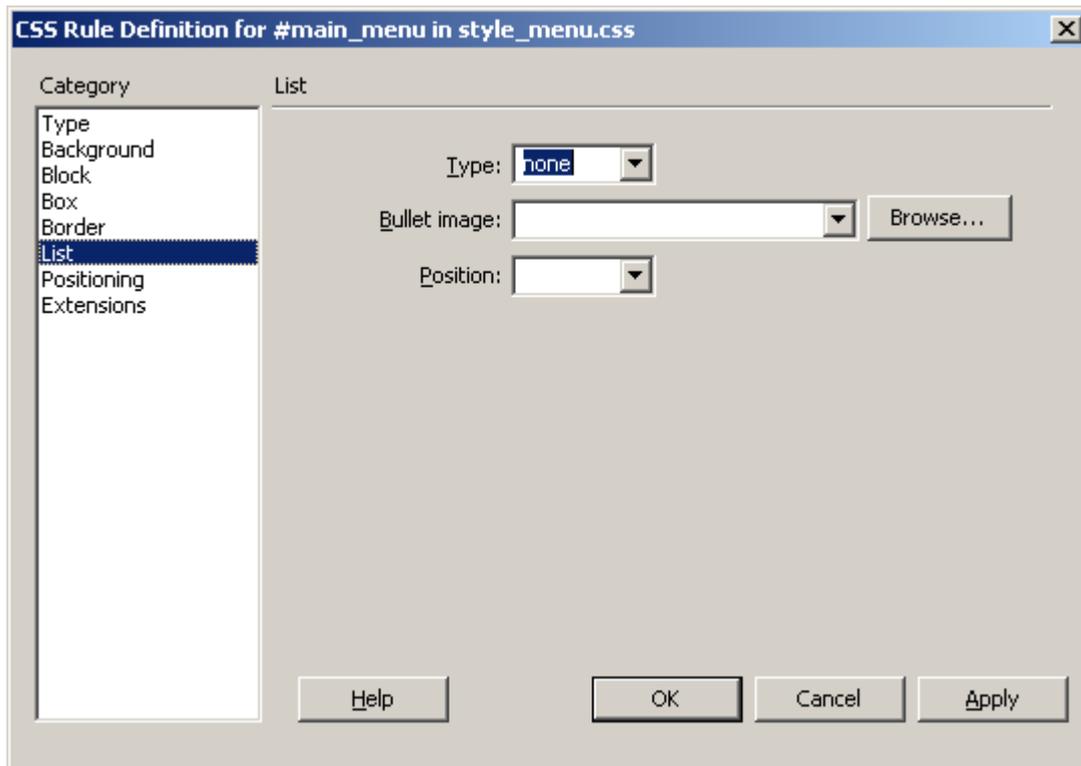


Рисунок 7

Должен получиться следующий код html см. рисунок 8.

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4 <head>
5 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6 <title>Меню</title>
7 <link href="style_menu.css" rel="stylesheet" type="text/css" />
8 <body>
9 <ul id="main_menu">
10     <li><a href="#">На главную</a></li>
11     <li><a href="catalog.html">Каталог</a></li>
12     <li><a href="registration.html">Регистрация</a></li>
13     <li><a href="about.html">0 компаний</a></li>
14 </ul>
15 </body>
16 </html>

```

Рисунок 8

В коде html добавилась строчка, связывающая каскадную таблицу стилей с файлом html:

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

Код каскадной таблицы стилей style_menu.css рисунок 9.

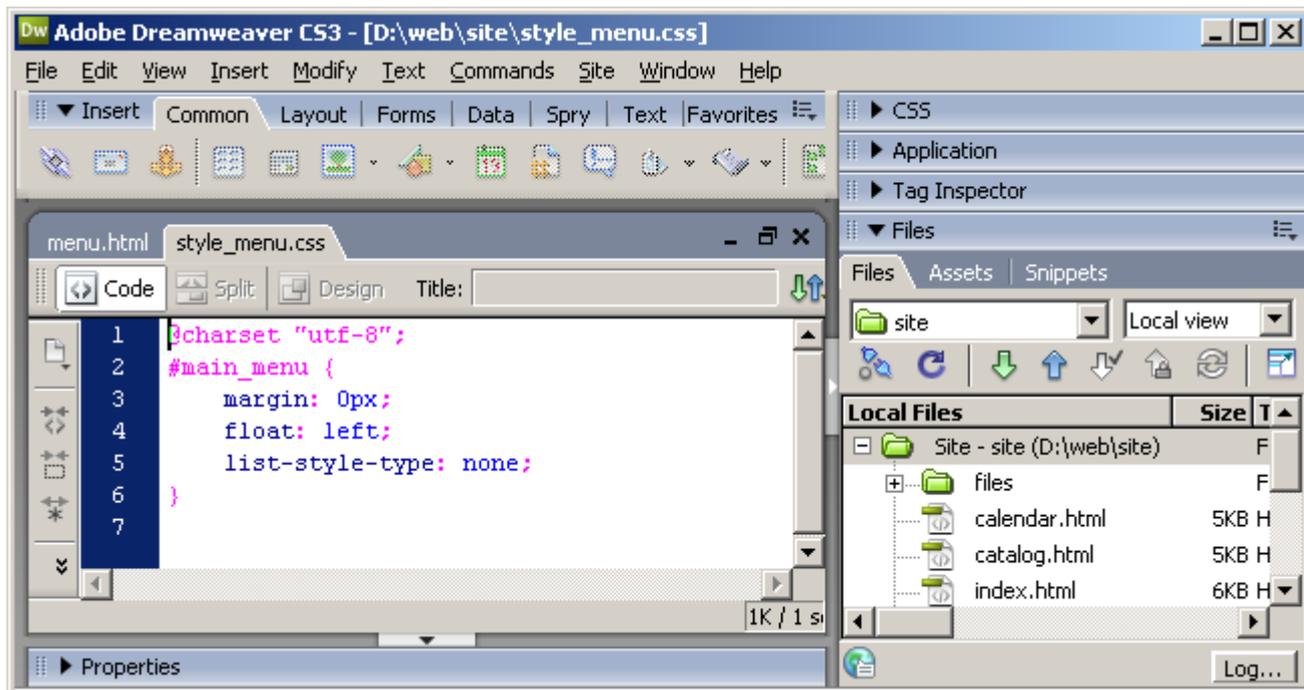


Рисунок 9

Строчкой `margin:0px`, мы убрали поля, задавая `list-style-type: none`; мы убрали маркеры, `float: left`; сделали меню блочным элементом, плавающим слева см. рисунок 10.

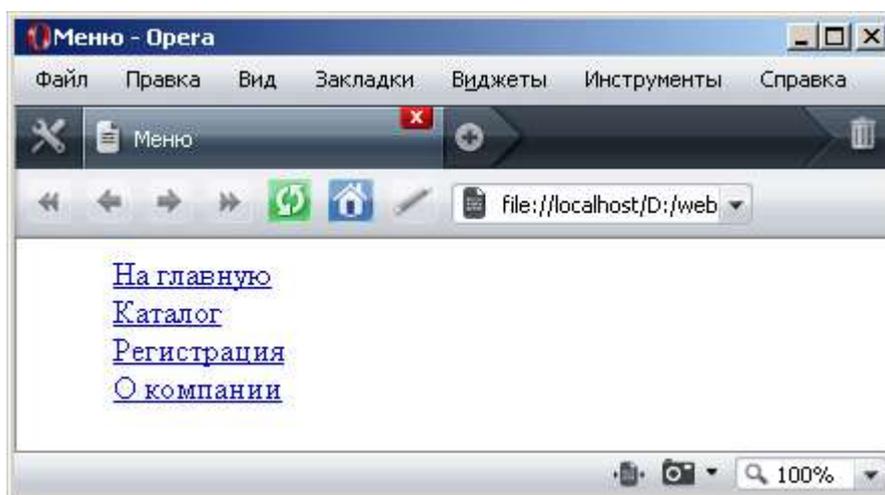


Рисунок 10

Добавляем к стилю меню цвет фона `background:#516` и так как у нас горизонтальное меню во всю ширину окна (или родительского элемента), то задаем `width:100%` (рисунок, 11,12).

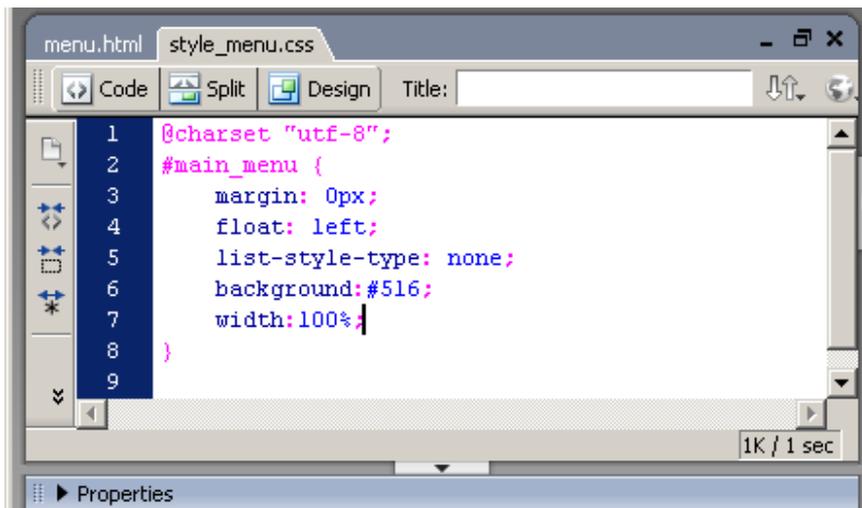


Рисунок 11

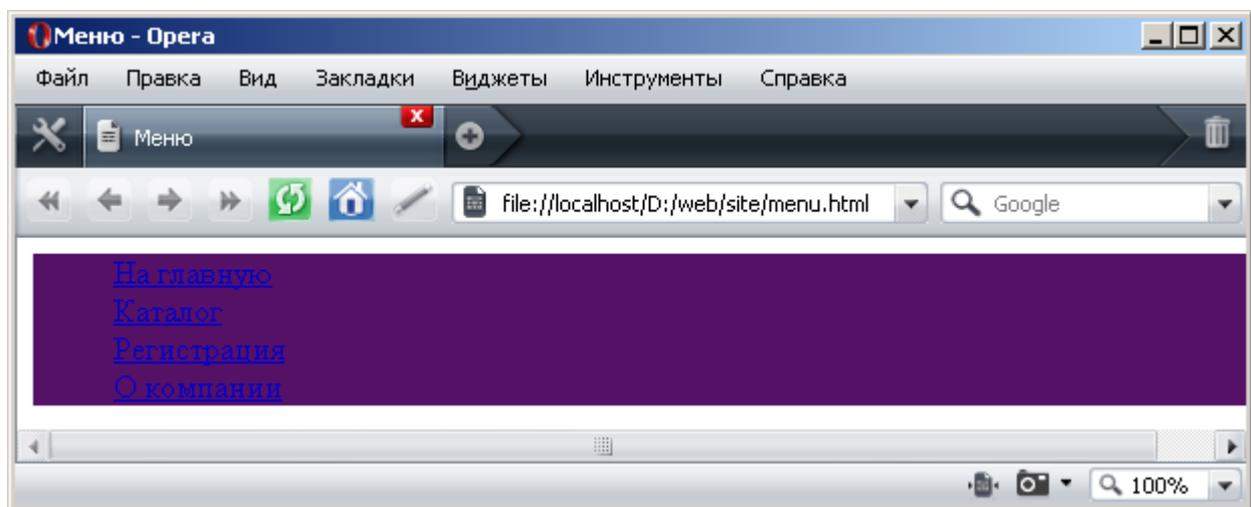
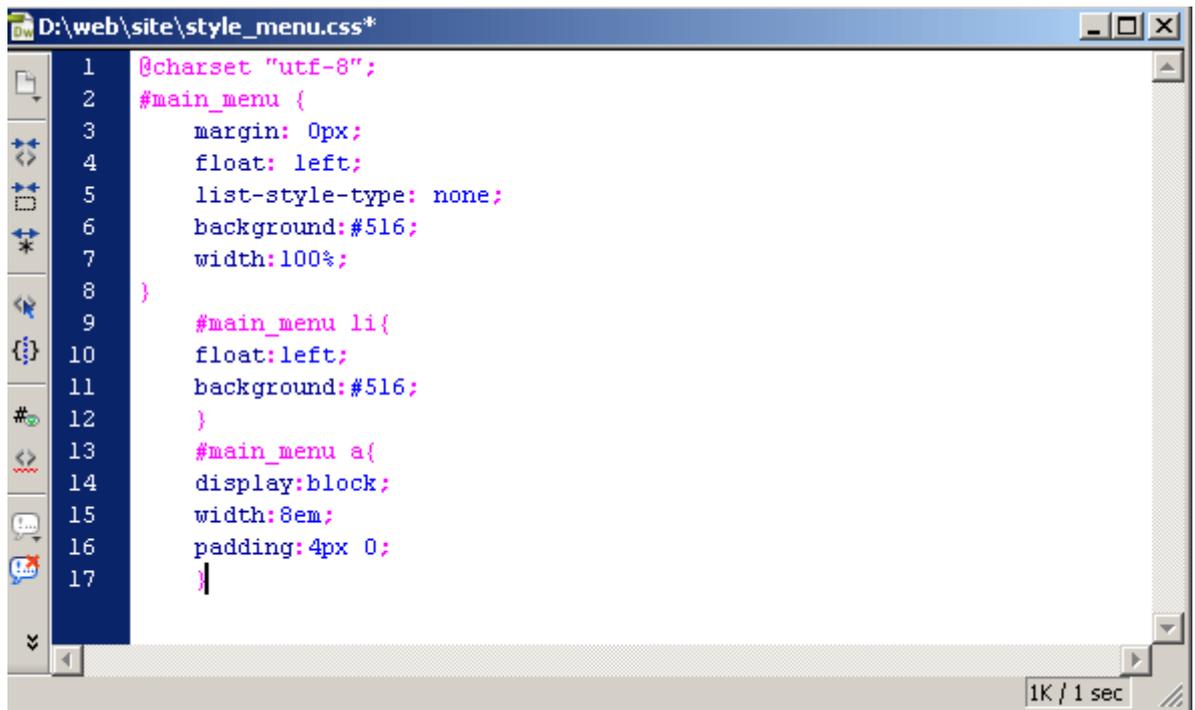


Рисунок 12

Чтобы элементы списка меню располагались на одной линии друг за другом их надо сделать блочными плавающими элементами, для этого добавляем в каскадную таблицу следующие правила см. рисунок 13.



```
D:\web\site\style_menu.css*
1  @charset "utf-8";
2  #main_menu {
3      margin: 0px;
4      float: left;
5      list-style-type: none;
6      background:#516;
7      width:100%;
8  }
9
10 #main_menu li{
11     float:left;
12     background:#516;
13 }
14 #main_menu a{
15     display:block;
16     width:8em;
17     padding:4px 0;
18 }
```

Рисунок 13

Мы сделали элементы li списка меню #main_menu блочными элементами, плавающими слева, а гиперссылки меню сделали блочными элементом с отступами сверху и снизу 4px и шириной в 8em (8 ширин буквы m). Теперь меню выглядит следующим образом см. рисунок 14.

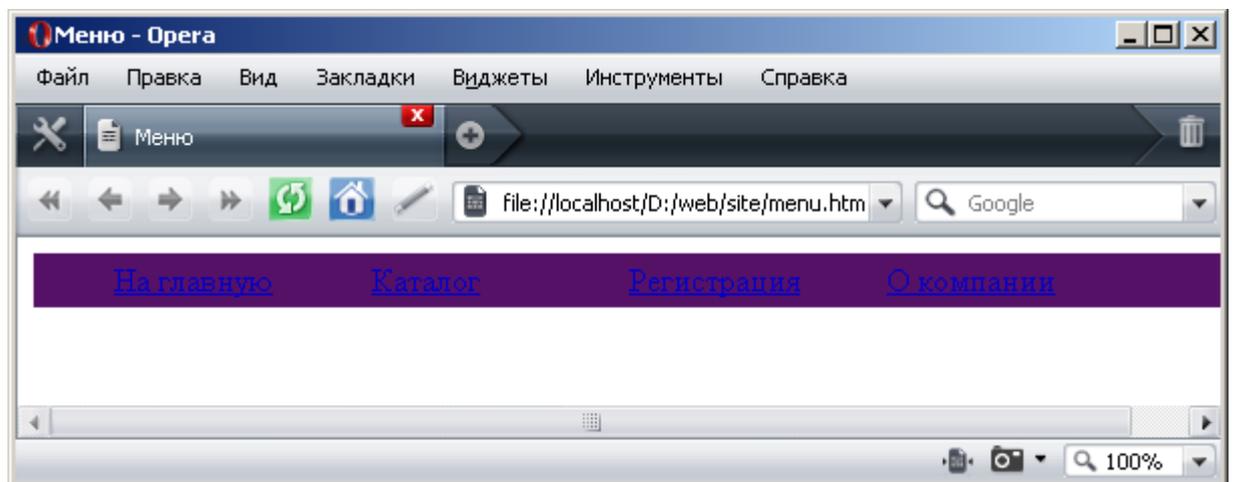
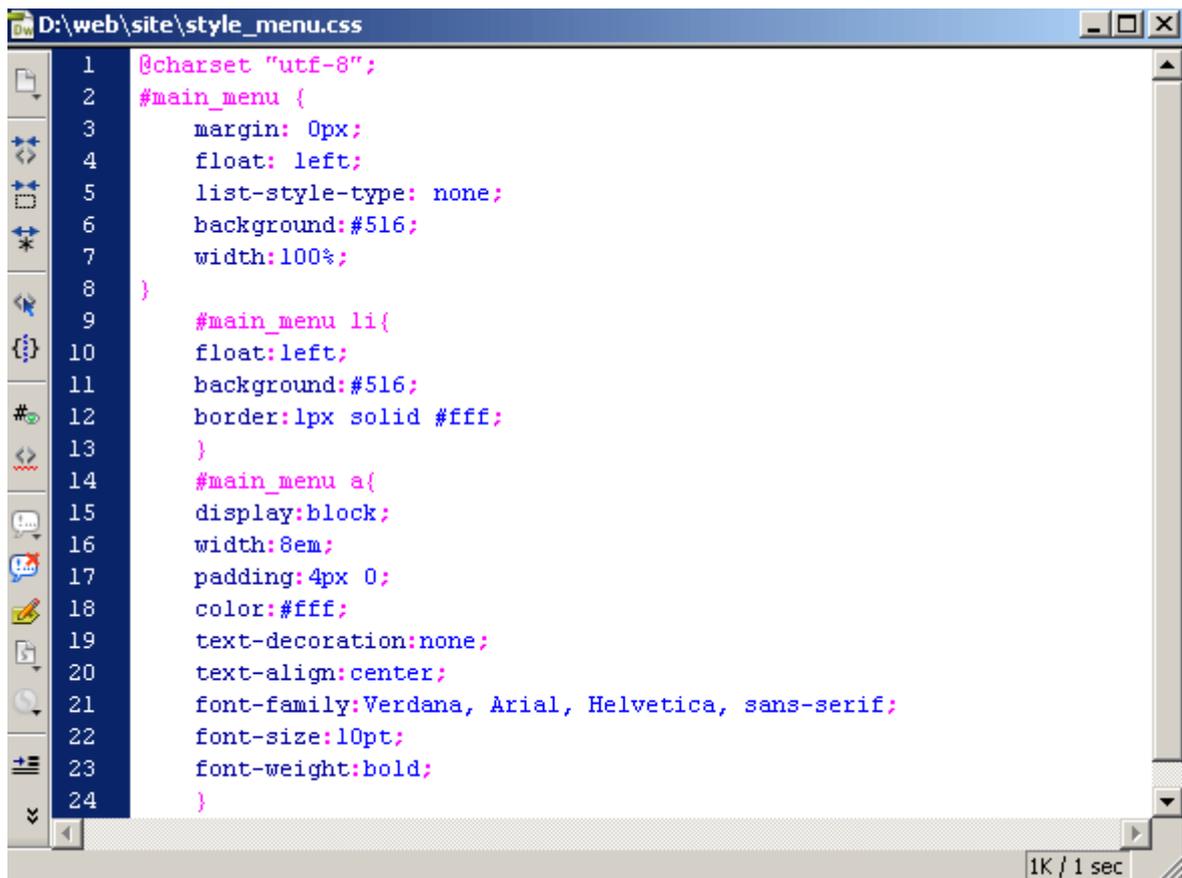


Рисунок 14

Добавим оформление: уберем подчеркивание гиперссылок (text-decoration:none), зададим белый цвет, выравнивание по центру, тип, размер и начертание (полужирный) шрифта (рисунок 15).



```
D:\web\site\style_menu.css
1 @charset "utf-8";
2 #main_menu {
3     margin: 0px;
4     float: left;
5     list-style-type: none;
6     background:#516;
7     width:100%;
8 }
9
10 #main_menu li{
11     float:left;
12     background:#516;
13     border:1px solid #fff;
14 }
15 #main_menu a{
16     display:block;
17     width:8em;
18     padding:4px 0;
19     color:#fff;
20     text-decoration:none;
21     text-align:center;
22     font-family:Verdana, Arial, Helvetica, sans-serif;
23     font-size:10pt;
24     font-weight:bold;
25 }
```

Рисунок 15

Теперь меню выглядит так.

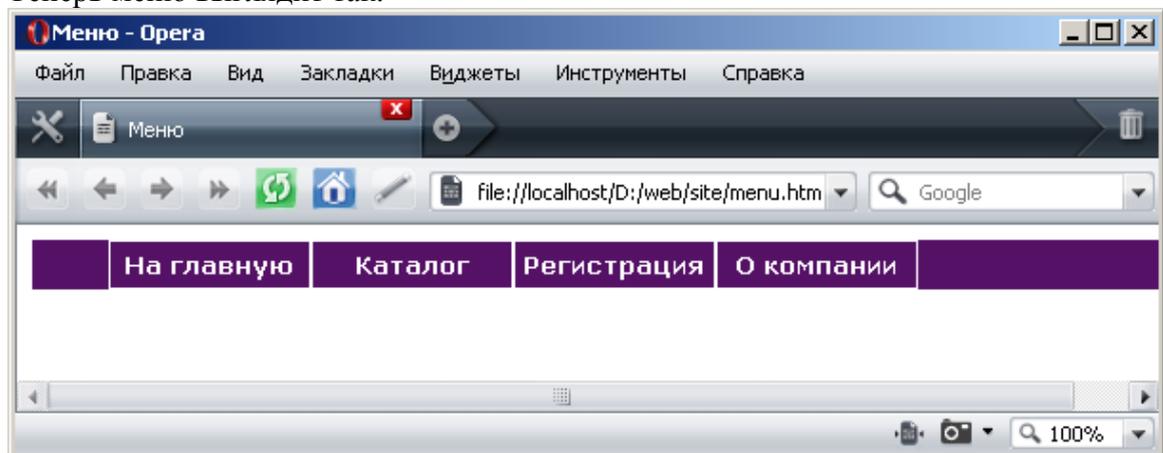
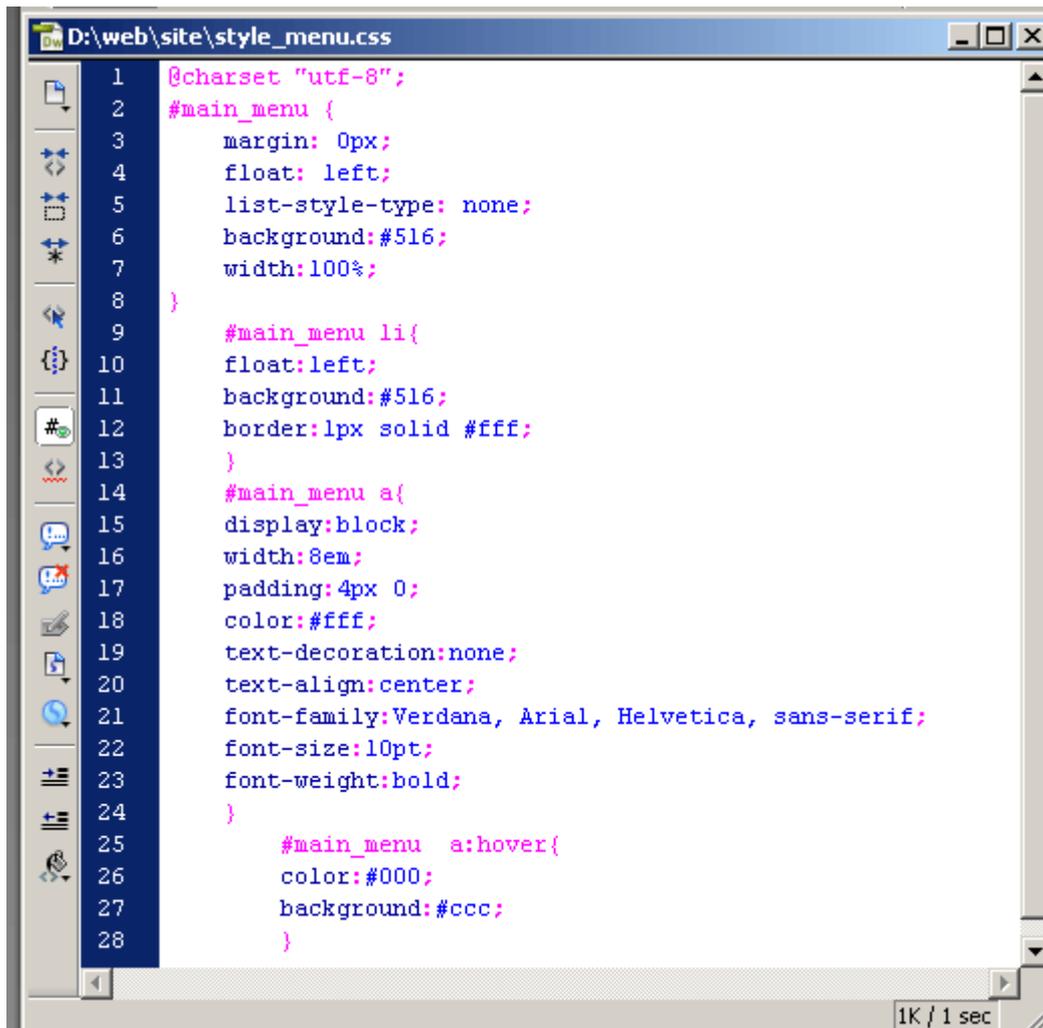


Рисунок 16

Меню должно быть интерактивным и реагировать на указатель (курсор мыши) для этого используем псевдокласс гиперссылок¹ `hover` (объект при наведении на него мыши) см. рисунок 17.

¹ Для стандарта CSS3 этот псевдокласс можно и применять для других объектов html.

A screenshot of a text editor window titled "D:\web\site\style_menu.css". The editor contains CSS code for a menu. The code is as follows:

```
1 @charset "utf-8";
2 #main_menu {
3     margin: 0px;
4     float: left;
5     list-style-type: none;
6     background:#516;
7     width:100%;
8 }
9
10 #main_menu li{
11     float:left;
12     background:#516;
13     border:1px solid #fff;
14 }
15 #main_menu a{
16     display:block;
17     width:8em;
18     padding:4px 0;
19     color:#fff;
20     text-decoration:none;
21     text-align:center;
22     font-family:Verdana, Arial, Helvetica, sans-serif;
23     font-size:10pt;
24     font-weight:bold;
25 }
26 #main_menu a:hover{
27     color:#000;
28     background:#ccc;
29 }
```

The editor has a dark blue sidebar with various icons and a status bar at the bottom right showing "1K / 1 sec".

Рисунок 17

В итоге должно получиться меню как на рисунке 18.

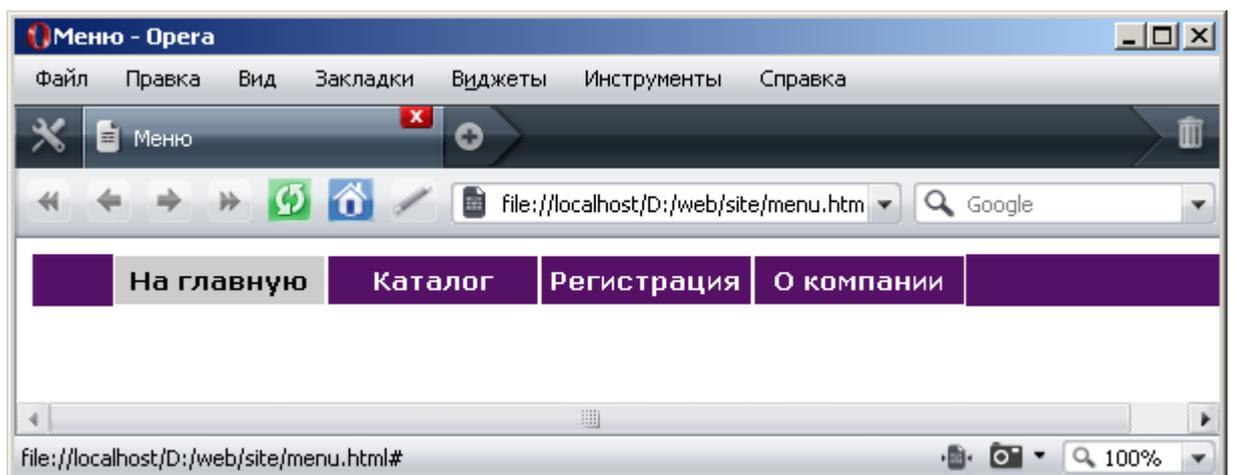


Рисунок 18.

Заготовка меню готова. Сохраните menu.html как: index.html, catalog.html, registration.html, about.html. Чтобы не перезагружалась активная страница, в каждом

файле при совпадении имени файла со ссылкой, поставить ей атрибут href="#" . Пример для страницы index.html:

```
<ul id="main_menu">
  <li><a href="#">На главную</a></li>
  <li><a href="catalog.html">Каталог</a></li>
  <li><a href="registration.html">Регистрация</a></li>
  <li><a href="about.html">О компании</a></li>
</ul>
```

Пример для страницы catalog.html:

```
<ul id="main_menu">
  <li><a href="index.html">На главную</a></li>
  <li><a href="#">Каталог</a></li>
  <li><a href="registration.html">Регистрация</a></li>
  <li><a href="about.html">О компании</a></li>
</ul>
```

Чтобы выделить активную ссылку, например для станицы index.html это ссылка «На главную», которой задан атрибут href="#" , и убрать интерактивность (реакцию на наведение курсора мыши) допишем в css следующие правила:

```
#main_menu a[href="#"], #main_menu a:hover[href="#"] {
  text-decoration:underline;
  background:#aaa;
  color:#000;
}
```

Теперь меню выглядит следующим образом (см. рисунок 19).

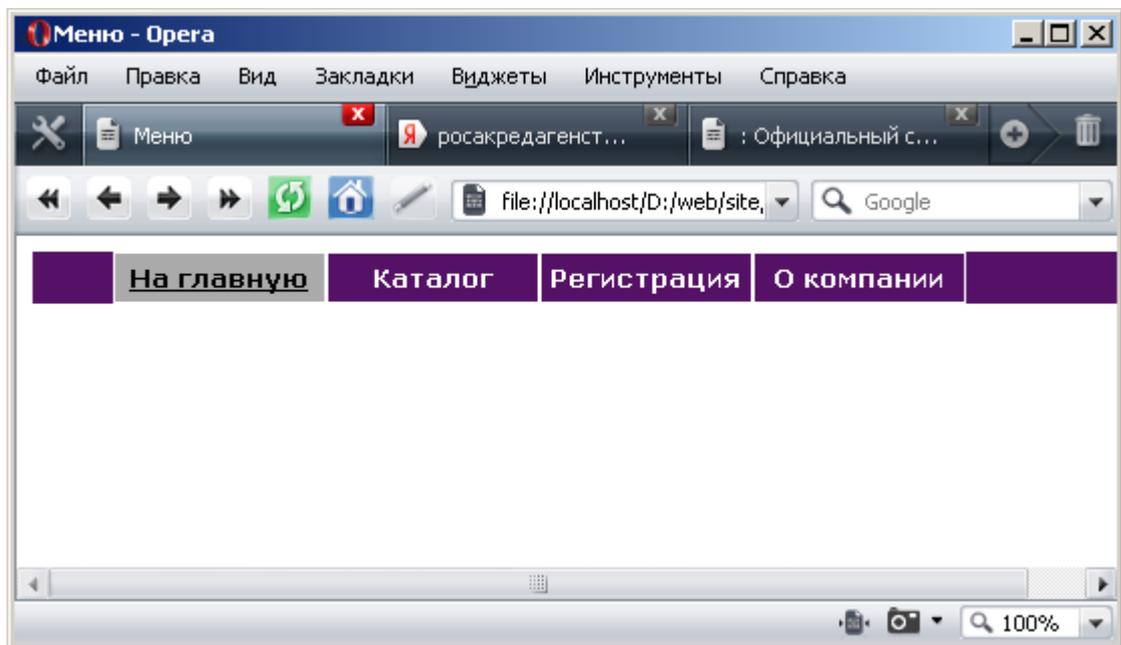
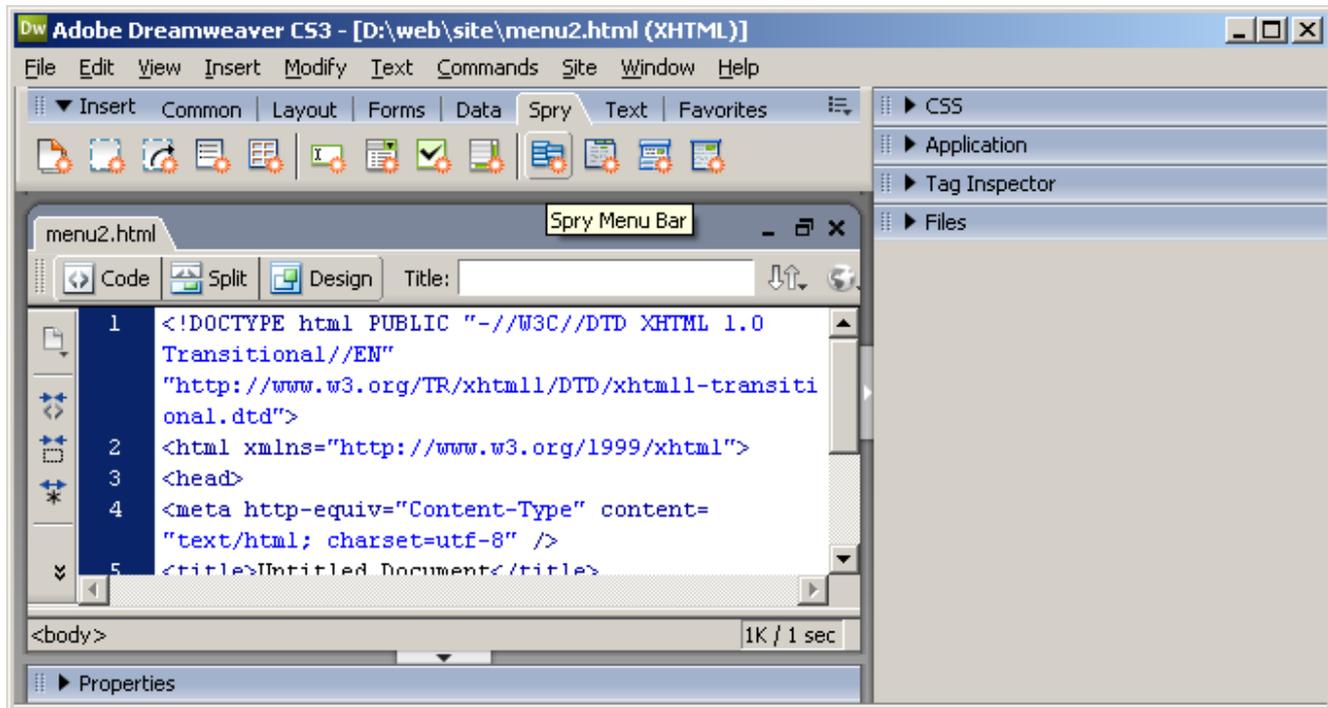


Рисунок 19

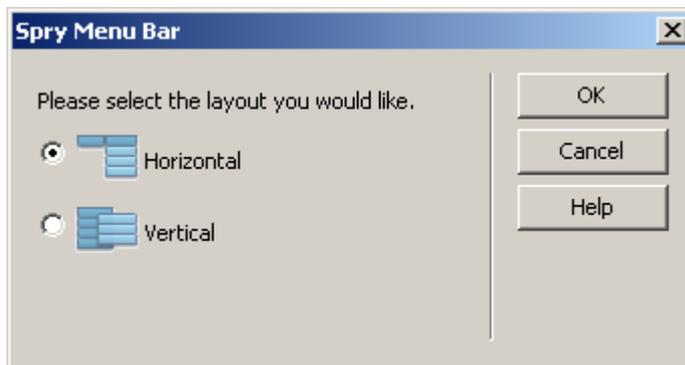
Поменяйте title в каждом файле самостоятельно.

2 Создание меню средствами MS SharePoint Designer

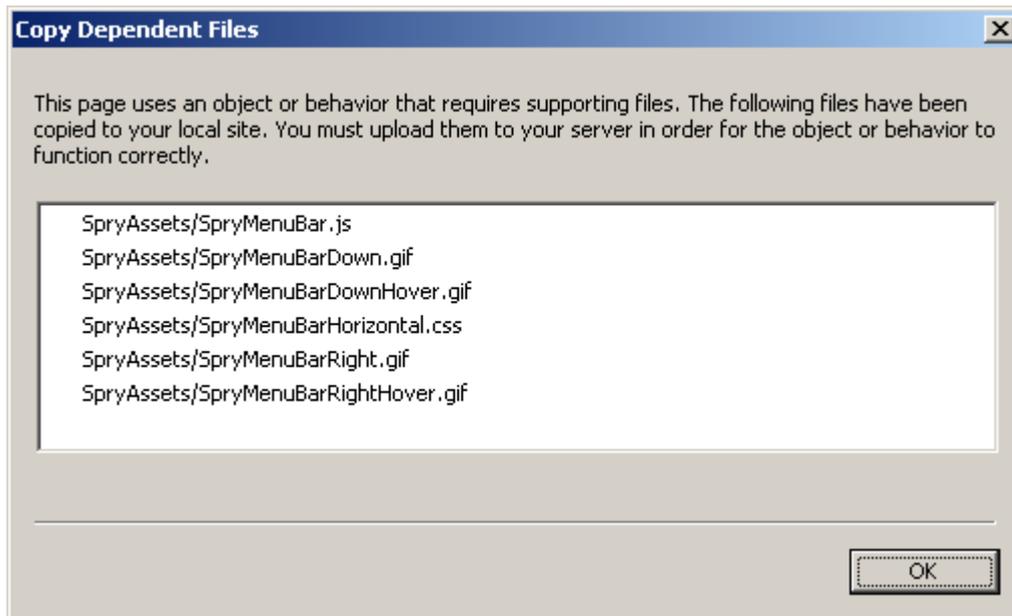
Создайте документ html-документ, сохраните как menu2.html. На панели меню вкладки Spry выберите команду Spry Menu Bar.



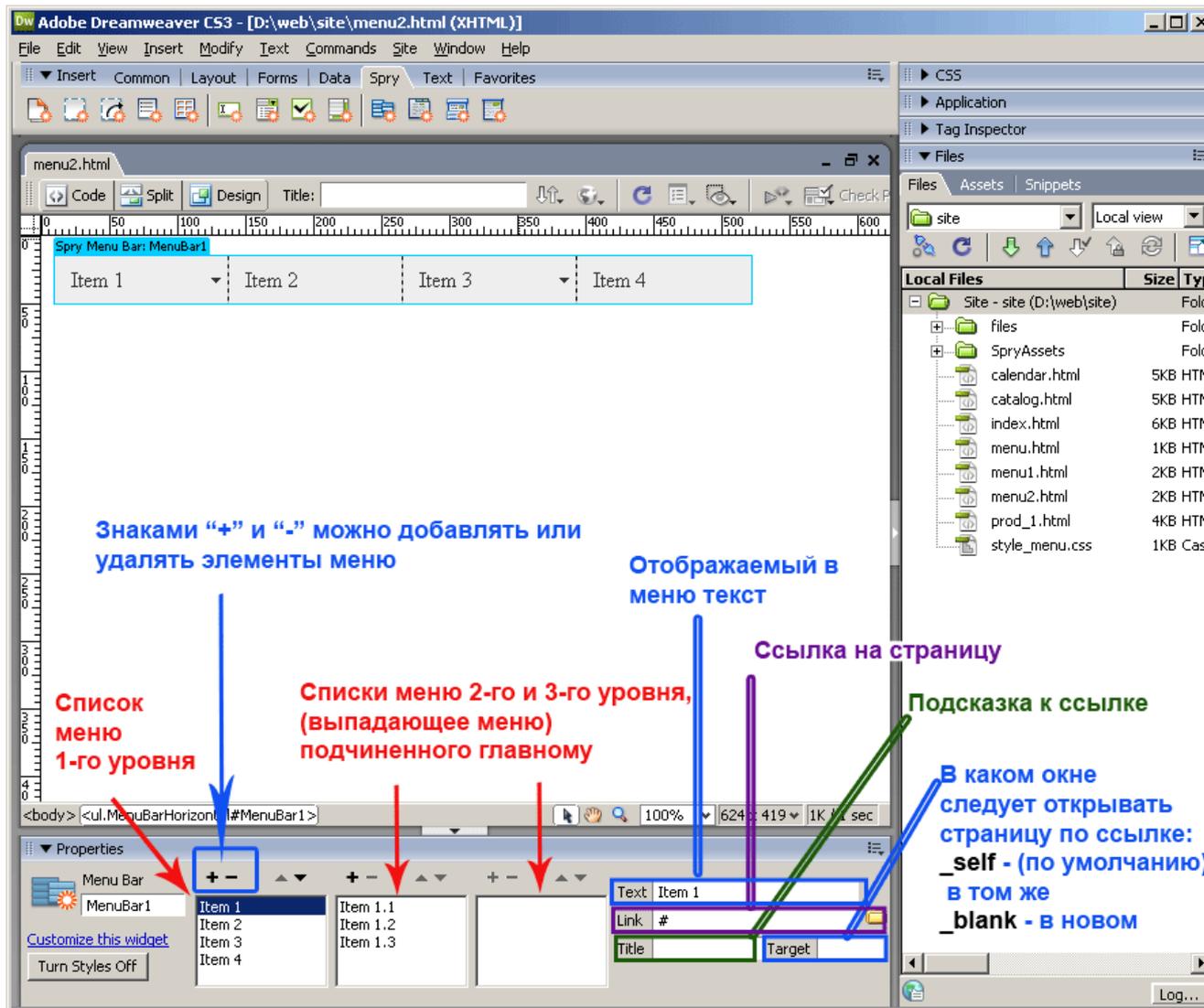
В появившемся диалоговом окне выберите тип меню Horizontal см. рисунок.



Сохраните документ. При сохранении программа предупредит о сохранении в локальную версию в папку SpryAssets файлов скриптов, картинок и css связанных с меню см. рисунок.



Полученную заготовку меню удобно редактировать с помощью панели Properties рисунка.



Отредактируйте меню.

Содержание отчета

Отчет по лабораторной работе должен содержать конспект по теоретической части и выполненные в ходе работы файлы.

6 Создание элементов страниц. Базовые разметки страниц

Цель работы – ознакомление с принципами создания разметки страниц.

Задачи работы – получение навыков создания базовых разметок страниц.

Теоретические сведения

Как правило все разметки страниц можно разделить на три категории:

Страницы с фиксированной шириной. Такие страницы имеют ограничение на ширину для своих элементов и вне зависимости от того какой размер окна браузера – ширина используемого региона фиксирована и не изменяется в процессе работы со страницей. Подобные страницы используются в тех местах где установлены жесткие требования к отображению элементов страницы и неконтролируемое расползание каких то элементов попросту недопустимо. В подобных случаях как правило устанавливают фиксированную ширину тега body и центрируют страницу по ширине окна браузера. Самая распространенное значение ширины для страниц с фиксированной шириной: 960px – такой размер позволяет хорошо выглядеть странице при разрешении экрана начиная с 1024x768.

Страницы с плавающей шириной. – У данного типа страниц ширина его элементов не задана, и они свободно изменяют свои размеры, растекаясь по всему экрану или родительской секции и занимая все свободное место. Такой подход наиболее целесообразно применять в случае если основной целью страницы является отображение текста. В других же случаях применение подобных подходов может сделать из вас дон-Кихота, бессмысленно сражающегося с ветрянной мельницей, коей в вашем случае будет браузер.

«Эластичные страницы». Данные страницы совмещают в себе оба подхода. Элементы таких страниц имеют фиксированную ширину, но способны растягиваться или сжиматься в зависимости от ширины окна браузера в предустановленных диапазонах. Для таких целей можно использовать свойства min-height, min-width, max-height и max-width, о которых я упоминал в первой части.

Для того что бы добиться наилучших результатов, часто имеет смысл объединить все три подхода к разметке страницы: скажем все вспомогательные и навигационные секции – в которых недопустимо неконтролируемое растекание элементов - сделать фиксированными по ширине, а центральную панель сделать «эластичной» или вовсе позволить ей занимать все пространство, остающееся после фиксированных секций в окне браузера.

Способы разметки страницы

CSS позволяет эффективно реализовать разметку, используя различные приемы и техники. К наиболее распространенным приемам относятся плавающие слои (в

англоязычных источниках, данный прием может называться float layout) и абсолютное позиционирование (absolute positioning). Причем, субъективно, разметка абсолютного большинства сайтов использует именно плавающие слои. Далее я постараюсь рассмотреть каждый из этих приемов более детально.

Плавающие слои

Плавающие слои реализуются посредством манипуляций со свойством float для элементов страницы. С помощью этого свойства слои, параграфы и прочие элементы можно позиционировать по левой или правой стороне страницы или внешнего контейнера. Для этого необходимо задать свойство float с соответствующим значением: float: left, right, none;. Остальное содержимое страницы будет «окружать» данный элемент прижатый к одной из сторон. Очень важный момент, который необходимо учесть, что содержимое будет окружать элемент с float только когда оно определено ниже этого тега в коде страницы и его ширина не больше оставшейся ширины страницы или внешнего контейнера. Поэтому в данном случае очень важно определить последовательность описания элементов страницы в HTML файле.

Иногда бывают ситуации, когда необходимо сделать так что бы какое то содержимое не окружало float-элемент а отрисовывалось ниже этого элемента. Примером такой ситуации может быть панель-footer, которая несмотря ни на что всегда должна быть в самом низу страницы. Здесь на помощь может прийти свойство clear которое принимает следующие значения: clear: left; right; both; none. Данное свойство принуждает содержимое отрисовываться ниже float-элемента. Причем с помощью значений этого свойства можно задать к каким float-элементам это относится: с левым выравниванием, с правым и с тем и с тем или ни к тому ни к другому. Так, если у вас есть элемент с float:left, для того что бы остальное содержимое отобразить ниже и не позволить ему окружить данный элемент следует также использовать clear:left.

Однако описанной функциональности разумеется недостаточно для эффективной реализации разметки страницы. В случае плавающих слоев положение элемента страницы по горизонтали как правило задают с помощью CSS свойства margin, которое, как мы уже знаем, используется для задания расстояния от границы одного элемента до другого элемента. На первый взгляд это может показаться несколько странным и не практичным. Однако, это только на первый взгляд. Уникальной особенностью данного свойства является то, что для него можно задавать отрицательные значения, смещая таким образом элемент влево относительно его исходного положения. Ну а соответственно положительно значение смещает элемент вправо (на самом деле разумеется элемент остается на своем месте, просто добавление свойства margin для элемента визуально смещает его на заданное значение).

Таким образом, можно для элементов задать фиксированную ширину и высоту с помощью width и height соответственно, указать требуемые значения float и margin.

Оборудование, программное обеспечение

Оборудование – ПК с установленной графической операционной системой (Windows или Linux).

Программное обеспечение – текстовый редактор, например, Блокнот, MS SharePoint Designer ; графический Web-браузер, например: Internet Explorer, Mozilla Firefox, Safari, Google Chrome или Opera.

Задание на работу

Создать следующие шаблоны разметки страниц:

страницу с фиксированной шириной;

страницу с плавающей верхушкой и подвалом;

страницу с плавающей шириной «резиновая» с фиксированными колонками;

страницу с плавающей шириной «резиновая» с фиксированными колонками и с заданной минимальной шириной.

Ход работы

1 Страница с фиксированной шириной

Код HTML - страницы с фиксированной шириной:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
```

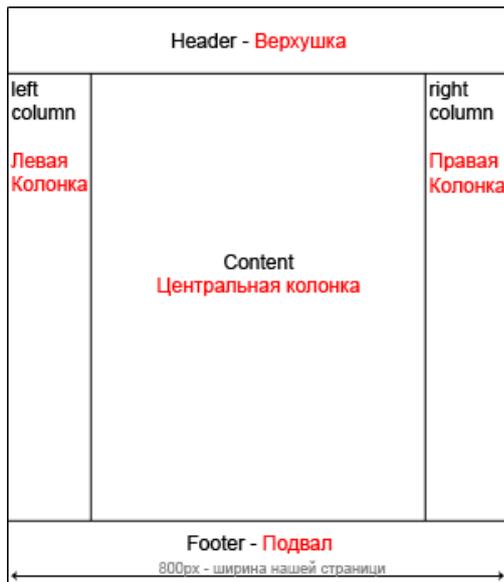
```
<link media="all" href="css/all.css" rel="stylesheet"
type="text/css" />
```

```
<title>структура страницы с фиксированной шириной</title>
</head>
<body>
<div id="main">
  <div class="header">
    Верхушка
  </div>
  <div class="leftcolumn">
    Левая колонка
  </div>
  <div class="content">
    Центральная колонка
  </div>
  <div class="rightcolumn">
    Правая колонка
  </div>
  <div class="footer">
    Подвал
  </div>
</div>
</body>
</html>
```

Код CSS - файла all.css страницы с фиксированной шириной:

```
#main {
margin: 0 auto;
width: 800px;
```

```
}  
  
.header {  
width: 800px;  
height: 50px;  
background: #36f;  
}  
  
.footer {  
width: 800px;  
height: 50px;  
clear: both;  
background: #006;  
}  
  
.leftcolumn {  
width: 150px;  
float: left;  
}  
  
.content {  
width: 500px;  
float: left;  
background: #ccc;  
}  
  
.rightcolumn {  
width: 150px;  
float: left;  
}  
}
```



2 Страница с плавающей верхушкой и подвалом

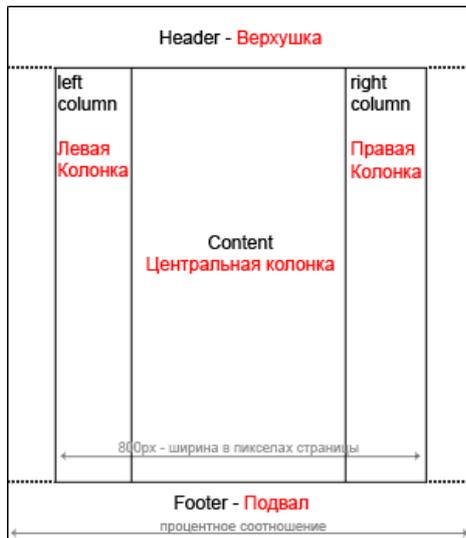
Код HTML - страницы с плавающей верхушкой и подвалом (приведен только код в body, все остальное тоже что и в первом примере):

```
<div id="header">
Верхушка
</div>
<div id="main">
  <div class="leftcolumn">
    Левая колонка
  </div>
  <div class="content">
    Центральная колонка
  </div>
  <div class="rightcolumn">
    Правая колонка
  </div>
</div>
```

```
<div id="footer">
```

Подвал

```
</div>
```



Код CSS - файла all.css с плавающей верхушкой и подвалом:

```
#main {  
margin: 0 auto;  
width: 800px;  
}  
  
#header {  
width: 100%;  
height: 50px;  
background: #36f;  
}  
  
#footer {  
width: 100%;  
height: 50px;  
clear: both;
```

```
background: #006;
}
.leftcolumn {
width: 150px;
float: left;
}
.content {
width: 500px;
float: left;
background: #ccc;
}
.rightcolumn {
width: 150px;
float: left;
}
```

3 Структура страницы с плавающей шириной «резиновая» с фиксированными колонками

Код HTML - страницы с плавающей шириной «резиновая» с фиксированными колонками:

```
<div id="main">
  <div class="header">
    Верхушка
  </div>
  <div id="content-columns">
    <div class="leftcolumn">
```

```

        Левая колонка
    </div>

    <div class="content">
        Центральная колонка
    </div>

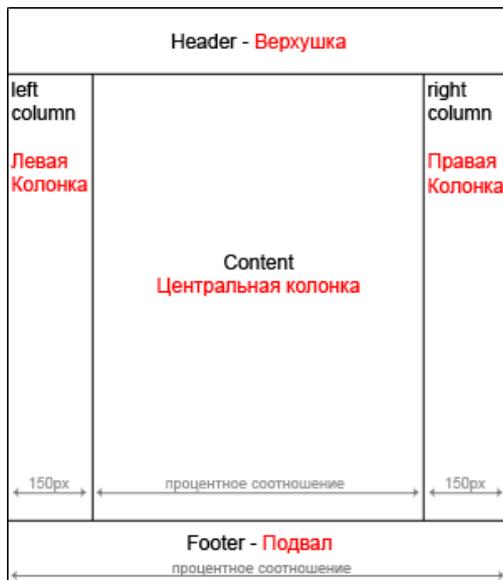
    <div class="rightcolumn">
        Правая колонка
    </div>

</div>

<div class="footer">
    Подвал
</div>

</div>

```



Код CSS - файла all.css с плавающей шириной «резиновая» с фиксированными колонками:

```

#main {
width: 100%;

}

```

```
.header {
width: 100%;
height: 50px;
background: #36f;
}

.footer {
width: 100%;
height: 50px;
clear: both;
background: #006;
}

#content-columns {
margin: 0 150px 0 150px;
}

.content {
width: 100%;
float: left;
background: #ccc;
}

.leftcolumn {
width: 150px;
float: left;
margin-left: -150px;
display: inline; /* решает проблему в IE */
position: relative;
}

.rightcolumn {
```

```
width: 150px;
float: right;
margin-right: -150px;
display: inline; /* решает проблему в IE */
position: relative;
}
```

4 Структура страницы с плавающей шириной «резиновая» с фиксированными колонками и с заданной минимальной шириной

«Резиновая» страница, но она имеет минимальную ширину при сворачивании окна браузера. В примере 1.3 если окно браузера свернуть то колонки слева и справа могут почти полностью закрыть центральную колонку, что частенько не приемлемо. А в данном случае указав минимальную ширину, допустим, в 800px, а боковые колонки по 150px, добьемся того, что браузер 500px для нашей центральной колонки оставит в любом случае.

Код HTML - страницы с плавающей шириной «резиновая» с фиксированными колонками и с заданной минимальной шириной:

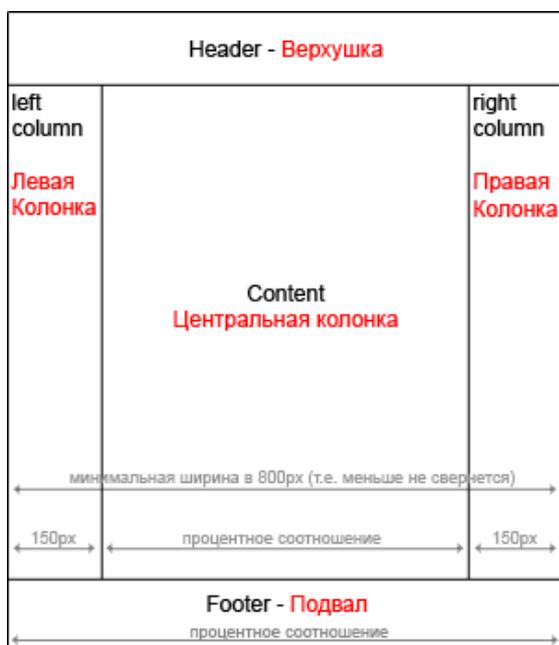
```
<div id="main">
  <div class="header">
    Верхушка
  </div>
  <div id="content-columns">
    <div class="leftcolumn">
      Левая колонка
    </div>
    <div class="content">
      Центральная колонка
    </div>
  </div>
</div>
```

```

        <div class="rightcolumn">
            Правая колонка
        </div>
    </div>

    <div class="footer">
        Подвал
    </div>
</div>

```



CSS - Структура страницы с плавающей шириной «резиновая» с фиксированными колонками и с заданной минимальной шириной:

```

body {
margin: 0;
padding: 0;
min-width: 800px;
}

/* Следующая запись нужна для IE */
* html body {

```

```
width:expression(document.documentElement.clientWidth < 800 ?
"800px" : "auto");

}

#main {
width: 100%;
}

.header {
width: 100%;
height: 50px;
background: #36f;
}

.footer {
width: 100%;
height: 50px;
clear: both;
background: #006;
}

#content-columns {
margin: 0 150px 0 150px;
}

.content {
width: 100%;
float: left;
background: #ccc;
}

.leftcolumn {
width: 150px;
float: left;
```

```
margin-left: -150px;
display: inline; /* решает проблему в IE */
position: relative;
}
.rightcolumn {
width: 150px;
float: right;
margin-right: -150px;
display: inline; /* решает проблему в IE */
position: relative;
}
```

Содержание отчета

Отчет по лабораторной работе должен содержать конспект по теоретической части и выполненные в ходе работы файлы.

7 Синтаксис и грамматика JavaScript

Цель работы – ознакомление с принципами создания сценариев на стороне клиента.

Задачи работы – получение навыков создания сценариев на стороне клиента средствами JavaScript.

Теоретические сведения

Включение JavaScript сценариев в (x)HTML

1. Включение JavaScript сценариев в документ возможно в окружении тега `<script>`:

```
<script>document.write("Вася");</script>
```

Тэг `<script>` можно использовать внутри тега `<head>` или `<body>`. В первом случае он выполнится до отображения страницы в браузере, во втором случае по мере выполнения браузером тегов указанных в `<body>`.

Пример

[Файл 1_0.html](#)

Adobe Dreamweaver CS3 - [D:\WebCtr\javaScript\jsпрактика1_1\1_0.html (XHTML)]

File Edit View Insert Modify Text Commands Site Window Help

1_0.html

Code Split Design Title: 1_0

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4 <head>
5 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6 <title>1_0</title>
7 <script>document.write("Баца");</script>
8 </head>
9 <body>
10 </body>
11 </html>
12
```

Файл 1_1.html

Adobe Dreamweaver CS3 - [D:\WebCtr\javaScript\jsпрактика1_1\1_1.html (XHTML)]

File Edit View Insert Modify Text Commands Site Window Help

1_1.html

Code Split Design Title: 1_1

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4 <head>
5 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6 <title>1_1</title>
7 </head>
8 <body>
9 <script>document.write("Баца");</script>
10 </body>
11 </html>
12 |
```

Проверьте работу страниц [1_0.html](#) и [1_1.html](#) в браузере. В данных примерах с помощью метода `write()` мы в теле документа `document`, пишем строку «Вася». Обратите внимание, что объект и метод разделен точкой.

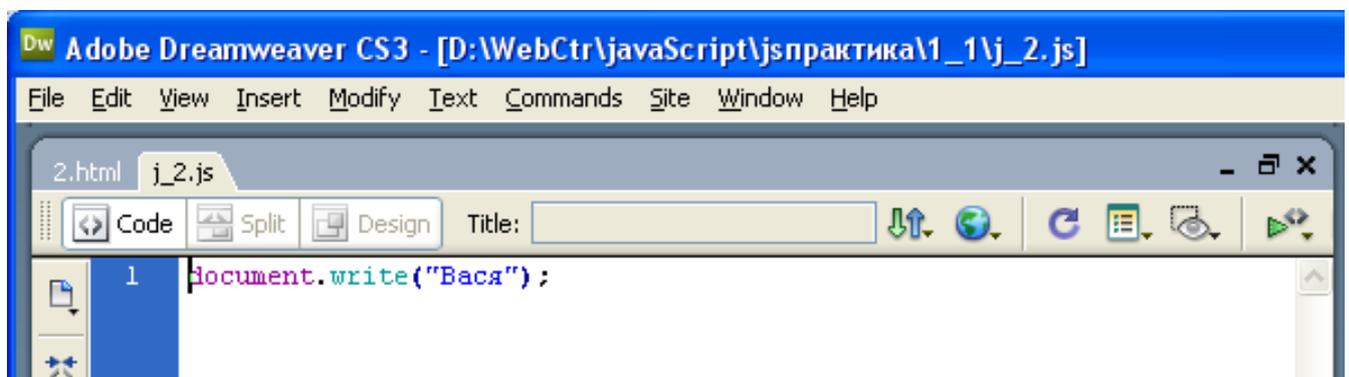
2. В виде ссылки на внешний файл атрибут `src` тэга `<script>`:

```
<script type="text/javascript" src="j_2.js">
```

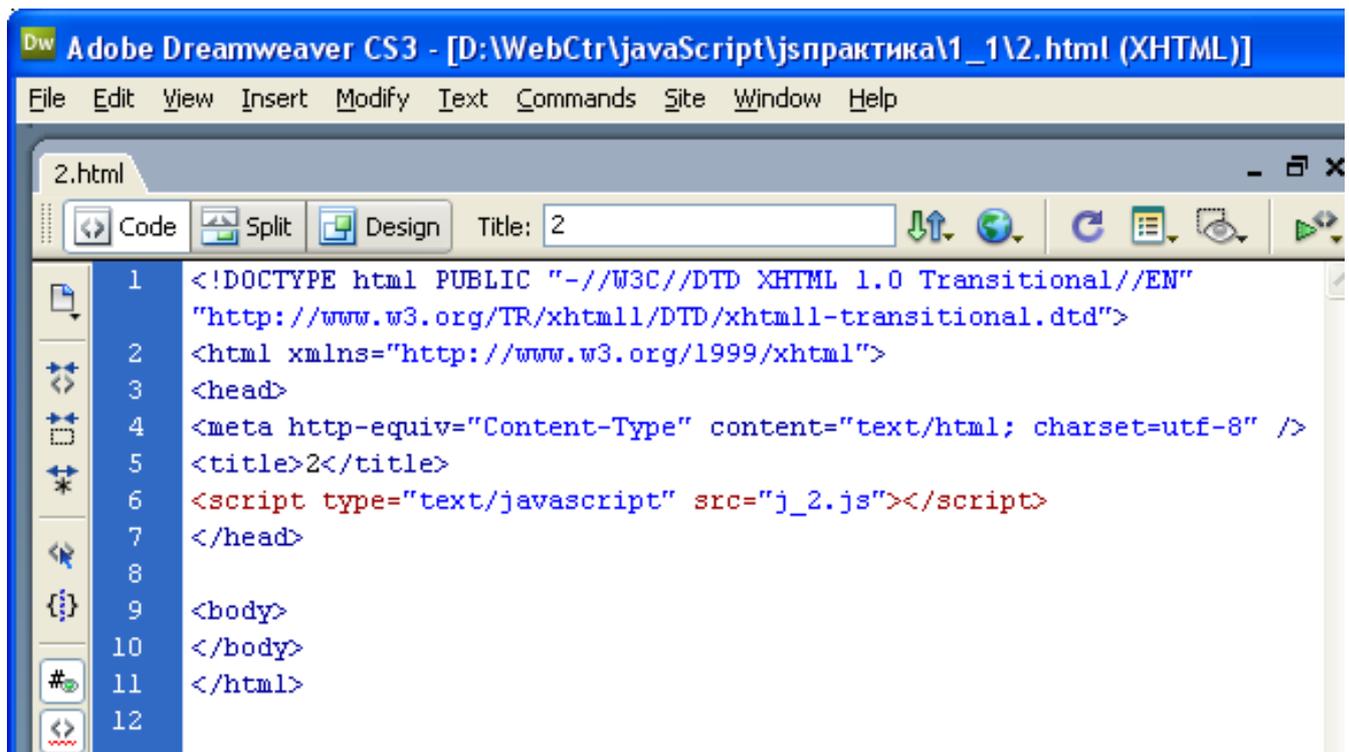
Таким способом удобно разграничивать текст скрипта и содержание документа.

Пример

Создайте сначала файл `j_2.js`



2.html



3. При обработке событий (x)HTML:

```
<body onLoad="document.write('Вася');">
```

Пример

Файл 3.html



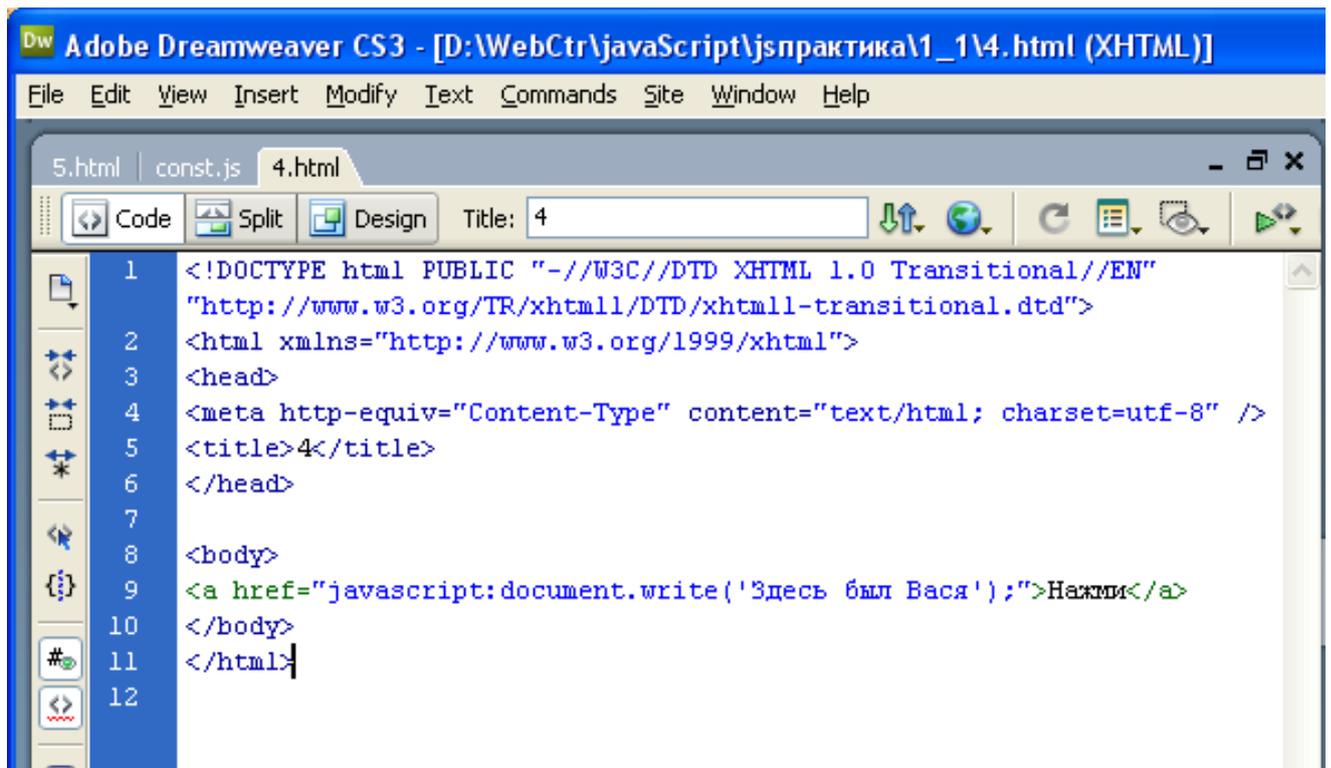
onLoad событие, возникающее при загрузке документа в браузер.

4. Заданием псевдо URL-ссылки:

```
<a href="javascript:document.write('Здесь был Вася');">Нажми</a>
```

Пример

Файл 4.html



Раньше код сценария должен был быть заключен в тэги `<!-- -->` комментария HTML для того, чтобы старые браузеры, не понимающие JavaScript, не отображали код на экране.

При помещении сценария JavaScript в заголовке `<head>` документа весь сценарий будет загружен до того, как потребуется его выполнить.

Разделение инструкций

Инструкции разделяются точкой с запятой - `;`. Конец строки воспринимается как конец инструкции.

```
var a=25;
var b=7;
s=a%b;
document.write("s");
```

```
var a=6
var b=7
s=a*b
document.write("s")
```

Обозначение переменных

Имена переменных должны начинаться с буквы или символа подчеркивания, второй и последующие символы имени переменной могут быть буквой, цифрой и подчеркиванием. Регистр учитывается. Переменная считается объявленной с момента ее упоминания в программе, специальных объявлений делать не нужно. Для объявления можно использовать ключевое слово `var`. Например: `var x=15`

Способы указания комментариев

```
// Для одной строки
/* Комментарий из нескольких
строк */
```

ТИПЫ ДАННЫХ

string (строка) – последовательность символов, заключенных в кавычки;

numbers (число) – целые числа и десятичные дроби;

boolean (логическое) – True или False;

null – отсутствие данных.

Специальные символы (для PHP тоже)

\n Новая строка
\t Табуляция
\f Новая страница
\b Забой
\r Возврат каретки

Для вывода спец.символов внутри строки используется обратный слеш (\). Чаще всего это используется для кавычек, косой черты или для включения восьмеричного значения символа.

Пример

```
document.write("\"Кавычки\" внутри строки");
document.write("Обратная косая черта \\");
document.write("Символ пробела: \040.");
```

Переменные

Тип переменных не нужно задавать при их объявлении (для PHP тоже). Каждая переменная может содержать значения различных типов. Преобразование типов переменных делается автоматически.

Пример 1

```
x=55; //x - числовая переменная, имеющая значение 55
y="55"; // y - строковая переменная, имеющая значение "55"
```

`z=1+y;` //если `y` - строковая переменная, то она будет автоматически преобразована в ближайшее целое значение так, чтобы 1 могла быть прибавлена к ней.

`document.write(x)` // выводится на экран 55. Если `x` - целое число, а не строковая переменная, то Javascript выполнит необходимые преобразования.

`n=3.1459;` // назначение действительного (дробного)

`n=0785;` // числа, начинающиеся с 0, - восьмеричные `n`

`n=0xFA8BCA;` // числа, начинающиеся с 0x, - шестнадцатеричные

`4.16E-7` // экспоненциальное представление десятичного числа (с плавающей точкой)

Пример 2

`document.write(2*3);` //выведет 6

`document.write("10"-6);` //выведет 4

`document.write("6"*3);` //выведет 18

`document.write("10"+6);` //выведет 106

ОПЕРАТОРЫ

Таблица – Арифметические операторы

Оператор	Операция	Пример
-	Смена знака числа	<code>x = - x;</code>
+	Сложение	<code>c = x + y;</code>
-	Вычитание	<code>c = x - y;</code>
*	Умножение	<code>c = x * y;</code>
/	Деление	<code>c = x / y;</code>
%	Остаток от деления	<code>c = x % y;</code>
++	Инкремент (увеличение на единицу)	<code>x = ++x;</code> <code>x = x++;</code>
--	Декремент (уменьшение на единицу)	<code>x = --x;</code> <code>x = x--;</code>

При префиксной операции `++x` сначала выполняется арифметическая операция, а затем приравнивание полученного результата. При постфиксной `x++` наоборот. Пример.

Операция постфиксная

`x=10`
`x++=10`
`x=11`
`x--=11`
`x=10`

Операция префиксная

`y=10`
`++y=11`
`y=11`
`--y=10`
`y=10`

Таблица – Операторы присваивания

Оператор	Эквивалентное выражение
=	Простое присваивание
x += y	x = x + y;
x -= y	x = x - y;
x *= y	x = x * y;
x /= y	x = x / y;
x %= y	x = x % y;

Таблица – Операторы сравнения

Оператор	Описание
<	Меньше
>	Больше
==	Равно (с преобразованием типа данных)
<=	Меньше или равно
>=	Больше или равно
!= или <>	Не равно(с преобразованием типа данных)
===	Строго равно (без преобразования типа данных)
!==	Строго не равно (без преобразования типа данных)

Таблица – Логические операторы

Пример	Название	Результат
a && b a and b	Конъюнкция	Истина, если истинны оба операнда
a & b a or b a b	Дизъюнкция	Истина, если истинен хотя бы один из операндов
!a not a	Отрицание Инверсия	Истина, если значение <i>a</i> ложно Ложно, если значение <i>a</i> истина
a xor b	Исключающая дизъюнкция	Истина, если истинен только один из операндов

Таблица – Строковые

Оператор	Описание
+	Склеивание строк
> >=	Сравнивает строки путем сравнения кодов ASCII, начиная с правого конца
< <=	Сравнивает строки путем сравнения кодов ASCII, начиная с левого конца

Таблица – Набор встроенных функций, доступных при выполнении кода:

Функция	Описание
escape (символ)	Возвращает строку в виде % XX , где XX –код ASCII символа в скобках

eval (выражение)	Возвращает результат вычисления выражения в скобках
isNaN (значение)	Возвращает True, если значение в скобках не является числом
parseFloat (строка)	Преобразует строку в число с плавающей запятой
parseInt (строка, основание)	Преобразует строку в целое число в системе счисления с указанным основанием
typeof (объект)	Возвращает тип указанного объекта как строку, например boolean, function и т. д.

Оборудование, программное обеспечение

Оборудование – ПК с установленной графической операционной системой (Windows или Linux).

Программное обеспечение – текстовый редактор, например, Блокнот, MS SharePoint Designer ; графический Web-браузер, например: Internet Explorer, Mozilla Firefox, Safari, Google Chrome или Opera.

Задание на работу

Создать странички выводящие таблицу умножения 10x10 используя все способы включения JavaScript сценариев

Ход работы

1. Сценарии внутри странички.

1.1. Непосредственно в теле странички в месте использования:

Файл page0.html

```
<html>
<head>
</head>
<body>
<h2 align="center">Таблица умножения</h2>
<script>
```

```
var col=10;

    row=10;

    document.write("<table align=\"center\" style=\"border:
1px solid #000;\">");

    for (i=1; i<=row; i++) {

        document.write("<tr>");

        for (j=1; j<=col; j++) {

            document.write("<td>"+(i*j)+"</td>");

        }

        document.write("</tr>");

    }

    document.write("</table>");

</script>

</body>

</html>
```

1.2. В заголовке между <head> </head>:

Файл page1.html

```
<html>

<head>

<script>

var col=10;

    row=10;

    document.write("<h2 align=\"center\">Таблица
умножения</h2>");

    document.write("<table align=\"center\" style=\"border:
1px solid #000;\">");

    for (i=1; i<=row; i++) {
```

```
        document.write("<tr>");
        for (j=1; j<=col; j++) {
            document.write("<td>"+(i*j)+"</td>");
        }
        document.write("</tr>");
    }
    document.write("</table>");
</script>
</head>
<body>
</body>
</html>
```

1.3. В ответ на событие:

Пример 1

Файл page2_1.html

```
<html>
<head>
</head>
<body>
<p onClick="
var col=10;
    row=10;
    document.write('<h2 align=center>Таблица умножения</h2>');
    document.write('<table align=center border=1>');
    for (i=1; i<=row; i++) {
        document.write('<tr>');
```

```
        for (j=1; j<=col; j++) {
            document.write('<td>'+(i*j)+'</td>');
        }
        document.write('</tr>');
    }
    document.write('</table>');
">Вывести таблицу умножения</p>
</body>
</html>
```

Пример 2

Файл page2_2.html

```
<html>
<head>
</head>
<body onLoad="
var col=10;
    row=10;
    document.write('<h2 align=center>Таблица умножения</h2>');
    document.write('<table align=center border=1>');
    for (i=1; i<=row; i++) {
        document.write('<tr>');
        for (j=1; j<=col; j++) {
            document.write('<td>'+(i*j)+'</td>');
        }
        document.write('</tr>');
    }
```

```
        document.write('</table>');">
</body>
</html>
```

2. Сценарии во внешнем файле. Как правило, тексты с JavaScript сценарием имеют расширение js.

2.1. В месте использования:

Файл `javas_1.js`

```
var col=10;

    row=10;

    document.write("<table align=\"center\" style=\"border:
1px solid #000;\">");

    for (i=1; i<=row; i++) {

        document.write("<tr>");

        for (j=1; j<=col; j++) {

            document.write("<td>"+(i*j)+"</td>");

        }

        document.write("</tr>");

    }

    document.write("</table>");
```

Файл `page3.html`

```
<html>
<head>
</head>
<body>
<h2 align="center">Таблица умножения</h2>
<script type="text/javascript" src="javas_1.js">
```

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

2.2. В заголовке между <head> </head>:

Файл `javas_2.js`

```
var col=10;
    row=10;

    document.write("<h2 align=\"center\">Таблица
умножения</h2>");

    document.write("<table align=\"center\" style=\"border:
1px solid #000;\">");

    for (i=1; i<=row; i++) {
        document.write("<tr>");
        for (j=1; j<=col; j++) {
            document.write("<td>"+(i*j)+"</td>");
        }
        document.write("</tr>");
    }

    document.write("</table>");
```

Файл `page4.html`

```
<html>
<head>
<script type="text/javascript" src="javas_2.js">
</script>
</head>
```

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

3. Как правило, тексты скрипов для удобства использования и отладки оформляют в виде отдельных функций:

Файл `javas.js`

```
function table_x(){
var col=10;
    row=10;
    document.write("<table align=\"center\" style=\"border:
1px solid #000;\">");
    for (i=1; i<=row; i++) {
        document.write("<tr>");
        for (j=1; j<=col; j++) {
            document.write("<td>"+(i*j)+"</td>");
        }
        document.write("</tr>");
    }
    document.write("</table>");
}
```

Файл `page4.html`

```
<html>
<head>
<script type="text/javascript" src="javas.js">
```

```
</script>
</head>
<body>
<h2 align="center">Таблица умножения</h2>
<script>
table_x();
</script>
</body>
</html>
```

Задание1. Переписать функцию, выводящую таблицу, см. рис.

Задание2. Сделать пример включение javascript-сценария как псевдо URL ссылки.

Содержание отчета

Отчет по лабораторной работе должен содержать конспект по теоретической части и выполненные в ходе работы файлы.



The screenshot shows a web browser window titled "Таблица умножения - Opera". The browser's menu bar includes "Файл", "Правка", "Вид", "Закладки", "Виджеты", and "Инст". The main content area displays a multiplication table with the title "Таблица умножения". The table is a 10x10 grid with blue headers and light blue cells. The numbers 1 through 10 are written in the header cells, and the products of the corresponding rows and columns are in the body cells.

	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

8 Синтаксис и грамматика JavaScript. Основные конструкции

Цель работы – ознакомление с принципами создания сценариев на стороне клиента.

Задачи работы – получение навыков создания сценариев на стороне клиента средствами JavaScript.

Теоретические сведения

ОСНОВНЫЕ КОНСТРУКЦИИ

Условные конструкции if

Основная конструкция

```
if (логическое выражение)
  оператор_1;
else
  оператор_2;
```

Конструкция с блоком операторов

```
if (логическое выражение)
{ оператор_1;
  ...
  оператор_i; }
else
{ оператор_i+1;
  ...
  оператор_n; }
```

Сокращенная конструкция

```
if (логическое выражение)
  оператор_1;
```

Полная конструкция

```
if (логическое выражение 1)
  оператор_1;
else if (логическое выражение
2)
  оператор_2;
else if (логическое выражение
3)
  оператор_3;
...
else
  оператор_n;
```

Условная конструкция ?:

(логическое выражение) ? оператор_если_истина : оператор_если_ложь ;

Циклические конструкции

Цикл – синтаксическая конструкция, позволяющая многократно повторять оператор (группу операторов). Прекращение выполнения цикла происходит если:

1. Перестало выполняться условие выполнения цикла.
2. В теле цикла выполнен оператор `break`.
3. В теле цикла выполнен оператор `return`.

Цикл с предусловием (while)

```
while (логическое выражение)
  инструкция;
```

Цикл с потусловием (do- while)

выполняется хотя бы 1 раз

```
do
{
```

```

        тело цикла;
    }
while (логическое выражение);

```

Цикл со счетчиком (for)

Синтаксис

```

for (команды_инициализации; условие; команды_после_итерации)
{
    тело цикла;
}

```

Цикл перебора свойств объекта (for in)

Синтаксис

```

for (переменная in объект)
{
    тело цикла;
}

```

Операторы прерывания цикла (break, continue)

break

Прерывает работу всего цикла

```

var x=0;
while (x<20){
    if (x==16) break; /*цикл
выполнится только до 16 */
    ++x;
    document.write(x+"<br>");
}

```

continue

Прерывает работу текущей итерации

```

var x=0;
while (x<20){
    x++;
    if (x==16) continue; /*цикл
выполнится до 20, только 16
будет пропущено*/
    document.write(x+"<br>");
}

```

Конструкция выбора (switch-case)

Синтаксис

```

switch (выражение)
{
    case значение_1 :
команды_1; [break;]
        ...
    case значение_N : команды_N;
[break;]
    [default: [break;]
}

```

Альтернативный синтаксис

```

switch (выражение):
    case значение_1 : команды_1;
[break;]
        ...
    case значение_N : команды_N;
[break;]
    [default: [break;]

```

Оборудование, программное обеспечение

Оборудование – ПК с установленной графической операционной системой (Windows или Linux).

Программное обеспечение – текстовый редактор, например, Блокнот, MS SharePoint Designer ; графический Web-браузер, например: Internet Explorer, Mozilla Firefox, Safari, Google Chrome или Opera.

Задание на работу

Создать страничку с использованием всех основных конструкций JavaScript сценариев

Ход работы

1. Создайте страничку pagejs.html:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
<title>5</title>
<script src="const.js"></script>
</head>

<body>

<h1>Функция do_while() с циклом do while</h1>
<script>
do_while();
</script>

<h1>Функция for_p() с циклом for</h1>
<script>
for_p();
</script>

<h1>Функция break_p() с циклом while и инструкцией
прерывания break</h1>
<script>
break_p()
</script>

<h1>Функция continue_p() с циклом while и инструкцией
прерывания continue</h1>
<script>
continue_p()
</script>
```

```
<h1>Функция switch_p() с конструкцией выбора (switch-  
case)</h1>  
<script>  
switch_p()  
</script>  
  
<h1>Функция for_in() с циклом for in</h1>  
<script>  
for_in();  
</script>  
  
</body>  
</html>
```

2. Создайте файл const.js:

```
// JavaScript Document  
  
function do_while(){  
    var x=12;  
    do {  
        document.write((x--)+"<br>");  
    }  
    while (x!=0);  
}  
  
function for_p(){  
    for(var i=0;i<12;i++) {  
        document.write(i+"<br>");  
    }  
}  
  
function for_in(){  
    for(i in document)//функция будет выводить сведения об  
    объектах JavaScript документа  
  
    //в документе не должен использоваться тег <base>
```

```
        {  
            document.write( i + " = " + document[i] + "<br>"  
);  
        }  
}
```

```
function break_p() {  
    var x=0;  
while (x<20){  
    if (x==16) break;/*цикл выполнится только до 16 */  
    ++x;  
    document.write(x+"<br>");  
    }  
}
```

```
function continue_p() {  
    var x=0;  
while (x<20){  
    x=x+1;  
    if (x==16) continue;  
    document.write(x+"<br>");  
    }  
}
```

```
function switch_p() {  
    x=parseInt(prompt("Введите оценку в 5 бальной  
системе","")); //parseInt преобразует строку, например "5" в  
число 5  
    switch(x) {  
        case 5: document.write("Вы набрали от 80 до 100  
балов");break;
```

```
        case 4: document.write("Вы набрали от 60 до 80
баллов");break;

        case 3: document.write("Вы набрали от 40 до 60
баллов");break;

        case 2: document.write("Вы набрали меньше 40
баллов");break;

        default:document.write("Таких оценок нет");

    }

}

function instr() {

    x=parseInt(prompt("Введите набранные балы на экзамене в
100 бальной системе", ""));

    document.write( (x>39) ? "Вы экзамен сдали" : "Вы
экзамен не сдали" );

}
```

Содержание отчета

Отчет по лабораторной работе должен содержать краткий конспект по теоретической части и выполненные в ходе работы файлы.

9 Методы ввода/вывода информации

Цель работы – ознакомление с принципами обмена информацией на стороне клиента.

Задачи работы – получение навыков создания сценариев обмена информацией на стороне клиента средствами JavaScript.

Теоретические сведения

Данные для сценариев JS могут быть получены следующими способами:

1. исходные данные сценария;
2. из форм методом POST, GET;
3. методом GET (через URL);
4. обработкой событий;
5. серверным сценарием.

Оборудование, программное обеспечение

Оборудование – ПК с установленной графической операционной системой (Windows или Linux).

Программное обеспечение – текстовый редактор, например, Блокнот, MS SharePoint Designer ; графический Web-браузер, например: Internet Explorer, Mozilla Firefox, Safari, Google Chrome или Opera.

Задание на работу

Создать страницы, содержащие следующие сценарии:

- 1) методы ввода/вывода информации;
- 2) динамическую верстка документа;
- 3) извлечение значения параметра из URL.

Ход работы

1. **Методы ввода/вывода информации.** Методы: `alert()`, `prompt()`, `confirm()`.

1.1. `alert()` – аргумент строка, окно-предупреждение с кнопкой ОК и строкой-аргументом.

```
<html>
<head>
<title>пример alert</title>
<script>
    alert("Окно с предупреждением");
</script>
```

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

1.2. `prompt ()` – два аргумента (“строка сообщение“, ”строка по умолчанию”). Выводит окно с полем для заполнения. Возвращает строку заданную пользователем.

```
<html>

<head>

<title>пример prompt</title>

<script>

    var name; //объявление переменной

    name=prompt("Введите пожалуйста Ваше
имя", "неизвестный");//переменной присваивается значение
введенное пользователем

    alert("Здравствуйете "+name+"!");

</script>

</head>

<body>

</body>

</html>
```

1.3 `confirm()`–аргумент строка, окно с сообщением-строкой и подтверждением ОК и Cancel.

```
<html>

<head>

<title>пример confirm</title>

<script>

    var name; //объявление переменной

    name=prompt("Введите пожалуйста Ваше
имя", "неизвестный");//переменной присваивается значение
введенное пользователем

    alert("Досвиданье "+name+"!");
```

```
        if(confirm("Закрыть окно?")) window.close();  
</script>  
</head>  
<body>  
</body>  
</html>
```

2. Динамическая верстка документа. Конструкция `document.write()` – параметр строка.

Пример 1

```
<html>  
<head>  
<title>метод document.write</title></head>  
<body>  
<script>  
        document.write("<h1 align='center'>Пример использования  
встроенных методов JavaScript</h1>");  
        window.status = "Пример метода  
document.write!"; //отображается в панели состояния браузера  
        setTimeout("window.status = 'Вот так!'",1500); //смена по  
таймеру строки в панели состояния браузера  
</script>  
</body>  
</html>
```

Пример 2

```
<html>  
<head>  
<title>Досье</title>  
</head>  
<body>
```

```

<script>
    var name, age, sex, img; //объявление переменных
    name=prompt("Введите пожалуйста Ваше полное имя","");
    age=prompt("Полных лет","");
    sex=prompt("Ваш пол","");
    img=prompt("адрес фото (150px на 150px)","");
    document.write("<table style='border: 1px solid
#000;'><tr><td>");
    document.write(' </td>');
    document.write('<td><h3
align="center">Досье</h3><table>');
    document.write('<tr><td>Ф.И.О.</td><td>'+age+'</td></tr>')
;
    document.write('<tr><td>Возраст,
лет</td><td>'+age+'</td></tr>');
    document.write('<tr><td>Пол</td><td>'+sex+'</td></tr>');
    document.write('</table></td></tr></table>');
</script>
</body>
</html>

```

Задание. Создайте динамическую страницу, в которой, на основе введенных Ф.И.О. студента, и оценок за прошедшую сессию, формировалась ведомость успеваемости. Номер группы и названия экзаменов должны быть заложены.

3. Извлечение значения параметра из URL.

Пример 1.

```

<html>
<head>
<title>url</title>
</head>

```

```
<body>
<a href="pr1.html?name=Masha">Имя 1</a><br />
<a href="pr1.html?name=Shasha">Имя 2</a><br />
<a href="pr1.html?name=Denis">Имя 3</a><br />
<script>
    if(window.location.href.indexOf("Masha")!=-1) alert
("Привет Masha");

    else if(window.location.href.indexOf("Shasha")!=-1) alert
("Привет Shasha");

    else if(window.location.href.indexOf("Denis")!=-1) alert
("Привет Denis");

    else alert("empty!!!");
</script>
</body>
</html>
```

Пример 2.

```
<html>
<head>
<title>url</title>
</head>
<body>
<a href="pr1.html?name=Masha">Имя 1</a><br />
<a href="pr1.html?name=Shasha">Имя 2</a><br />
<a href="pr1.html?name=Denis">Имя 3</a><br />
<script>
    var str_pos_1=window.location.href.indexOf("?name=");
    if(str_pos_1!=-1)
alert(window.location.href.substring(str_pos_1+"?name=".length
));
```

```
        else alert("empty!!!");  
</script>  
</body>  
</html>
```

Попробуйте вводить имя (любое) в строке адреса.

Задание. Создайте динамическую страницу, для формирования которой использовались бы данные из URL.

Содержание отчета

Отчет по лабораторной работе должен содержать краткий конспект по теоретической части и выполненные в ходе работы файлы.

10 Объектная модель документа

Цель работы – ознакомление с объектной моделью документа.

Задачи работы – получение навыков использования объектной модели документа различных браузеров в сценариях на стороне клиента средствами JavaScript.

Теоретические сведения

Объект – это программная единица, которую можно использовать в программах для выполнения различных задач. О текущем состоянии объекта можно судить только по значениям его свойств. Управлять состоянием объекта (если он допускает это) можно только посредством его свойств. Программный код объекта, конкретная реализация его функционирования неизвестны. Не знаем, как устроен объект, но знаем, как им пользоваться. Данная концепция называется еще концепцией черного ящика.

Объекты имеют фиксированные названия. *Пример*, окну браузера соответствует объект `window`, а HTML – документу, загруженному в браузер, – объект `document`. Это основные объекты. Объекты характеризуются тремя типами свойств:

- 1 собственно свойства;
- 2 методы;
- 3 другие объекты.

Свойства представляют собой переменные с фиксированными именами. Свойства имеют значения. Есть свойства, значения которых можно только читать (нельзя изменять). Значения других свойств можно изменять как обычные переменные. Эти свойства аналогичны элементам управления. Для доступа к свойству объекта используется синтаксис:

объект.свойство;

Пример, значением свойства `document.location` является URL-адрес HTML-документа.

Методы – это внутренние функции объектов. Они имеют фиксированные имена, могут принимать параметры и возвращать значения. Синтаксис применения метода:

объект.метод(список_параметров);

Пример, метод `window.open("www.mysite.ru/index.html")` открывает новое окно браузера и загружает в него страницу, расположенную по указанному адресу.

Некоторые методы не имеют параметров. Программы (объекты – это тоже программы) могут управляться как данными (т. е. значениями переменных), так и процедурами и функциями (т. е. методами).

Объект может иметь в качестве свойства другой объект. В этом случае говорят, что первый объект содержит (включает в себя) второй объект. При этом первый объект

называют также родительским (parent), а второй – дочерним, потомком или подобъектом (child). Обращение к свойству или методу объекта объект2, являющегося подобъектом объекта объект1:

объект1.объект2.свойство;

объект2.объект2.метод () ;

Пример, объект document является подобъектом объекта window. Если мы хотим что-то записать в документ, загруженный в текущее окно, то можем воспользоваться для этого методом write(). Вот пример:

window.document.write("Привет!")

Кроме свойств и методов, для каждого объекта определен список события, на которые он может реагировать. События имеют predetermined названия. Пример, событие «щелчок кнопкой мыши» обозначается onclick, «нажатие кнопки мыши» – onmousedown, «загрузка документа в браузер» – onload.

Все элементы страницы, созданные с помощью тэгов HTML, а также информация о состоянии браузера имеют объектное представление. Доступ к объектам осуществляется с помощью сценариев (скриптов).

Доступ к DOM-модели с помощью JavaScript. Три способа:

- 1 метод getElementByTagName() – возвращает список узлов с указанным именем тега;
- 2 метод getElementById() - обращение к элементу через уникальный атрибут id;
- 3 свойство documentElement с помощью древовидной модели Document.

Оборудование, программное обеспечение

Оборудование – ПК с установленной графической операционной системой (Windows или Linux).

Программное обеспечение – текстовый редактор, например, Блокнот, MS SharePoint Designer ; графические Web-браузеры: Internet Explorer, Mozilla Firefox, Opera.

Задание на работу

Задание 1. Создать сценарий страницы, позволяющий обращаться объектной моделью браузеров Internet Explorer, Mozilla Firefox, Opera.

Задание 2. Обратиться, используя объектную модель к изображениям на странице.

Задание 3. Создать интерактивную страницу с каталогом товаров оргтехники.

Ход работы

Задание 1.

```
<html>
<head>
<title>Объектная модель</title>
<script>
function show_1(){
    var p_list=document.getElementsByTagName("p");//получаем
    список всех элементов с тегом p
    var p_1=p_list.item(0);//первый элемент - первый абзац
    alert(p_1.tagName+" -
"+p_1.firstChild.nodeValue+"\n"+p_1.style.color);
}
function show_2(){
    var p_2=document.getElementById("text_2");//обращение по
    имени id
    alert(p_2.tagName+" - "+p_2.firstChild.nodeValue);
}
function show_3(){
    if(navigator.appName=="Microsoft Internet Explorer") {
        var bodyNode=document.documentElement.childNodes[1];
        var p_3=bodyNode.childNodes[3];
        alert(p_3.tagName+" - "+p_3.firstChild.nodeValue);
    }
    if(navigator.appName=="Opera") {
        var bodyNode=document.documentElement.childNodes[1];
        var p_3=bodyNode.childNodes[7];//О и N учитывают
перевод на строку
        alert(p_3.tagName+" - "+p_3.firstChild.nodeValue);
    }
    if(navigator.appName=="Netscape") {
        var bodyNode=document.documentElement.childNodes[2];
        var p_3=bodyNode.childNodes[7];
        alert(p_3.tagName+" - "+p_3.firstChild.nodeValue);
    }
}
</script>
</head>

<body>
```

```

<p id="text_1" align="justify" style="color:rgb(100%,60%,0%);"
onClick="show_1()">Метод getElementByTagName() - возвращает
список узлов с указанным именем тега</p>
<br>
<p id="text_2" style="color:rgb(0,255,0);"
onClick="show_2()">Метод getElementById() - обращение к
элементу через уникальный атрибут id</p>
<p id="text_3" align="right" onClick="show_3()">Свойство
documentElement древовидной модели DOM3</p>
</body>
</html>

```

Задание 2. Выполнить самостоятельно. Обратиться к другим свойствам элементов страницы. Вставить картинку и обратиться к ее свойствам.

Задание 3.

```

<html>

<head>

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

<title>Объектная модель2</title>

<style>

.t th {text-align:right;}

.w{width:150px;}

.w1{width:146px;}

</style>

<script>

function add_tovar() {

var yes;

mas=new Array();

var list=document.getElementById("tovar");

mas[0]=list.options[list.selectedIndex].text;

mas[1]=document.getElementById("text").value;

mas[2]=document.getElementById("nom").value;

```

```
mas[3]=document.getElementById("cost").value;

(document.getElementById("yes").checked)?mas[4]="полная":mas[4]
]="неполная";

tbody=document.getElementById("mast").firstChild;
tr=document.createElement("tr");

n=new Date();
n=parseInt(n)+"";
tr.setAttribute("id",n);
function_b="del_tovar("+n+")";

tbody.appendChild(tr);

for (i=0; i<mas.length;i++){
    td=document.createElement("td");
    text=document.createTextNode(mas[i]);
    td.appendChild(text);
    tr.appendChild(td);
}

var button=document.createElement("input");
button.setAttribute("type","button");
button.setAttribute("value","-");
button.setAttribute("onclick",function_b);

td=document.createElement("td");
td.appendChild(button);
tr.appendChild(td);
}
```

```
function del_tovar(id){
tbody=document.getElementById("mast").firstChild;
del_tr=document.getElementById(id);
tbody.removeChild(del_tr);
}
</script>
</head>

<body>
<table class="t" border="0" cellspacing="1" cellpadding="1">
  <tr>
    <th>Товар</th>
    <td><select id="tovar" class="w">
      <option value="0">Сканер</option>
      <option value="1">Принтер</option>
      <option value="2">Плоттер</option>
      <option value="3">Монитор</option>
    </select>
    </td>
  </tr>
  <tr>
    <th>Описание</th>
    <td><textarea id="text" class="w"></textarea></td>
  </tr>
  <tr>
    <th>Количество</th>
    <td><input type="text" id="nom" class="w1"/></td>
```

```

</tr>
<tr>
  <th>Цена, &#8364;</th>
  <td><input type="text" id="cost" class="w1"/></td>
</tr>
<tr>
  <th>Комплектация</th>
  <td>Полная<input type="checkbox" id="yes" /></td>
</tr>
<tr>
  <th colspan="2"><input type="button" onclick="add_tovar()"
value="Добавить" class="w"/></th>
</tr>
</table>
<br />
<br />
<table id="mast" border="0" cellspacing="1" cellpadding="1"
style="width:500px;border:double 3px #CCC">
  <tr style=" background-color:#DDD">
    <th>Товар</th>
    <th>Описание</th>
    <th>Кол-во</th>
    <th>Цена, &#8364;</th>
    <th>Комплектация</th>
    <th>Удалить</th>
  </tr>
</table>
</body></html>

```

Содержание отчета

Отчет по лабораторной работе должен содержать краткий конспект по теоретической части и выполненные в ходе работы файлы.

11 Серверные сценарии MySQL+PHP

Цель работы – ознакомление с работой серверных приложений.

Задачи работы – получение навыков работы с phpMyAdmin, получения навыков создания и использования базы данных в сценариях на стороне сервера.

Теоретические сведения

Серверные программы делятся на следующие четыре вида.

Исполняемые программы, работающие через интерфейс CGI (Common Gateway Interface – общий интерфейс обмена), так называемые CGI-программы. Эта разновидность серверных программ – самая старая, однако отнюдь не устаревшая.

Расширения Web-сервера (приложения формата ISAPI, NSAPI, модули расширения Apache и т. п.). Новый способ, позволяющий встраивать серверные программы в сам Web-сервер, делая их его составными частями. Впервые предложен фирмой Microsoft для их сервера Microsoft Internet Information Server (интерфейс ISAPI) и разработчиками популярного бесплатного Web-сервера Apache.

Активные серверные страницы (ASP, JSP и др.). Фактически это обычные статические Web-страницы, сохраненные в файлах, которые, кроме обычного HTML-кода, включают в себя команды, обрабатываемые либо самим Web-сервером, либо его расширением. Также новый способ, впервые предложенный Microsoft для того же Internet Information Server.

Серверные сценарии, написанные на интерпретируемом языке (Perl, Python, VBScript, JavaScript и др.). Обычные сценарии, работающие через интерфейс CGI или ISAPI на стороне сервера.

CGI-программы представляют собой обычные исполняемые файлы, написанные на любом языке программирования и откомпилированные в машинный код процессора. Они не имеют интерфейса пользователя (как и все серверные программы), а работают с Web-сервером, получают от него входные данные и ему же пересылают результаты своей работы. Запускаются они самим Web-сервером, когда в них возникает нужда (когда необходимо обработать полученные от пользователя данные), и работают под управлением операционной системы серверного компьютера. При этом, если Web-серверу поступает одновременно несколько запросов на обработку данных от пользователей, он запускает соответствующее количество копий CGI-программы.

К достоинствам CGI-программ можно отнести легкость создания (многие среды разработки программ поддерживают создание таких приложений, в частности популярнейший Borland Delphi, начиная с версии 3) и простоту отладки. Также, поскольку CGI-приложения представляют собой независимые программы, они

выполняются отдельно от Web-сервера (как говорят программисты и системные администраторы, выполняются в другом адресном пространстве). Это значит, что при сбое в CGI-программе завершается только она – сам Web-сервер остается "на плаву". А недостаток у CGI-программ всего один: большой расход системных ресурсов, поскольку для обработки каждого набора данных запускается отдельная копия серверной программы. И если Web-серверу поступит слишком много запросов на обработку данных, серверный компьютер может и зависнуть.

Расширения Web-сервера – более новая разновидность серверных программ. Они представляют собой обычные библиотеки DLL, в которых реализована вся логика серверной программы. Такие библиотеки как бы встраиваются в программу Web-сервера и работают как ее неотъемлемая часть. Поскольку библиотеки DLL работают только в среде Windows, для того чтобы создавать расширения в иных операционных системах, были придуманы и другие форматы. В частности, модули расширения сервера Apache не являются библиотеками DLL,

Именно в виде библиотек DLL создаются расширения Web-серверов Internet Information Server фирмы Microsoft и Netscape Web Server фирмы Netscape. В первом случае расширения имеют формат ISAPI (Internet Server Application Programming Interface – интерфейс программирования приложений интернет-сервера), а во втором – NSAPI (Netscape Server Application Programming Interface – интерфейс программирования приложений сервер Netscape). Формат модулей расширения Apache так и называется – модули Apache.

Достоинство у расширений Web-сервера одно: бережный расход системных ресурсов. Дело в том, что для обработки всех наборов данных пользователя запускается всего один экземпляр расширения, который отнимает существенно меньше ресурсов, чем уйма запущенных CGI-программ. Однако расширения труднее создавать и отлаживать, к тому же они не так безопасны.

Как CGI-программы. Поскольку они работают как часть Web-сервера, любая ошибка в расширении приведет к зависанию сервера.

Оба описанных выше вида серверных программ обладают одним огромным недостатком. Прежде чем они смогут работать, они должны быть написаны на языке программирования и откомпилированы в машинные коды процессора, что отнимает много времени, особенно при отладке. Конечно, откомпилированные программы работают быстрее интерпретируемых, т. е. тех, где каждая инструкция читается, расшифровывается и обрабатывается специальной программой-интерпретатором. Но у интерпретируемых программ есть и свои преимущества, главными из которых являются простота и быстрота написания. Две следующие разновидности серверных программ, которые будут описаны, как раз будут интерпретируемыми.

Как уже говорилось, активные серверные страницы – это обычные Web-страницы, включающие в себя особые серверные сценарии, выполняемые самим Web-сервером или специальной серверной программой (CGI-приложением или расширением Web-сервера). В частности, ASP (Active Server Pages – активные серверные страницы), поддерживаемые Microsoft Internet Information Server, и JSP (Java Server Pages – серверные страницы, написанные на JavaScript), поддерживаемые рядом других Web-серверов, работают именно таким образом. Серверные страницы ASP пишутся на языках JavaScript и VBScript, а JSP – только на JavaScript.

Достоинства активных серверных страниц: легкость и быстрота написания и простота отладки. Кроме того, поскольку активные серверные страницы -- это обычные Web-страницы с "вкраплениями" программного кода, их написание легко освоют все, кто знаком с HTML. Недостаток: относительная медлительность и повышенные требования к системным ресурсам.

Серверные сценарии подобны активным серверным страницам тем, что являются интерпретируемыми, однако представляют собой "чистый" программный код, без HTML-"примесей". Интерпретатор практически всегда представляет собой CGI-программу, однако ничто не мешает разработать его в виде расширения Web-сервера. Сценарии обычно пишутся на языке программирования Perl, специально предназначенном для обработки текста; также используются языки Python, JavaScript, VBScript и даже (как говорят) язык командных файлов MS-DOS. Фактически писать сценарии можно на любом языке программирования, для которого есть интерпретатор.

Достоинства и недостатки серверных сценариев те же, что у активных серверных страниц. Однако сценарии потребляют исключительно много системных ресурсов, даже больше, чем CGI-приложения. Ведь для обработки каждого набора данных пользователя запускается своя копия интерпретатора, а интерпретатор, в свою очередь, расходует много ресурсов на обработку сценария. И все же, несмотря на это, сценарии – самый популярный способ создания серверных программ. Оборудование, программное обеспечение

Дамп – представляет собой SQL – запросы, по которым можно воссоздать, как таблицу, так и базу данных. Дампы используют для резервного копирования и для переноса таблиц и баз данных с хоста на хост (т.е. можно носить из дома в университет на занятия и наоборот).

Оборудование, программное обеспечение

Оборудование – ПК с установленной графической операционной системой (Windows или Linux).

Программное обеспечение – текстовый редактор, например, Блокнот, MS SharePoint Designer ; установленный пакет Denwer; графический Web-браузер: Internet Explorer, Mozilla Firefox или Opera.

Задание на работу

Задание 1. Создать базу данных для хранения информации о зарегистрированных пользователях. Создать файл с формой для добавления записей в созданную базу.

Задание 2. Создать базу данных каталога товаров. Сохранить дамп базы данных.

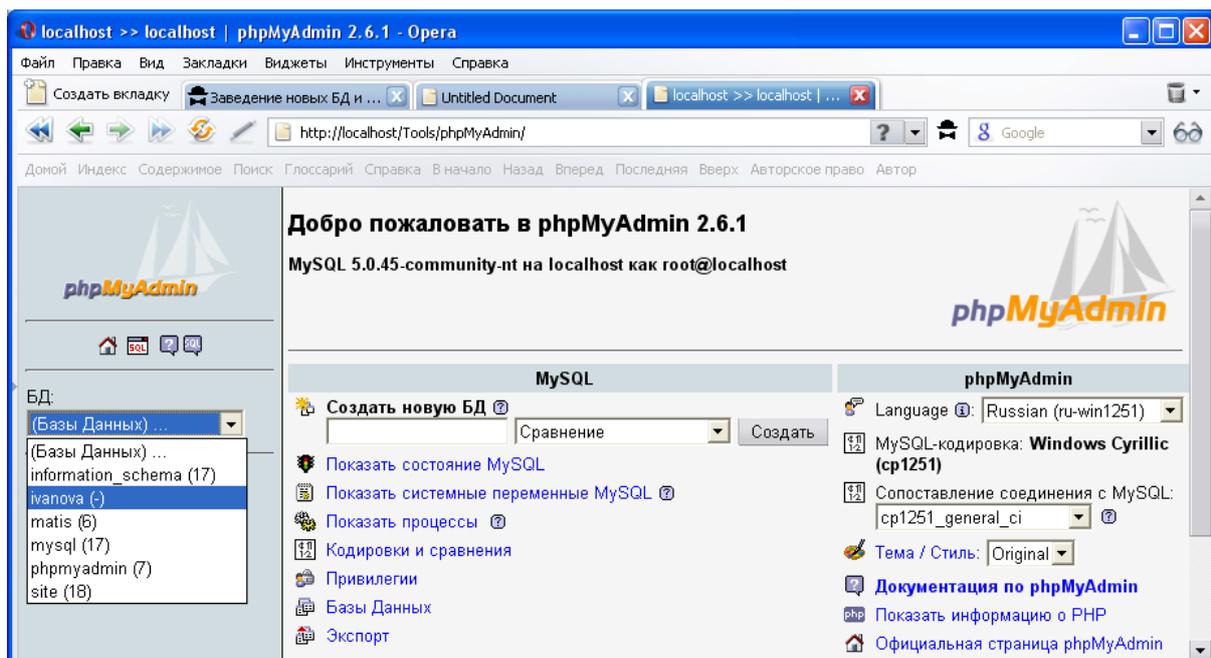
Ход работы

Задание 1.

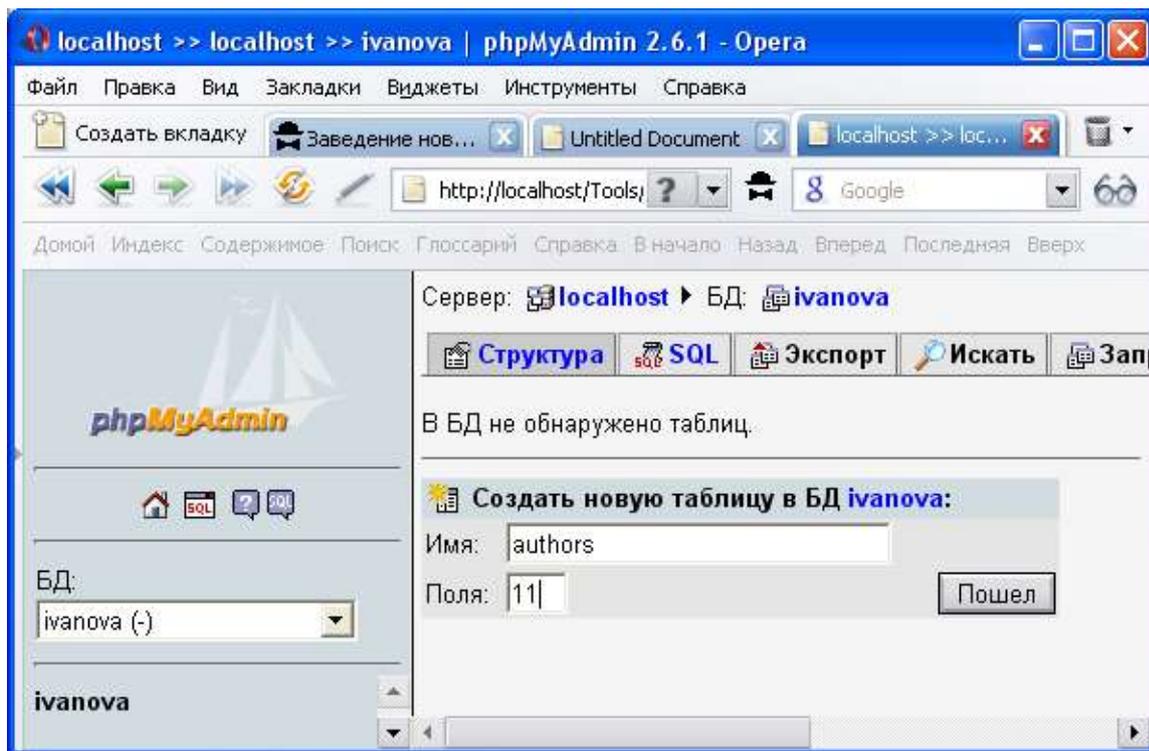
1. Откройте в Орега следующую вкладку <http://localhost/denwer/>. В разделе **Утилиты** перейдите по ссылке **Заведение новых БД и пользователей MySQL**.

Пароль администратора MySQL:	Ничего не вводить!
Имя базы данных: ivanova	Своя фамилия или ее часть на латинице (английскими буквами)
Логин пользователя: ivanova	Тот же что и имя базы данных
Пароль: iv62ps!	Пароль свой (запомнить и записать!)
...еще раз: iv62ps!	

2. Откройте в Орега вкладку <http://localhost/Tools/phpMyAdmin/>. Выберите БД ivanova(-) (выбрать свою базу).



3. Создайте таблицу **authors** (см. рисунок ниже). Для продолжения выбрать **Пошел**.



localhost >> localhost >> ivanova >> authors | phpMyAdmin 2.6.1 - Opera

Файл Правка Вид Закладки Виджеты Инструменты Справка

Создать вкладку Заведение новых БД и ... Untitled Document localhost >> localhost >...

http://localhost/Tools/phpMyAdmin/ Google

Домой Индекс Содержимое Поиск Глоссарий Справка В начало Назад Вперед Последняя Вверх Авторское право Автор

Сервер: localhost ▶ БД: ivanova ▶ таблица: authors

Поле	Тип	Длины/ Значения*	Сравнение	Атрибуты	Ноль	По умолчанию**	Дополнительно				
id_author	INT				not null		auto_increment	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
name	TINYTEXT				null			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
password	TINYTEXT				null			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
email	TINYTEXT				null			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
home_url	TINYTEXT				null			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
icq	TINYTEXT				null			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
about	TINYTEXT				null			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
photo	TINYTEXT				null			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
time	DATETIME				null	NULL		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
last_time	DATETIME				null	NULL		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
status	INT				null	NULL		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Комментарий к таблице:

Тип таблицы: По умолчанию

Сравнение:

Add field(s)

* Для типов поля "enum" и "set", введите значения по этому формату: 'a','b','c'...
Если вам понадобится ввести обратную косую черту ("\") или одиночную кавычку (") среди этих значений, поставьте перед ними обратную косую черту (например, \'xyz' или a\b).

** Для значений по умолчанию просто введите единственное значение без экранирования и кавычек, используя этот формат: a

*** Пожалуйста, введите значения для трансформации, используя этот формат: 'a','b','c'...
Если вам нужно поставить бэкслэш ("\") или кавычку ("), экранируйте их (например \'xyz' or a\b).
Для списка доступных опций трансформации и трансформаций их MIME-типов кликните на [описание трансформаций](#)

4. Заполните поля как на рисунке.

После выбрать Сохранить.

localhost >> localhost >> ivanova >> authors | phpMyAdmin 2.6.1 - Opera

Файл Правка Вид Закладки Виджеты Инструменты Справка

Создать вкладку Заведение новых БД и ... Untitled Document localhost >> localhost >...

http://localhost/Tools/phpMyAdmin/

Домой Индекс Содержимое Поиск Глоссарий Справка В начало Назад Вперед Последняя Вверх Авторское право Автор

Сервер: localhost ▶ БД: ivanova ▶ таблица : authors

Структура Обзор SQL Искать Вставить Экспорт Операции Очистить Уничтожить

таблица authors была создана.

SQL-запрос:

```
CREATE TABLE `authors` (  
  `id_author` INT NOT NULL AUTO_INCREMENT ,  
  `name` TINYTEXT ,  
  `password` TINYTEXT ,  
  `email` TINYTEXT ,  
  `home_url` TINYTEXT ,  
  `icq` TINYTEXT ,  
  `about` TINYTEXT ,  
  `photo` TINYTEXT ,  
  `time` DATETIME DEFAULT NULL ,  
  `last_time` DATETIME DEFAULT NULL ,  
  `status` INT DEFAULT NULL ,  
  PRIMARY KEY (`id_author` )  
);
```

[Правка] [Создать PHP-код]

	Поле	Тип	Сравнение	Атрибуты	Ноль	По умолчанию	Дополнительно	Действие
<input type="checkbox"/>	id_author	int(11)			Да	NULL	auto_increment	
<input type="checkbox"/>	name	tinytext	cp1251_general_ci		Да	NULL		
<input type="checkbox"/>	password	tinytext	cp1251_general_ci		Да	NULL		
<input type="checkbox"/>	email	tinytext	cp1251_general_ci		Да	NULL		
<input type="checkbox"/>	home_url	tinytext	cp1251_general_ci		Да	NULL		
<input type="checkbox"/>	icq	tinytext	cp1251_general_ci		Да	NULL		
<input type="checkbox"/>	about	tinytext	cp1251_general_ci		Да	NULL		
<input type="checkbox"/>	photo	tinytext	cp1251_general_ci		Да	NULL		
<input type="checkbox"/>	time	datetime			Да	NULL		
<input type="checkbox"/>	last_time	datetime			Да	NULL		
<input type="checkbox"/>	status	int(11)			Да	NULL		

↑ Отметить все / Снять отметку со всех С отмеченными

5. В верхнем меню выберите **Вставить** и добавьте первую запись.

Поле	Тип	Функция	Ноль	Значение
id_author	int(11)	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
name	tinytext	<input type="text"/>	<input type="checkbox"/>	Vasay2008
password	tinytext	MD5 <input type="text"/>	<input type="checkbox"/>	vasay2008!ay
email	tinytext	<input type="text"/>	<input type="checkbox"/>	vasay@yandex.ru
home_url	tinytext	<input type="text"/>	<input type="checkbox"/>	http://vasayiv.com/
icq	tinytext	<input type="text"/>	<input type="checkbox"/>	237533566
about	tinytext	<input type="text"/>	<input type="checkbox"/>	Занимаюсь робототехникой и нанотехнологиями. Очень общительный.
photo	tinytext	<input type="text"/>	<input type="checkbox"/>	vasey.jpg
time	datetime	<input type="text"/>	<input type="checkbox"/>	2008-09-01 21:56:18 
last_time	datetime	<input type="text"/>	<input type="checkbox"/>	2008-10-24 21:56:33 
status	int(1)	<input type="text"/>	<input type="checkbox"/>	1

6. Добавьте еще две-три записи (придумайте сами). Чтобы исправить добавленные записи (если требуется) выберите в меню **Обзор**, отметьте галочкой записи требующие исправление и выберите **С отмеченными:** .

7. Создайте файл **bd_config.php** для подключения к базе данных (подключение к базе данных лучше выполнять в отдельном файле, который можно, подключать потом к страницам проекта, где требуется использование базы):

<?

```
$dblocation="localhost"; //сетевой адрес сервера (на локальной машине localhost или 127.0.0.1)
```

```
$dbuser="ivanova"; //пользователь имеющий доступ к базе данных
```

```
$dbpasswd="iv62ps!"; //пароль пользователя
```

```
$dbname="ivanova"; //имя базы данных
```

```
/*
```

Соединяемся с сервером базы данных (MySQL) и присваиваем результат переменной \$dbconnect

функция `mysql_connect(сервер [,пользователь[,пароль]])` - устанавливает соединение с MySQL, в случае неудачи возвращает FALSE, если успешно, то идентификатор соединения (идентификатор_ресурса)

`mysql_connect(сервер [,пользователь[,пароль]])` - в пояснениях к программным кодам принято в квадратных скобках указывать параметры, задавать которые необязательно!!!

```
*/
```

```
$dbconnect=@mysql_connect($dblocation,$dbuser,$dbpasswd);
```

```
/*
```

если \$dbconnect=0 (т.е. FALSE), то соединиться сервером MySQL не удалось, поэтому выводим сообщения об ошибке

```
*/
```

```
if(!$dbconnect){
```

```
    echo "Нет соединения с сервером базы данных";
```

```
exit();
```

```
}
```

```
/*
```

Соединяемся с базой данных \$dbname="ivanova" или выводим сообщения об ошибке, если соединиться не удалось

функция mysql_select_db(имя_базы_данных [,идентификатор_ресурса]) - устанавливает соединение с конкретной базой, в случае неудачи возвращает FALSE, если успешно, то TRUE

```
*/
```

```
if(!@mysql_select_db($dbname,$dbconnect)){
```

```
    echo "Нет соединения с базой данных";
```

```
    exit();
```

```
}
```

```
?>
```

8. Создайте файл **page1.php** выводящий данные из таблицы **authors**.

```
<?
```

```
include_once("bd_config.php"); //подключаем скрипт, выполняющий соединение с базой данных
```

```
/*
```

функция mysql_query("Запрос на SQL" [, идентификатор_ресурса]) - возвращает дескриптор запроса (дескриптор_запроса) в случае успешного выполнения, а в случае неудачи возвращает FALSE

select * from authors - выбираем все столбцы из таблицы authors

```
*/
```

```
$savl=mysql_query("select * from authors;");
```

```
if ($savl) {
```

```
/*
```

функция `mysql_fetch_assoc`(дескриптор_запроса) - обрабатывает результат запроса и возвращает ассоциативный массив

```
*/
```

```
    echo "<h3>Использование функции для обработки результата запроса и вывода как  
    ассоциативного массива</h3>";
```

```
/*
```

До тех пор (`while`) пока есть ряды в запросе (это столбцы в таблице `authors`) помещаем их в ассоциативный массив `$author`. В качестве индексов массива будут имена рядов таблицы `authors`.

```
*/
```

```
    while($author=mysql_fetch_assoc($savl)){  
        echo $author['id_author']." ".$author['name']." ".$author['email']."<br />";  
    }  
}
```

```
}
```

```
?>
```

9. Создайте файл **page2.php** выводящий данные из таблицы **authors**.

```
<?
```

```
include_once("bd_config.php");
```

```
$savl=mysql_query("select id_author, name, email, icq, time from authors;");
```

```
/* "select id_author, name, email, icq, time from authors" – выбираем столбцы id_author, name, email,  
icq, time в таблице authors*/
```

```
if ($savl) {
```

```
    echo "<h3>Использование функции для обработки результата запроса и вывода как  
    неассоциативного массива</h3>";
```

```
/*
```

функция `mysql_fetch_row`(дескриптор_запроса) - обрабатывает результат запроса и возвращает неассоциативный массив (строки выбранных столбцов), поэтому используем функцию `list` для превращения результатов запроса в переменные

```
*/
```

```

while(list($id_author,$name,$email,$icq,$time)=mysql_fetch_row($avt)) {

    echo $id_author.". $name, email - $email, ICQ - $icq, зарегистрирован с $time.<br />";

}

}

?>

```

10. Создайте файл **page3.php** выводящий данные из таблицы **authors**.

```

<?

include_once("bd_config.php");

$avt=mysql_query("select * from authors;");

if ($avt) {

/*функция mysql_fetch_array(дескриптор_запроса [, тип_результата]) - обрабатывает результат
запроса и возвращает массив, если тип_результата:

MYSQL_ASSOC - как ассоциативный массив
MYSQL_NUM - как численный массив
MYSQL_BOTH - как численный и ассоциативный массив (это значение по умолчанию)
*/

    echo "<h3>Использование функции для обработки результата запроса и вывода как
массива</h3>";

    while($author=mysql_fetch_array($avt)){

        echo $author['id_author'].". $author[1], ICQ ".$author['icq'].". $author[6]<br />";

    }

}

?>

```

11. Создайте файл **page4.php** добавляющий данные в таблицы **authors**.

<?

```
include_once("bd_config.php");
```

```
//Добавляем запись в таблицу authors
```

```
mysql_query ("INSERT INTO authors VALUES ('', 'Petrov', MD5('pet5h'), 'petrov.kio@yandex.ru',  
NULL, '355413365', 'Студент факультета ЕН ТулГУ.', 'petrov.jpg', NOW(), NOW(), '1')");
```

```
//эта часть для вывода данных
```

```
$savl=mysql_query("select * from authors;");
```

```
if ($savl) {
```

```
while($sauthor=mysql_fetch_array($savl)){
```

```
    echo "$sauthor[0]. $sauthor[1], ICQ . $sauthor[5]. $sauthor[6]<br />";
```

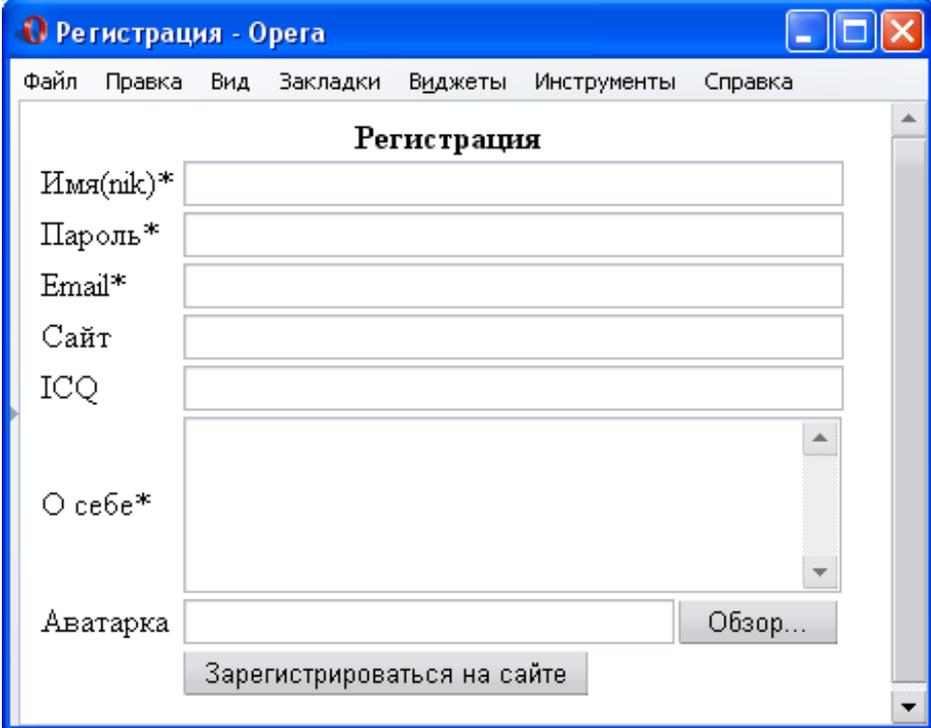
```
}
```

```
}
```

?>

В <http://localhost/Tools/phpMyAdmin/> посмотреть **Обзор** таблицу **authors**.

12. Создать файл с формой добавления записей в таблицу authors (см. рисунок).



The image shows a screenshot of a web browser window with the title "Регистрация - Орега". The browser's menu bar includes "Файл", "Правка", "Вид", "Закладки", "Виджеты", "Инструменты", and "Справка". The main content area is titled "Регистрация" and contains a registration form with the following fields:

- Имя(ник)*: Text input field
- Пароль*: Text input field
- Email*: Text input field
- Сайт: Text input field
- ICQ: Text input field
- О себе*: Text area with a vertical scrollbar
- Аватарка: Text input field with a button labeled "Обзор..." to its right

At the bottom of the form is a large button labeled "Зарегистрироваться на сайте".

Задание 2.

13. Скопируйте папку **dress** к себе в каталог.

14. Создайте в базе данных таблицу **dress**. Со следующими полями см. рисунок.

| Поле | Тип | Длины/
Значения* | Ноль | По
умолчанию** | Дополнительно | | | | | | Комментарии |
|----------|----------|---------------------|----------|-------------------|----------------|----------------------------------|-----------------------|-----------------------|----------------------------------|--------------------------|----------------|
| id_dress | INT | | not null | | auto_increment | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="checkbox"/> | |
| articul | VARCHAR | 6 | null | | | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="checkbox"/> | артикул |
| marker | TINYTEXT | | null | | | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="checkbox"/> | производитель |
| cost | INT | | null | | | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="checkbox"/> | цена |
| write_up | TEXT | | null | | | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="checkbox"/> | описание |
| pic | TINYTEXT | | null | | | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="checkbox"/> | картинка |
| hidden | TINYINT | 1 | null | 0 | | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="checkbox"/> | показать/скрыт |

Комментарий к таблице:

Тип таблицы:

Сравнение:

15. Добавьте 15 записей (по числу картинок в папке **dress**) с описанием товара в таблицу **dress**. К примеру:

| id_dress | articul
артикул | marker
производитель | cost
цена | write_up
описание | pic
картинка | hidden
показать/скрыть |
|----------|--------------------|-------------------------|--------------|---|-----------------|---------------------------|
| 1 | S5600 | Hot Sauce | 2160 | Легкое летнее мини-платье с завязками на шее украш... | 01.jpg | 0 |
| 2 | S4654 | Hot Sauce | 3400 | Короткое платье с завязками на шее. | 02.jpg | 0 |

16. В главном меню выбрать **Экспорт**, отметить галочкой поля **Добавить в комментарии** -> **Комментарии** и **послать** (см. рисунок). Далее выбрать **Пошел**, затем **Сохранить на диске**

Просмотреть дампы таблиц

Экспорт

- SQL
- LaTeX
- CSV для данных MS Excel
- CSV данные
- XML

Опции SQL

Добавить частные комментарии к header (\n разбивает строки):

Экспортировать при переводе

Выключить проверку инородных ключей

Структура:

- Добавить удаление таблицы
- Добавить IF NOT EXISTS
- Добавить значение AUTO_INCREMENT
- Обратные кавычки в названиях таблиц и полей

Добавить в комментарии

- Создание/Обновление/Проверка дат
- Связи
- Комментарии
- MIME-тип

SQL export compatibility:

Данные:

- Полная вставка
- Расширенные вставки
- Использовать запаздывающие вставки
- Игнорировать вставки
- Использовать шестнадцатеричные (hexadecimal) бинарные поля

Тип экспорта:

Дамп записей, начиная с .

послать

Шаблон имени файла: (запомнить шаблон)*

Сжатие

- Нет
- архивировать в zip
- архивировать в gzip

(сохраните на диск D к себе в папку!!!). Таким образом, вы создали **дампы** таблицы **dress**.

17. Выберите в главном меню **Уничтожить**, и уничтожьте таблицу **dress**.

18. В главном меню выберите **SQL** и в **Месторасположение текстового файла** указать сохраненный дампы таблицы **dress**. Далее выбрать **Пошел** (см. рисунок). По дампы таблицы **phpMyAdmin** выполняя SQL – запросы создаст таблицу **dress**.

Структура SQL Экспорт Искать Запрос по примеру

Выполнить SQL запрос(ы) на БД : ?

Показать данный запрос снова Пошел

Или

Месторасположение текстового файла:
"D:\WebCтр\php\dres Обзор... (Минимальный размер: 2,048KB)

Сжатие:
 Автодетект Нет архивировать в gzip

Кодировка файла: utf8 Пошел

Или

Закладка на SQL-запрос: Пошел

Переменная ?

Выполнить Только просмотр Удалить Пошел

Содержание отчета

Отчет по лабораторной работе должен содержать краткий конспект по теоретической части и выполненные в ходе работы файлы.

12 Сценарии двунаправленного обмена данными, MooTools+Ajax+PHP

Цель работы – ознакомление с технологией создания двунаправленного обмена данными между пользователем и веб-сервером.

Задачи работы – получение навыков работы с готовыми библиотеками, для создания двунаправленного обмена данными между пользователем и веб-сервером.

Теоретические сведения

MooTools.js представляет собой frameworks для javascript.

frameworks - это модульные, объектно-ориентированные библиотеки для rapid-разработки. MooTools берет на себя работу с объектной моделью браузера (например, используется один метод addEvent для различных браузеров вместо addEventListener(Mozilla, Safari, Opera) и attachEvent(IE)). Библиотека определяет браузер и какой метод он поддерживает, выполняет обработку ошибок – если такой элемент, можно ли привязать события и пр., и берет на себя работу по обработке свойств элементов.

Данная библиотека содержит готовые объекты, которые значительно облегчают программирование на javascript. MooTools включает в себя следующие модули:

Core – базовые функции

Class – работа с классами элементов

Native – работа с массивами, строками, функциями, числами

Element – работа с элементами (события, фильтры, формы, размеры)

Window – размеры окна, DOM

Effects – стили, скролинг, слайдинг и переходы

Drag – перемещения

Remote – AJAX, COOKIES, подключение стилевых и скриптовых файлов «на лету»

Plugins – скроллы, слайдеры, цветовые палитры, сортировки, подсказки (tips), группировки элементов, «аккордеон».

Синтаксис

\$(Id) метод mooTools для получения ссылки на элемент с Id – вместо (то же)
document.getElementById('Id');

\$('.class') метод mooTools для получения ссылки на элементы с определенным классом (возвращает разумеется массив элементов с данным классом– следовательно чтобы обратиться потом к конкретному элементу нужно вызвать его, например, по индексу \$('.class')[0]);

\$('.tag') метод mooTools для получения ссылки на элементы с определенным тегом (возвращает массив)– вместо (то же) document.getElementsByTagName('tag');

Так как \$('что-то') используется для получения массива, то можно писать все подряд, например \$('p', '#print', el) – будет массив из всех элементов с тегом p, какого-то элемента с Id='print' и элемента на который ссылается переменная el (где-то выше у нас уже должно быть записано, что-то вроде: var el=\$(Id) или var el=document.getElementById('Id') или еще как-нибудь)

для связи элемента с событием в mootools применяется метод addEvent:

```
объект.addEvent('событие',имяФункции);  
или  
объект.addEvent('событие',функция([, параметры]){  
    [; переменные];  
    операторы функции;  
    }  
);
```

window.addEvent('domready', function(){... используем в mootools вместо <body onload="какая-то_функция()">

в mootools событие 'domready' тоже что 'onload'

объект.setStyle('свойство', 'значение') из библиотеки mootools применяем вместо
объект.style.свойство='значение'

AJAX

AJAX (от англ. Asynchronous JavaScript and XML – «асинхронный JavaScript и XML») – это подход к построению интерактивных пользовательских интерфейсов веб-приложений. При использовании AJAX веб-страница не перезагружается полностью в ответ на каждое действие пользователя. Вместо этого с веб-сервера догружаются только нужные пользователю данные. AJAX – один из компонентов концепции DHTML.

AJAX по-английски произносится эй-джэкс, по-русски довольно распространено ая кс.

Об AJAX заговорили после появления 18 февраля 2005 г. статьи Джесси Джеймса Гарретта (Jesse James Garrett) «Новый подход к веб-приложениям». AJAX – это не самостоятельная технология, а идея.

AJAX базируется на двух основных принципах:

использование DHTML для динамического изменения содержания страницы;

использование технологии динамического обращения к серверу «на лету», без перезагрузки

всей страницы полностью, например:

с использованием XMLHttpRequest;

через динамическое создание дочерних фреймов;

через динамическое создание тега < script>.

Использование этих двух принципов позволяет создавать намного более удобные веб-интерфейсы пользователя на тех страницах сайтов, где необходимо активное взаимодействие с пользователем. Использование AJAX стало наиболее популярно после того, как компания Google начала активно использовать его при создании своих сайтов, таких как Gmail, Google Maps и Google Suggest. Использование AJAX на этих сайтах подтвердило эффективность использования данного подхода.

Flash стек технологий в dbлт ActionScript 3, Flex, Flash Remoting составляет технологическую основу так называемых RIA (Rich Internet Applications) активно продвигаемых Macromedia (теперь часть Adobe). RIA являются главными конкурентами Ajax.

Оборудование, программное обеспечение

Оборудование – ПК с установленной графической операционной системой (Windows или Linux).

Программное обеспечение – текстовый редактор, например, Блокнот, MS SharePoint Designer ; пакет Denver; библиотека Mootools, графический Web-браузер, например: Internet Explorer, Mozilla Firefox, Safari, Google Chrome или Opera.

Задание на работу

Разработать динамические приложения осуществляющие «на лету» проверку вводимых данных по базе данных веб-сервера, вручную и с использованием готовой библиотеки.

Ход работы

1. Скопируйте файл mootools.js к себе в папку.

2. Создайте файл page1.html, просмотрите работу в браузере.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />
<title>Использование безымянных(анонимных) функций: Пример 2</title>
<script src="mootools.js"></script>
<script>
window.addEvent('domready', function(){
    $('nameOn').addEvent('click', function(){
        $("print").setStyle('color', '#F00');
        $("print").innerHTML="пример с библиотекой";
    }
    );
});
</script>
</head>

<body>
<p id="nameOn">кликни здесь</p>
<div id="print"></div>
</body>
</html>
```

3. Создайте файл page2.html, просмотрите работу в браузере.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

```

<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />
<title>Использование безымянных(анонимных) функций: Пример 2</title>
<script src="mootools.js"></script>
<script>
window.addEvent('domready', function(){
    el=$("print");
    el.setStyle('opacity', 0);
    var eff = new Fx.Morph(el, {duration: 1000});
        $('nameOn').addEvent('click', function(){
            el.innerHTML="пример с библиотекой";
            eff.start({
                'font-size': [9, 16],
                'opacity': [0, 1],
                'color': '#F00',
                'font-weight': 900
            });
        });
    });
});
/*

```

тот же самый пример, но с эффектом Morph
var eff = new Fx.Morph(el, {duration: 1000}); создаем свой экземпляр класса Fx.Morph с именем eff и следующими параметрами - ссылкой на элемент el над которым мы будем это проделывать, и временем выполнения (можно и другие параметры задать)
в нужном месте вызываем метод класса Morph start, который позволяет плавно менять свойства CSS, в скобках [] указывают от какого до какого

```

*/
</script>
</head>

```

```

<body>
<p id="nameOn">кликни здесь</p>
<div id="print"></div>
</body>

```

```

</html>

```

4. Создайте файл page3.html, посмотрите работу в браузере.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />
<title>Использование безымянных(анонимных) функций: Пример 2</title>
<script src="mootools.js"></script>
<script>

```

```

/*
тот же самый пример, но создан как функция
*/
function printStr(){
    var el=$("#print");
    var eff = new Fx.Morph(el, {duration: 1000});
    el.setStyle('opacity', 0);
    eff.start({
        'font-size': [9, 16],
        'opacity': [0, 1],
        'color': '#F00',
        'font-weight': 900
    });
    el.innerHTML="пример с библиотекой";
}
</script>
</head>

<body>
<p onclick="printStr()">кликни здесь</p>
<div id="print"></div>
</body>
</html>

```

5. AJAX (без библиотеки js). Создайте следующие файлы:

ajax.html	ajax.js
<pre> <html> <head> <title>AJAX 1</title> <script type="text/javascript" src="ajax.js"></script> </head> <body onload="process()"> <label>Введите имя пользователя</label> <input type="text" id="name" /> <div id="show"></div> </pre>	<pre> var xmlhttp=createXmlHttpRequestObject(); function createXmlHttpRequestObject() { var xmlhttp; if(window.ActiveXObject) { try { xmlhttp=new ActiveXObject("Microsoft.XMLHTTP"); } catch (e) { </pre>

<pre></body> </html></pre>	<pre>xmlHttp=false; } }</pre>
<p>php_ajax.php</p>	<pre>else {</pre>
<pre><? header('Content-Type:text/xml'); echo '<?xml version="1.0" encoding="windows-1251" standalone="yes"?>'; echo '<ответ>'; \$name_m=\$_GET['name_m']; \$privat=array("admin","localadmin","user"); if (in_array(\$name_m,\$privat)) echo 'Есть пользователь [strong style="color:#00F;"]'.htmlentities(\$name_ m).'/strong]!'; elseif(trim(\$name_m)==") echo "[h1]Пользователь?[h1]"; else echo '[strong style="color:#F00;"]'.htmlentities(\$name_ m).'/strong] нет пользователя с таким именем!'; echo '</ответ>'; ?></pre>	<pre>try { xmlHttp=new XMLHttpRequest(); } catch (e) { xmlHttp=false; } } if(!xmlHttp) alert("Объект XMLHttpRequest не создан!!!"); else return xmlHttp; } function process() { if(xmlHttp.readyState==4 xmlHttp.readyState==0) { name_m=encodeURIComponent(document.getElementByI d("name").value); xmlHttp.open("GET","php_ajax.php?name_m="+name_m, true); xmlHttp.onreadystatechange=handleServerResponse; xmlHttp.send(null); } else setTimeout('process()',1000); } function handleServerResponse() { if(xmlHttp.readyState==4) {</pre>

```
if (xmlHttp.status==200)
{
xmlResponse=xmlHttp.responseXML;
xmlDocumentElement=xmlResponse.documentElement;
text=xmlDocumentElement.firstChild.data;
text=text.replace(/\[/g,"<");
text=text.replace(\/\]/g,">");
document.getElementById("show").innerHTML=text;
setTimeout('process()',1000);
}
else {alert("нет обращения к серверу")}
}
}
```

Для проверки запустите файл ajax.html (через протокол http), и в поле имя пользователя наберите localadmin

6. AJAX (с библиотекой mootools, скопируйте еще файл mootoolsm.js). Создайте следующие файлы:

ajax_m.html

```
<html>
<head>
<title>AJAX 2</title>
<script type="text/javascript" src="mootools.js"></script>
<script type="text/javascript" src="mootoolsm.js"></script>

<script type="text/javascript" src="ajax_m.js"></script>

</head>
<body>
```

```
<label>Введите имя пользователя</label>
```

```
<input type="text" id="name" />
```

```
<div id="show"></div>
```

```
</body>
```

```
</html>
```

php_ajax_m.php

```
<?
```

```
header('Content-Type:text/xml');
```

```
echo '<?xml version="1.0" encoding="windows-1251" standalone="yes"?>';
```

```
echo '<ответ>';
```

```
$name_m=iconv("UTF-8","WINDOWS-1251",urldecode($_POST['name_m']));
```

```
$privat=array("admin","localadmin","user","Вася");
```

```
if (in_array($name_m,$privat))
```

```
    echo 'Есть пользователь <strong style="color:#00F;">'.$name_m.'</strong>!';
```

```
    elseif(trim($name_m)=="")
```

```
        echo "<h1>Пользователь?</h1>";
```

```
        else echo '<strong style="color:#F00;">'.$name_m.'</strong> нет пользователя с таким  
именем!';
```

```
echo '</ответ>';
```

```
?>
```

ajax_m.js

```
window.addEvent('domready', function() {
```

```
var req = new Request.HTML({ method: 'post', url:'php_ajax_m.php',
```

```
    onSuccess: function(html) {
```

```
        $('show').set('text', "");
```

```
        $('show').adopt(html);
```

```
    },
```

```

        onFailure: function() {
            $('show').set('text', 'Не удалось обратиться к серверу');
        }
    });
    var name=$('name');
    name.addEvent('keyup', function(e){
        req.send('name_m=' + name.value);
    });
});

```

7. Создайте новую папку **gall**. Скопируйте в нее файлы **mootools**, **mootoolsm.js** и папки **img_sm** и **img**. В ней создайте следующие файлы:

index.html

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />
    <script type="text/javascript" src="mootools.js"></script>
    <script type="text/javascript" src="mootoolsm.js"></script>

    <script type="text/javascript" src="is.js"></script>
    <title>Прозрачность</title>
    <style type="text/css">
*{padding:0px;
margin:0px;}
.center{
    margin:10px auto;

```

```
}  
.bImage{  
    width:300px;  
    height:400px;  
}  
#Image{  
    float:left;  
    width:300px;  
    height:400px;  
}  
#Image1{  
    position:absolute;  
    width:300px;  
    height:350px;  
    overflow:hidden;  
}  
</style>  
</head>  
<body>  
<div id="smImages">  
      
      
      
      
      
</div>  
<div class="center bImage">  
<div id="Image1"> 
```

```
</div>
<div id="Image"> 

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

img.php

```
<?
    echo "<img src=\"".$_POST['id']."\" id=\"Img\"/><br />";
    echo "<h3>".$_POST['id']."</h3>";

?>
```

ajax_m.js

```
window.addEventListener('domready', function() {
var req = new Request.HTML({ method: 'post', url:'img.php',
    onSuccess: function(html) {
        $('Image').set('text', "");
        $('Image').adopt(html);
    },
    onFailure: function() {
        $('Image').set('text', 'Не удалось обратиться к серверу');
    }
});

var el= $('Image');
var el2= $('Image1');
var im= $$('.im');
```

```
var opaEffect = new Fx.Morph('Image1', {duration: 2000});

for(i=1; i<im.length; i++) im[i].setStyle('opacity', 0.5);

    $$('.im').addEvent('click', function(e){

        e.stop;

        opaEffect.start({ opacity:[1, 0], width:[300, 0], height:[350, 0]});

        src=this.src.replace('img_sm/', 'img/');

        req.send('id=' + src);

        for(i=0; i<im.length; i++) {if (im!=this) im[i].fade(0.5);}

        this.fade(1);

        var img=$( 'Img' );

        el2.innerHTML='';

    }

);

});
```

Проверьте работу приложений, исправьте ошибки.

Содержание отчета

Отчет по лабораторной работе должен содержать краткий конспект по теоретической части и выполненные в ходе работы файлы.

13 Создание сайта средствами CMS

Цель работы – ознакомление с принципами создания сайта с помощью CMS.

Задачи работы: получение навыков создания сайта с помощью CMS; получение навыков работы с шаблонами; получение навыков создания страниц с использованием шаблонов.

Теоретические сведения

Сайт – это набор взаимосвязанных файлов, составляющих завершенный web-ресурс. Средства MS SharePoint позволяют работать как с локальным, размещенным на локальном компьютере, так и с удаленным web-сайтом, размещенным в сети. Локальный сайт используют для создания, тестирования и отладки веб-сайта. С помощью локального узла проверяется корректность работы новых или измененных страниц, которые затем размещают на удаленном сайте. Чтобы на локальном сайте (узел) производить отладку, он должен быть максимально копией удаленного веб-сайта (зеркалом удаленного веб-сайта), т.е. иметь одинаковые настройки сервера (если для тестирования применяется локальный сервер), операционной системы (если возможно), иметь одинаковую структуру каталогов.

Оборудование, программное обеспечение

Оборудование – ПК с установленной графической операционной системой (Windows или Linux).

1. Программное обеспечение – Denwer, свободные CMS (Joomla!, Drupal, WordPress), графический Web-браузер, например: Internet Explorer, Mozilla Firefox, Safari, Google Chrome или Opera.

Задание на работу

Создать узел сайта используя шаблоны встроенные в визуальный редактор.

Ход работы

1 Установка

Как известно, чтобы создать сайт - не нужно иметь хостинг в Интернете, для этого нужно скачать локальный веб-сервер, который будет имитировать работу хостинга. Если у вас, вдруг еще нету локального веб-сервера, то прошу изучить эту статью: <http://joomla-create.ru/index.php/ustanovka-denwer>. В этом уроке мне хотелось бы продемонстрировать процесс установки новой CMS Joomla 3.x на наш веб-сервер.

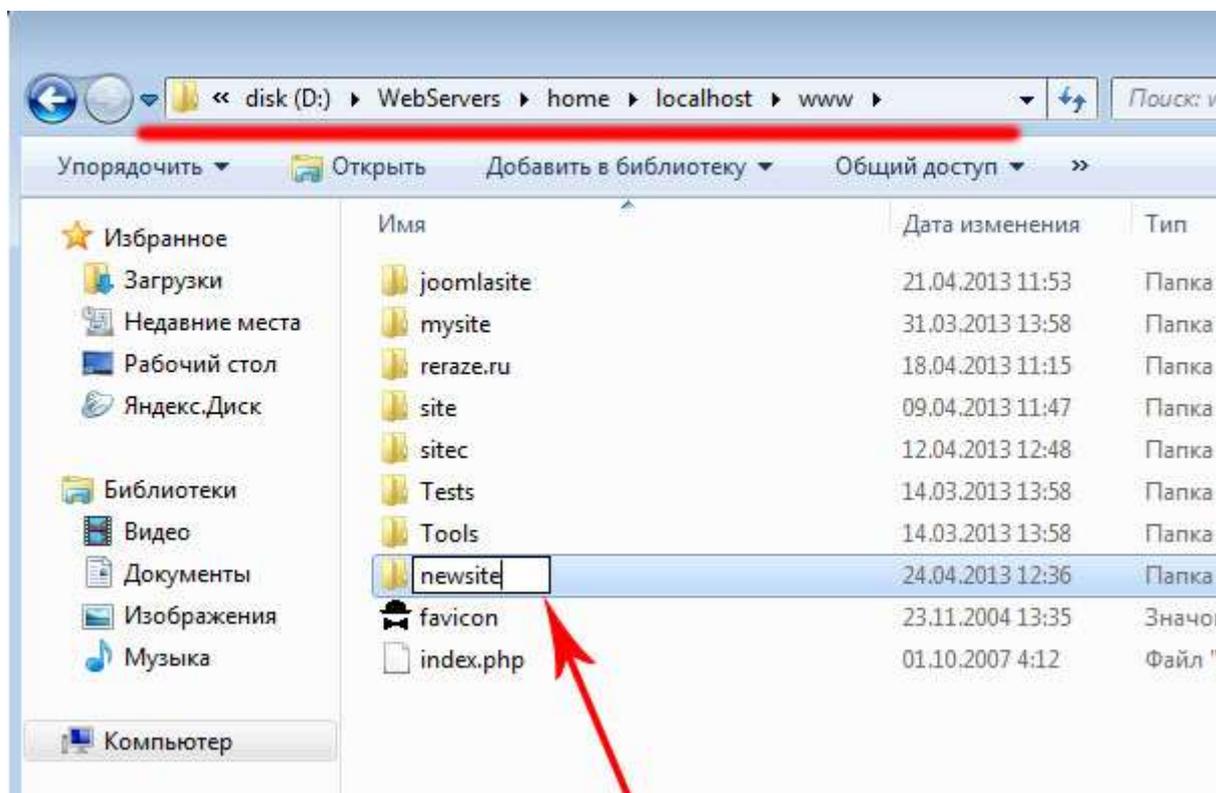
1. СКАЧИВАНИЕ JOOMLA 3.X

Скачиваем последнюю стабильную версию Joomla с официального сайта: <http://www.joomla.org/>. С официального, потому, что там безопаснее и к тому же, теперь не нужно искать русскую версию Joomla.

Скаченный файл представляет собой архив, названия Joomla 3.x-Stable Full, в зависимости от версии.

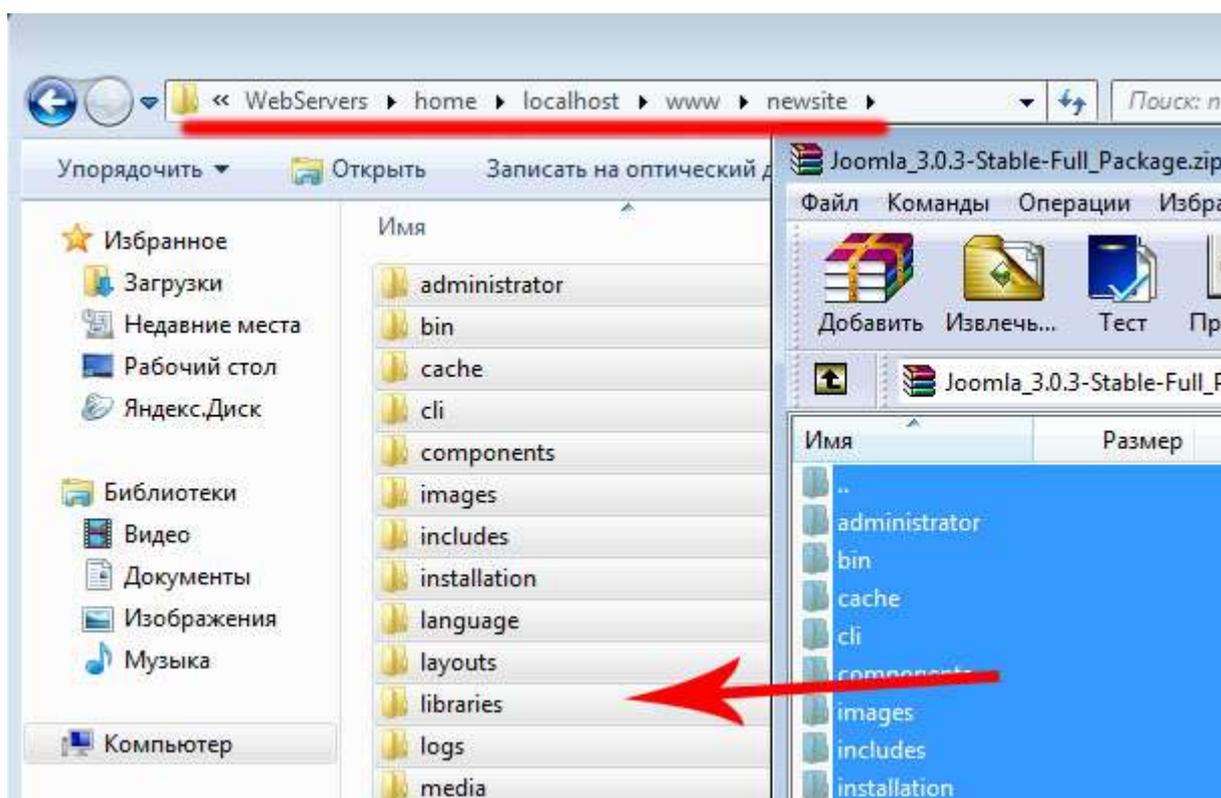
2. СОЗДАНИЕ НОВОЙ ПАПКИ

Перед распаковкой файлов Joomla, необходимо подготовить папку. Если вы установили локальный веб-сервер Denwer, то папку нужно располагать в WebServers/home/localhost/www/newsite, newsite - произвольное название сайта. Если вашим веб-сервером является Хампрр, то папка должна быть создана вxampp/htdocs/newsite.



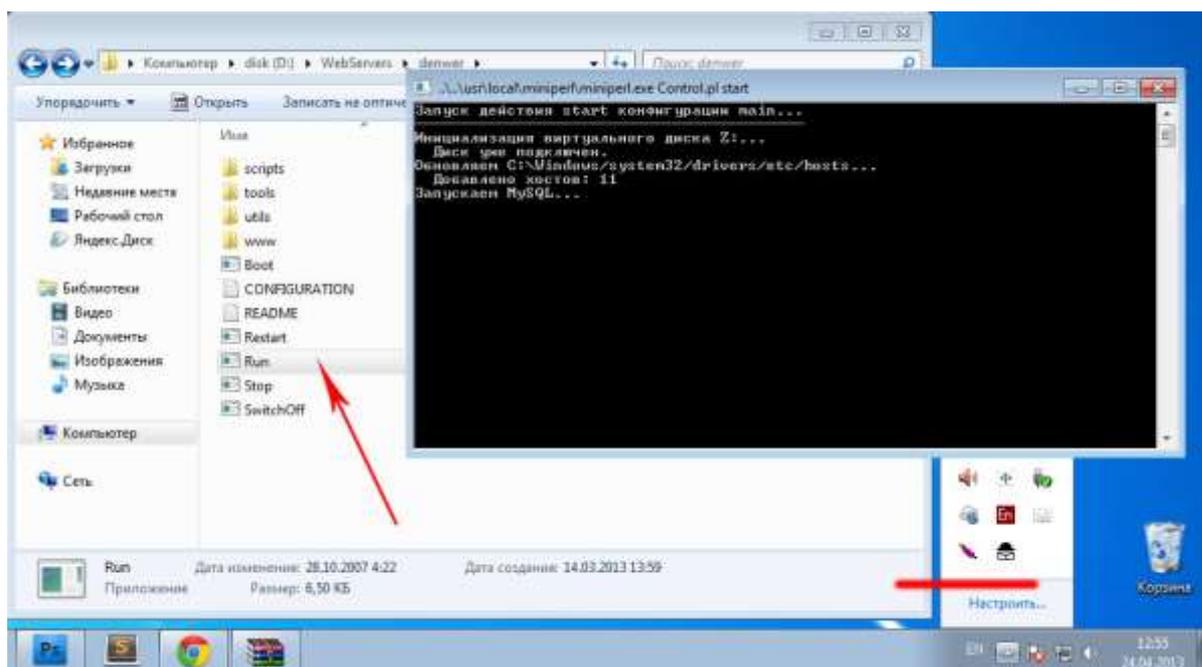
3. РАСПАКОВКА ФАЙЛОВ JOOMLA

Содержимое архива Joomla, необходимо распаковать в созданную вами папку, у нас это newsite.



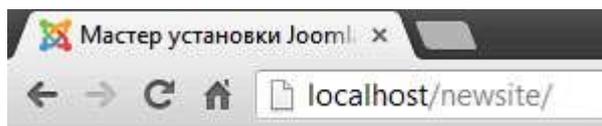
4. ЗАПУСК ЛОКАЛЬНОГО ВЕБ-СЕРВЕРА

Joomla, готова к установке, и чтобы это сделать нужно произвести запуск локального веб-сервера. Если у вас веб-сервер Denwer, то запуск осуществляется через ярлык Start denwer/Run. Если ярлыка нет, то запускайте напрямую через: WebServers/denwer/Run.exe. У веб-сервера Хамрр ярлык расположен в корневой папке и называется - хамрр_start.exe



5. ЗАПУСК УСТАНОВКИ JOOMLA

Работать с сайтом мы будем исключительно через браузер, и установку производить будем тоже с помощью браузера. Поэтому запускаем любой браузер и в адресной строке вводим: localhost/newsite



6. УСТАНОВКА JOOMLA 3.X

Перед нами открывается страница установки Joomla. Установка представляет собой несколько шагов, в каждом из которых нужно будет заполнить некоторую информацию.

6.1. УСТАНОВКА JOOMLA - ШАГ 1

На первом шаге установки мы задаем:

Название сайта - будет отображаться в заголовках страницы и в шапке

Описание сайта - небольшой текст описания сайта в meta - тегах description для всех страниц, где нет подобного описания

E-mail - ваш рабочий почтовый ящик, куда будут отправлять все сообщения и уведомления

Логин - данные для входа в админ. панель

Пароль - пароль для входа в админ. панель (рекомендую ставить admin)



Joomla!® распространяется по лицензии GNU/GPL.

1 Конфигурация сайта 2 Конфигурация БД 3 Обзор

Выберите язык Russian Русский x

7.
→ Далее

Конфигурация сайта

Название сайта * Мой сайт 1.

Введите название вашего сайта.

Описание Мой первый сайт на Joomla 2.

Введите описание вашего сайта для Поисковых систем. Оптимальная длина описания - 20 слов.

Выключить сайт Нет Да

Е-mail администратора * mymail@gmail.com 3.

Введите e-mail адрес для учетной записи Суперадминистратора сайта.

Логин администратора * admin 4.

Вы можете изменить логин Суперадминистратора сайта. По умолчанию - **admin**.

Пароль администратора * 5.

Введите пароль для учетной записи Суперадминистратора сайта и подтвердите введенный пароль в соответствующем поле ниже.

Подтверждение пароля * 6.

После ввода всех данных переходим на шаг 2.

6.2. УСТАНОВКА JOOMLA - ШАГ 2

На втором шаге необходимо заполнить всего два поля:

Имя пользователя - всегда root

Имя базы данных - название базы, где будут храниться все данные вашего сайта (рекомендуем ставить как название сайта: newsite)



Joomla!® распространяется по лицензии GNU/GPL.

1 Конфигурация сайта 2 Конфигурация БД 3 Обзор

3.

Конфигурация базы данных

← Назад → Далее

Тип базы данных * MySQLi

Это обычно "MySQLi"

Имя сервера базы данных * localhost

Это обычно "localhost"

Имя пользователя * root

1.

Введите имя пользователя базы данных, выданное хостером. На локальном сервере обычно используется учётная запись "root" без пароля.

Пароль

Введите пароль пользователя базы данных. Не рекомендуется применять учётную запись без пароля.

Имя базы данных * newsite

2.

Нажимаем далее и переходим на следующий этап.

6.3. УСТАНОВКА JOOMLA - ШАГ 3

На третьем шаге можно выбрать только одну опцию:

Установка демо - данных - позволяет сразу наполнять сайт содержимым. Если вы будете самостоятельно разбираться в возможностях Joomla, то имеет место установить Визитку (все содержимое будет на английском)



Joomla!® распространяется по лицензии GNU/GPL.

1 Конфигурация сайта 2 Конфигурация БД 3 Обзор

2.

Завершение установки

← Назад

→ Установка

Установка демо-
данных

- Нет
- Блог English (GB) демо-данные
- Визитка English (GB) демо-данные
- Стандартные English (GB) демо-данные
- Изучаем Joomla! English (GB) демо-данные
- Тестовые English (GB) демо-данные



Начинающим пользователям рекомендуется установить демо-данные. Они помогут легче освоиться с основными функциями системы.

Обзор

Отправить
конфигурацию сайта
на e-mail

Нет Да

Позволяет после завершения установки отправить параметры конфигурации на e-mail: mymail@gmail.com

Все остальное - это просто информация по данным вашего локального веб-сервера, которые вас не должны интересовать.

6.4. УСТАНОВКА JOOMLA - ШАГ 4

На четвертом шаге необходимо установить русский язык. Для этого необходимо перейти по кнопке: "Установка языковых пактов".



Joomla!® распространяется по лицензии GNU/GPL.

Поздравляем, вы установили Joomla!

ВНИМАНИЕ: НЕ ЗАБУДЬТЕ ПОЛНОСТЬЮ УДАЛИТЬ ДИРЕКТОРИЮ INSTALLATION.
Установка Joomla! не будет завершена, пока вы не удалите данную директорию. Это требование безопасности Joomla!.

Удалить директорию 'installation'

Сайт

Панель управления

Подробности учетной записи администратора:

E-mail mymail@gmail.com

Имя пользователя admin

Желаете переключить интерфейс Joomla! на ваш родной язык?

Прежде чем вы удалите директорию 'installation' вы можете установить дополнительные языковые пакеты. Для начала процесса установки дополнительных языковых пакетов нажмите кнопку «Установка языковых пакетов».

→ Установка языковых пакетов

На странице будет представлен список всех возможных языков Joomla, здесь нам нужно выбрать соответствующий и нажать кнопку далее.

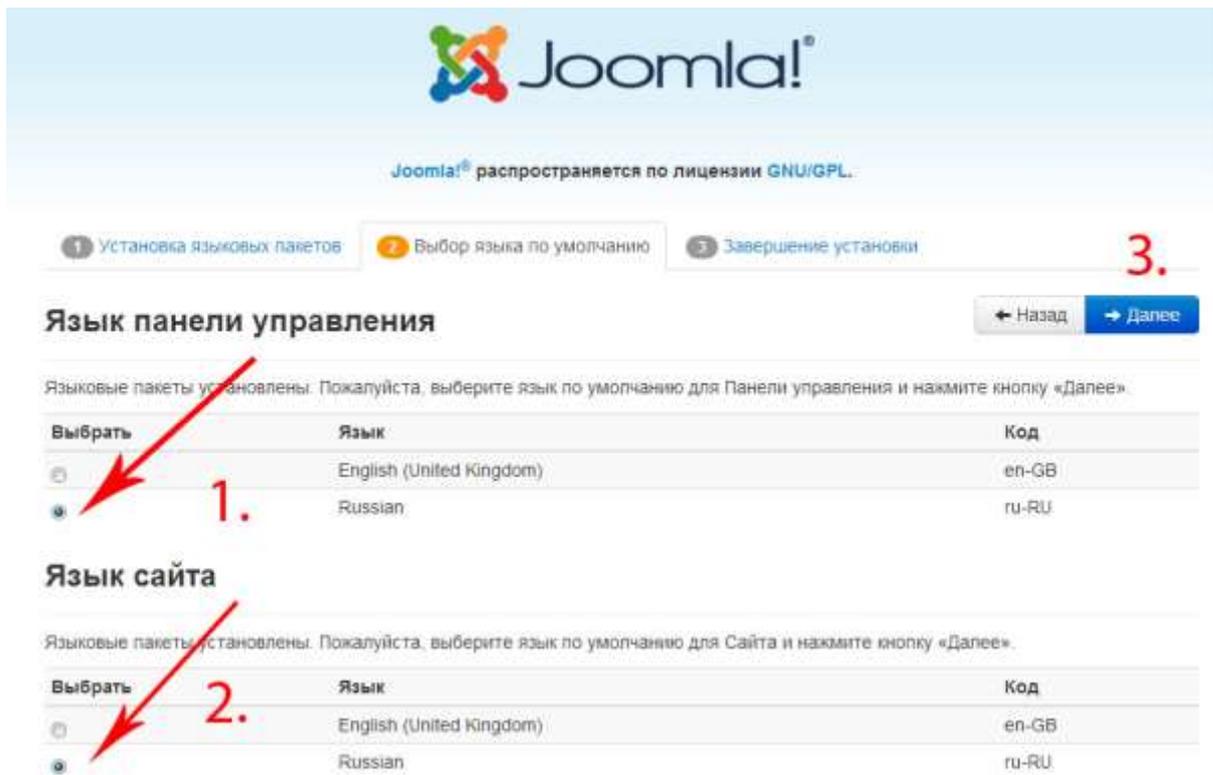
Portuguese Brazil

Russian

Serbian Cyrillic

Serbian Latin

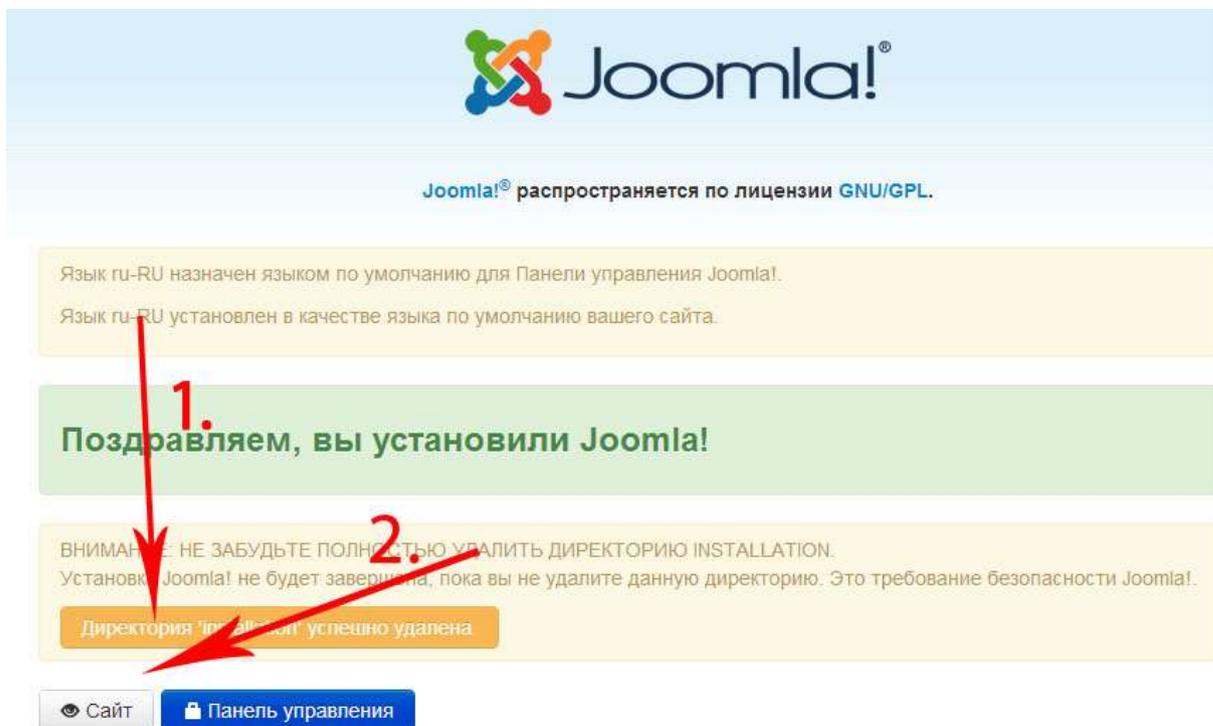
Затем, нужно выбрать русский язык для сайта и административной панели.



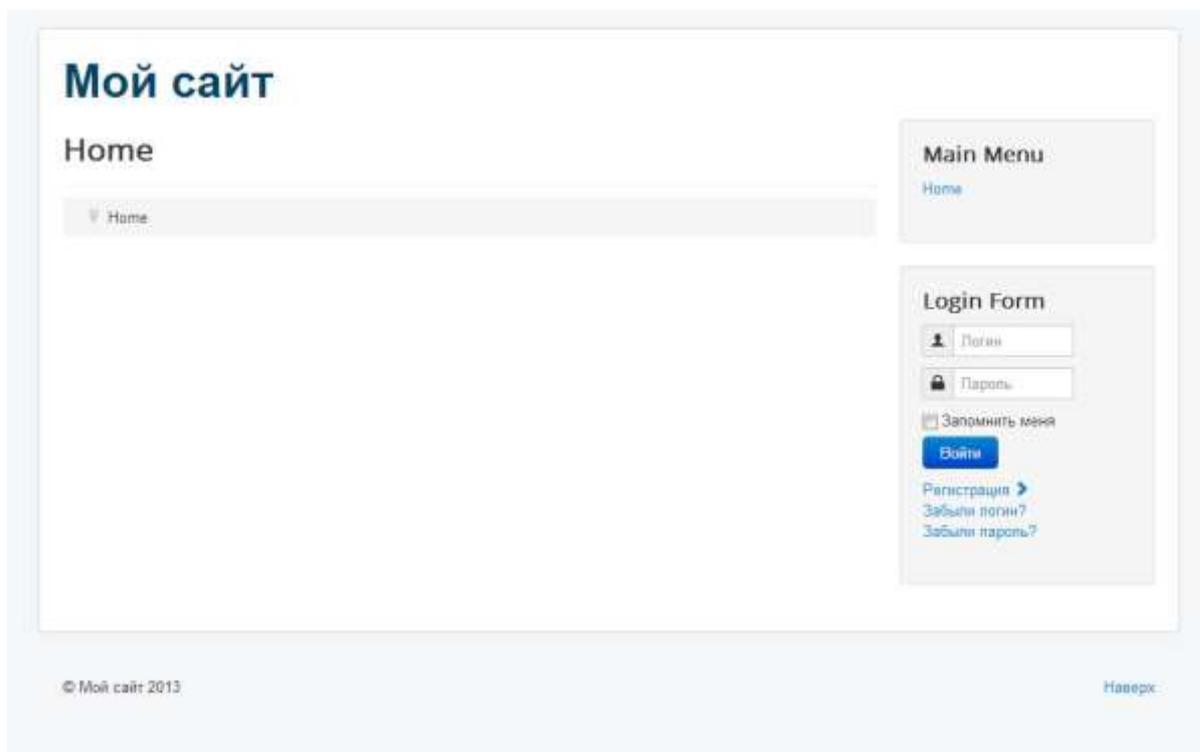
6.5. ЗАВЕРШЕНИЕ УСТАНОВКИ

Что завершить процесс установки сайта, необходимо нажать кнопку: "Удалить директорию Installation".

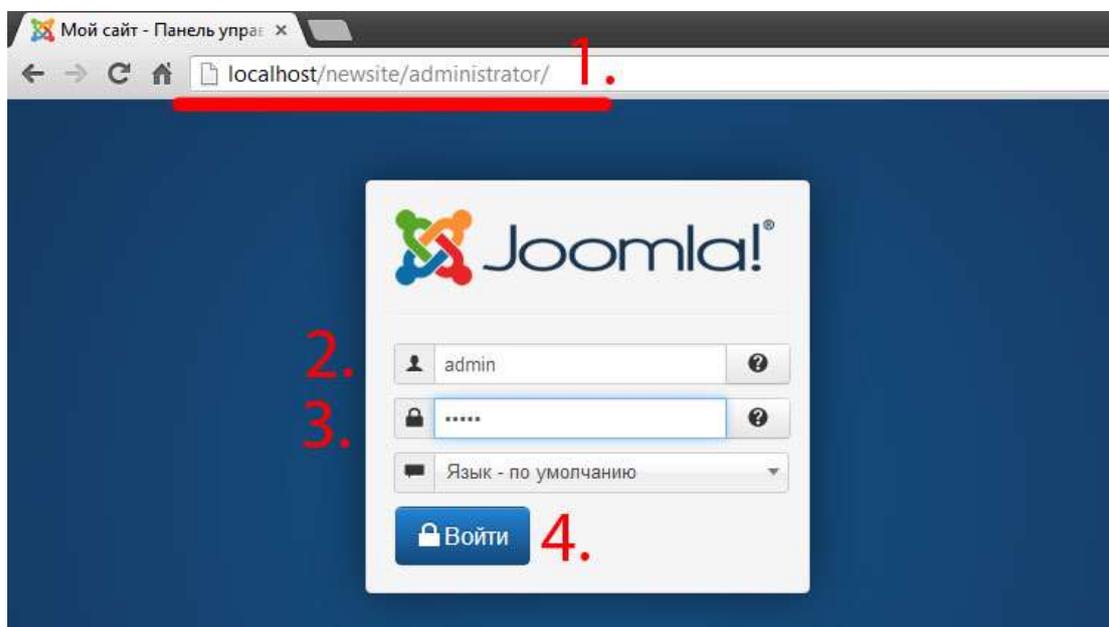
После этого можно просмотреть уже готовый сайт.



Если вы не устанавливали демо данные то сайт у вас будет пустой: только меню и форма входа.



Вы можете приступать к наполнению вашего сайта содержимым. Для этого нужно войти в административную часть. Чтобы в неё войти нужно прописать к названию вашего сайта слово administrator. Затем, ввести логин и пароль, указанные в шаге 1.



Содержание отчета

Отчет по лабораторной работе должен выполненные в ходе работы файлы.

14 Создание презентаций. Интерфейс программы. Основные элементы презентации

Цель работы – ознакомление с интерфейсом программы MS PowerPoint или Impress (LibreOffice) и основными элементами презентации.

Задачи работы – ознакомление с правилами создания презентации, получение навыков создания презентаций, навыков работы с основными элементами презентации.

Теоретические сведения

При создании презентации следует придерживаться ниже приведенных правил.

Этапы создания презентации:

1. Предварительный этап. Презентация является иллюстративным материалом к докладу (выступлению). На предварительном этапе составляется текст доклада.
2. Подготовительный этап. На этом этапе необходимо определить, какой материал должна содержать презентация, подготовить необходимые материалы (графики, рисунки фотографии) к докладу.
3. Этап 1. Под тематику выступления создается или выбирается шаблон оформления, компоновки, выбирается шрифтовая схема. Определяются с основными разделами выступления.
4. Этап 2. Заполнение слайдов информацией.
5. Этап 3. Настройка смены слайдов.
6. Отладка презентации проводится вместе с репетицией выступления.
7. Публикация презентации и печать раздаточного материала.

Оборудование, программное обеспечение

Оборудование – ПК с установленной графической операционной системой (Windows или Linux).

Программное обеспечение – MS PowerPoint 2007 или Impress (LibreOffice).

Задание на работу

1. Ознакомиться с интерфейсом программы MS PowerPoint или Impress (LibreOffice).
2. Создать презентацию, использующую стандартные шаблоны и темы.

15 Создание презентаций. Работа со слайдами

16 Создание презентаций. Создание шаблонов

Цель работы – ознакомление со средством создания презентаций MS PowerPoint или Impress (LibreOffice), средствами создания шаблонов презентаций.

Задачи работы – получение навыков создания шаблона презентаций, навыков работы с графическими объектами презентация, получение навыков создания фонов, текстовых и шрифтовых схем.

Теоретические сведения

Создание шаблонов в программе осуществляется путем команд «Образец слайдов». Образец слайдов позволяет не менять каждый слайд вручную, а путем создания или внесения изменений в используемый макет изменять все слайды, которые на его основе созданы. Шаблон используются для создания однотипных презентаций, так как позволяют хранить информацию, на основе которой можно единообразно форматировать слайды. Шаблон можно создать или править в любой момент создания презентации переключаясь в режим Вид / Образец слайдов. Макет можно добавить в шаблоны, сохранив как шаблон PowerPoint или Impress (LibreOffice).

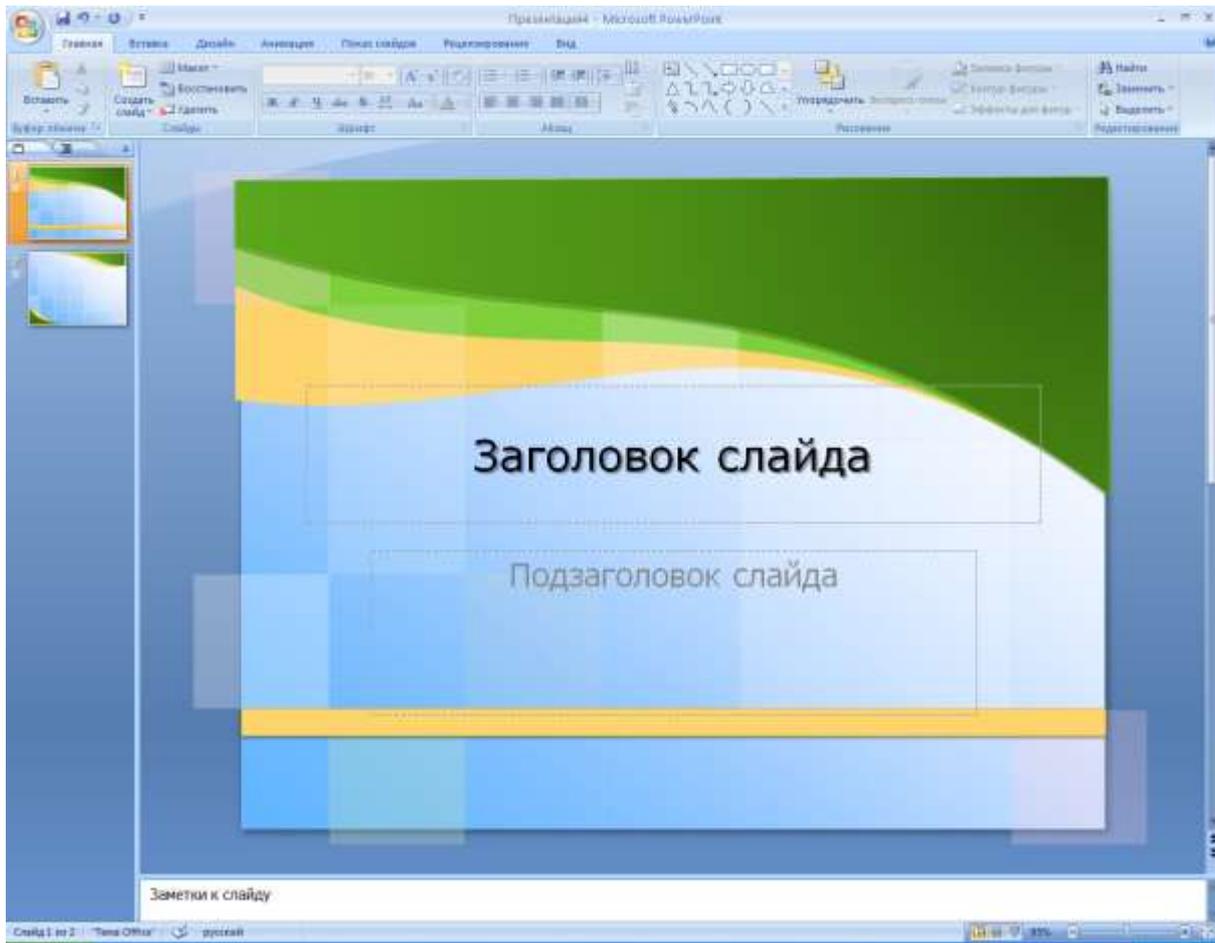
Оборудование, программное обеспечение

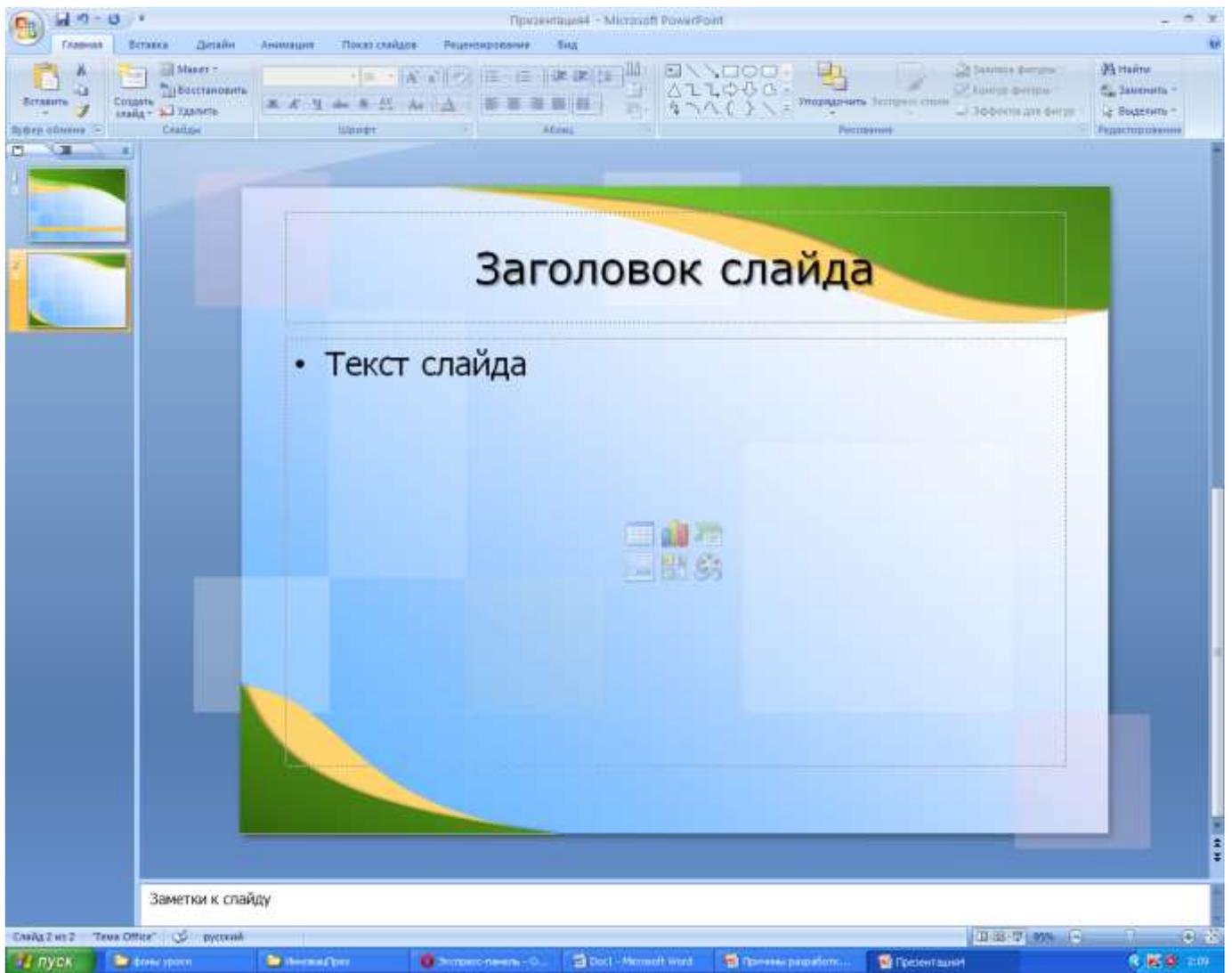
Оборудование – ПК с установленной графической операционной системой (Windows или Linux).

Программное обеспечение – MS PowerPoint или Impress (LibreOffice) 97-2003 или MS PowerPoint или Impress (LibreOffice) 2007.

Задание на работу

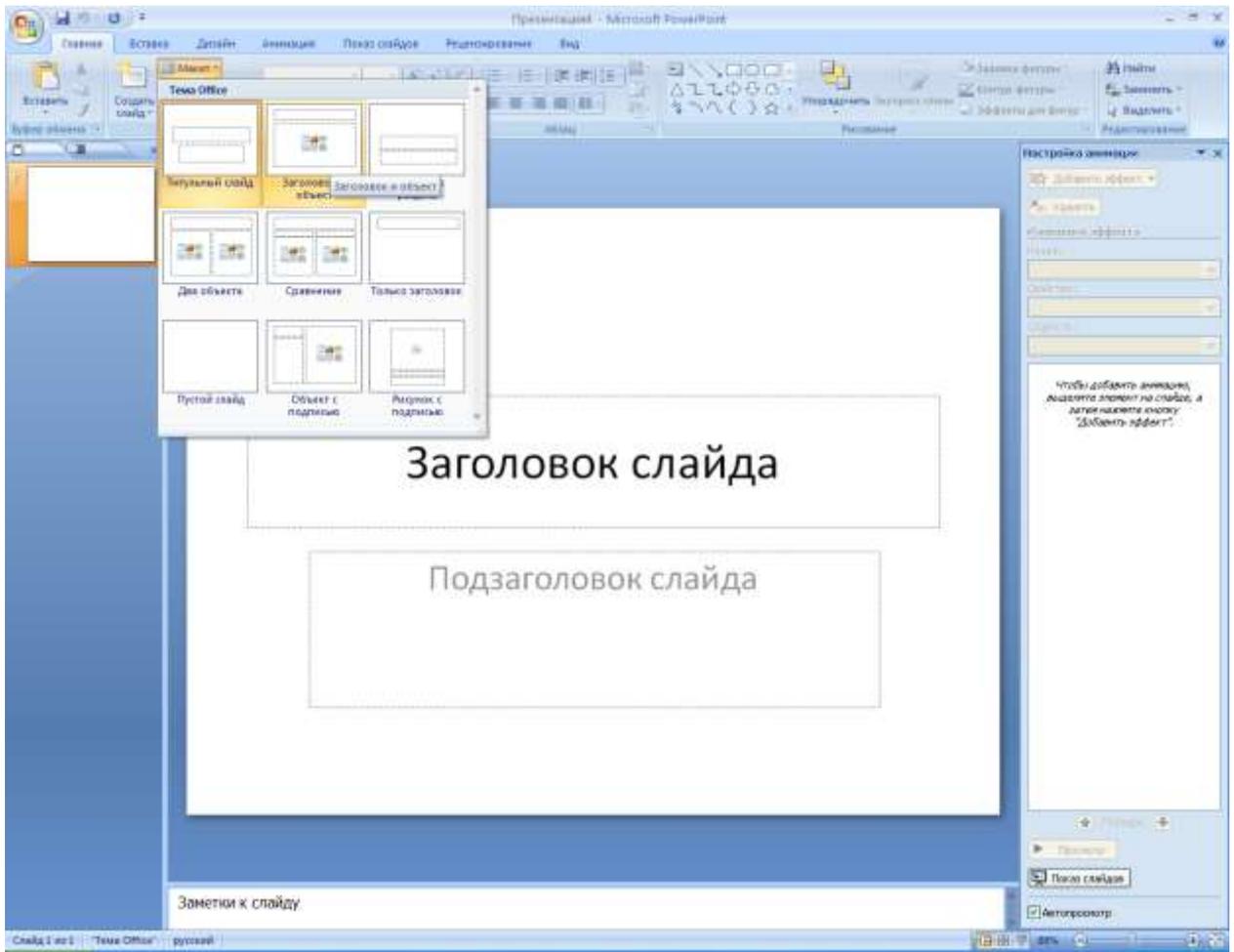
Создать шаблон титульного и обычного слайда к презентации см. рисунок 1, 2.



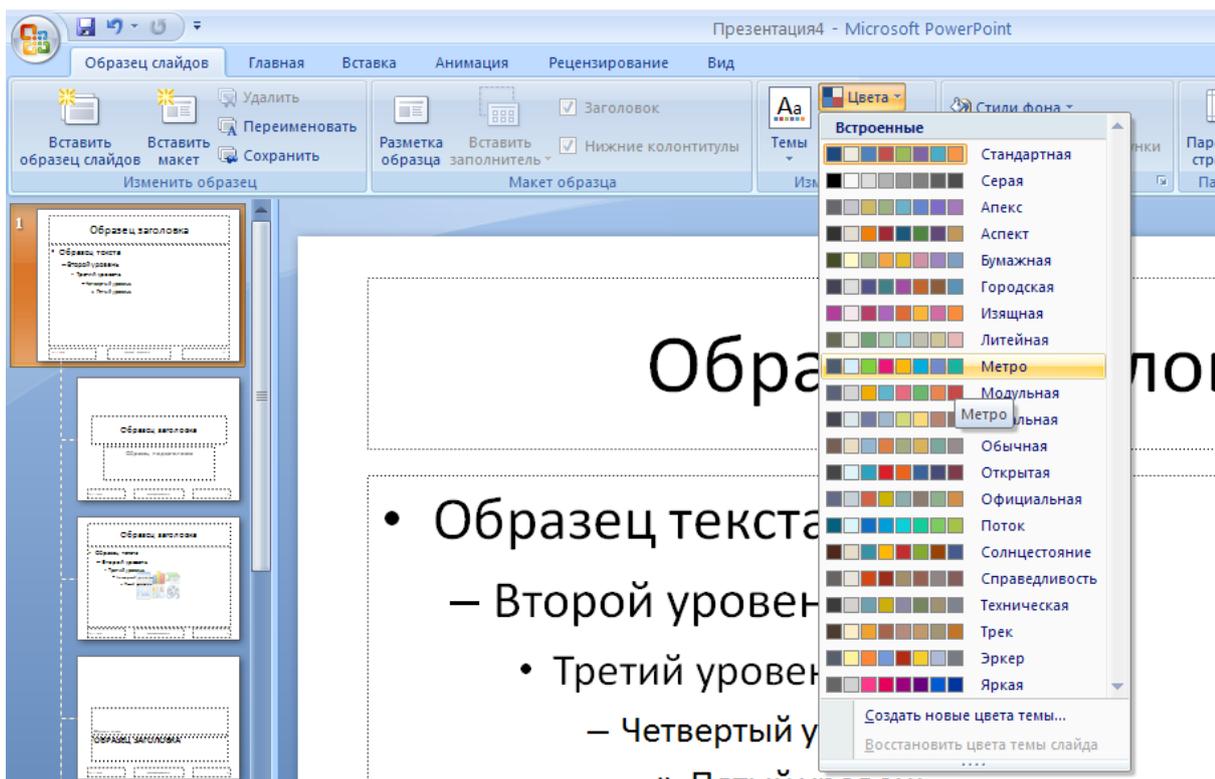


Ход работы

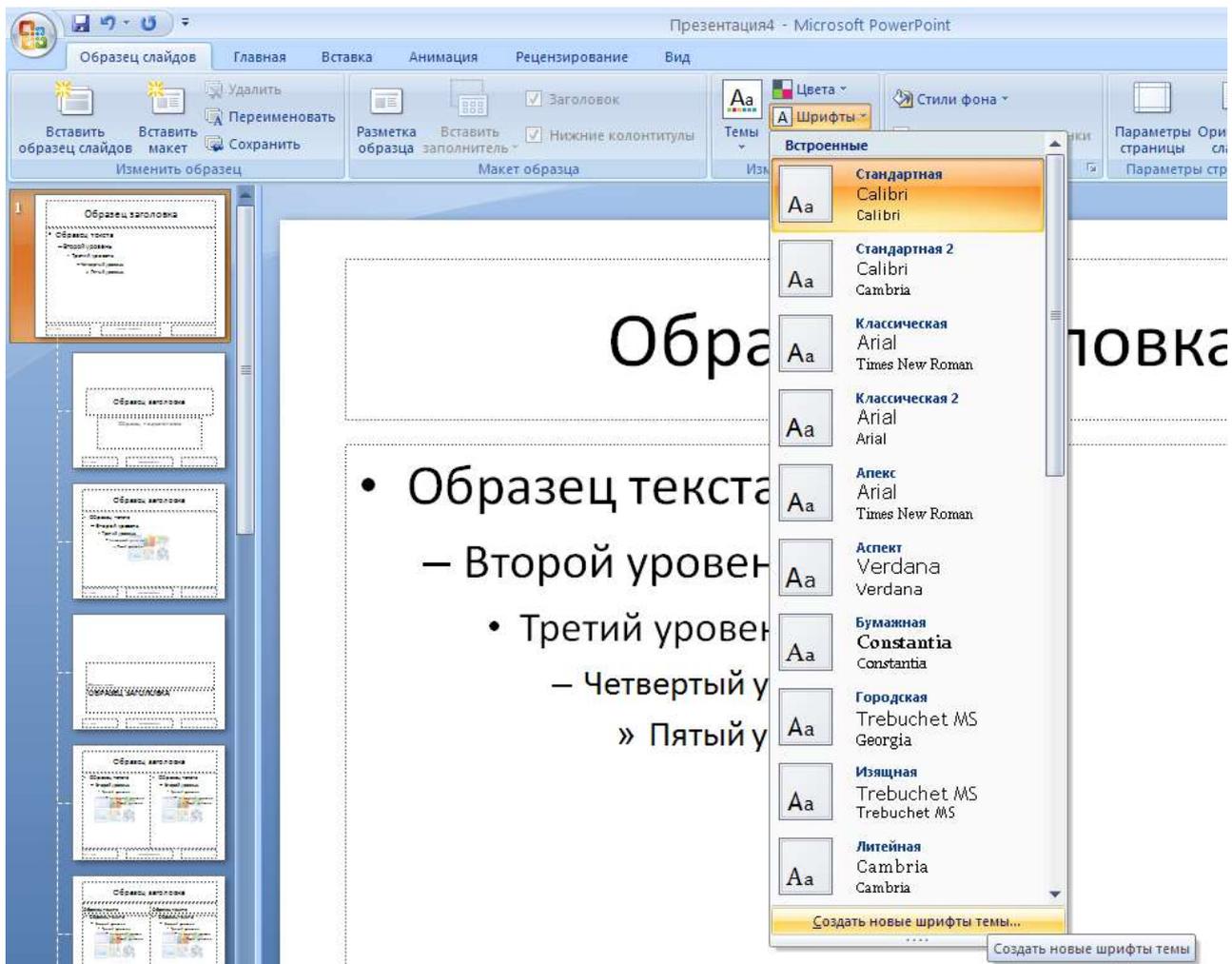
1. Создадим новую презентацию. Кроме титульного, создадим еще один слайд на основе макета «Заголовок и объект».



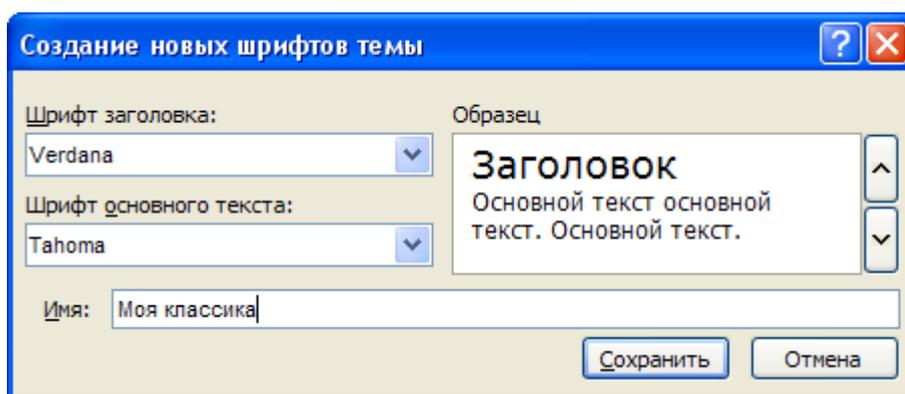
2. Во вкладке Вид выберем команду Образец слайдов. Выберите главный макет (самый верхний).
3. Во вкладке цвета выберите цветовую схему «метро».



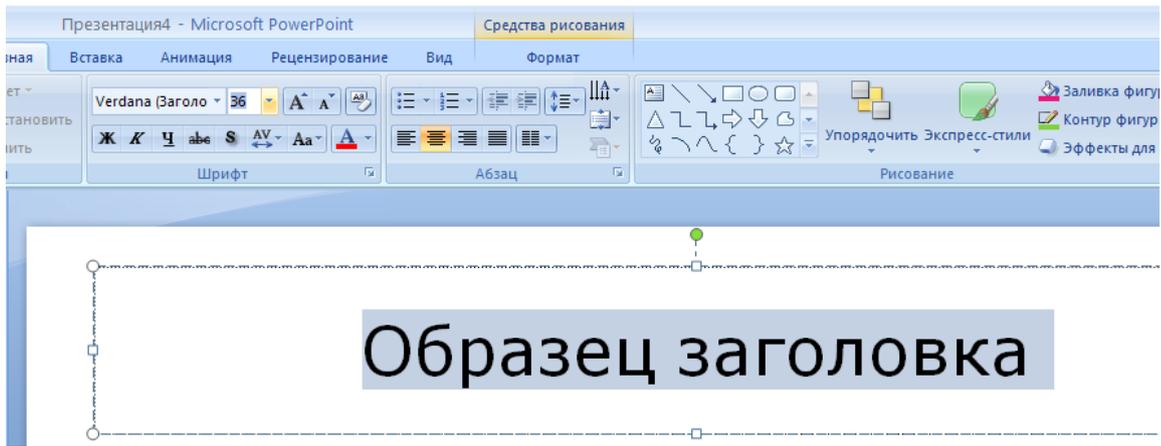
4. Создадим новую шрифтовую схему Шрифты/Создать новые шрифты темы.



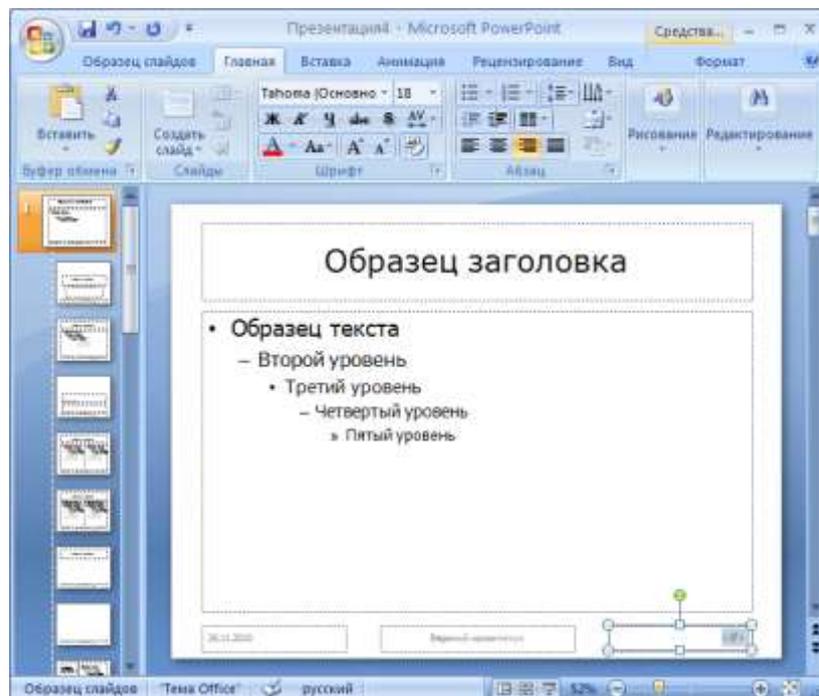
5. Задайте в качестве акцентного шрифта Verdana, а в качестве основного Tahoma. Дайте имя стилю.



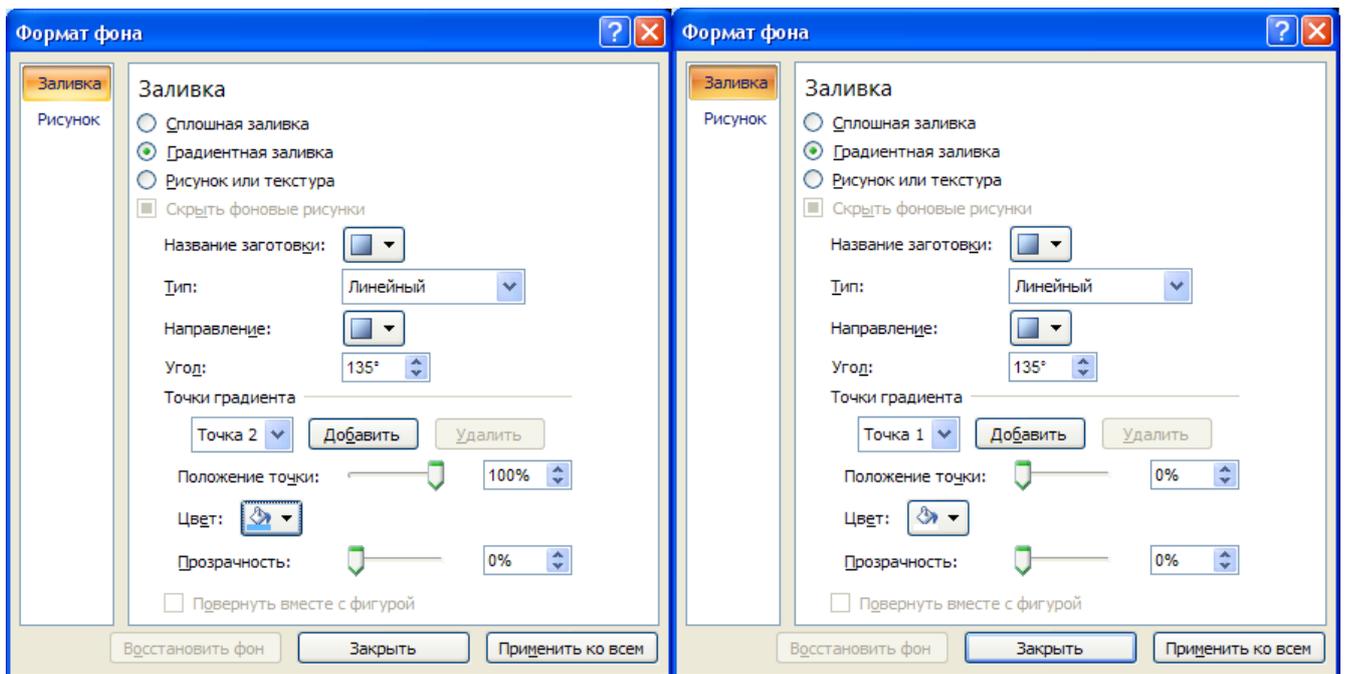
6. Выделите заголовок и во вкладке Главная задайте ему размер в 36пт. Для первого, второго, третьего, четвертого и пятого уровня основного шрифта слайда выберите 28, 24, 22, 20, 18 соответственно.



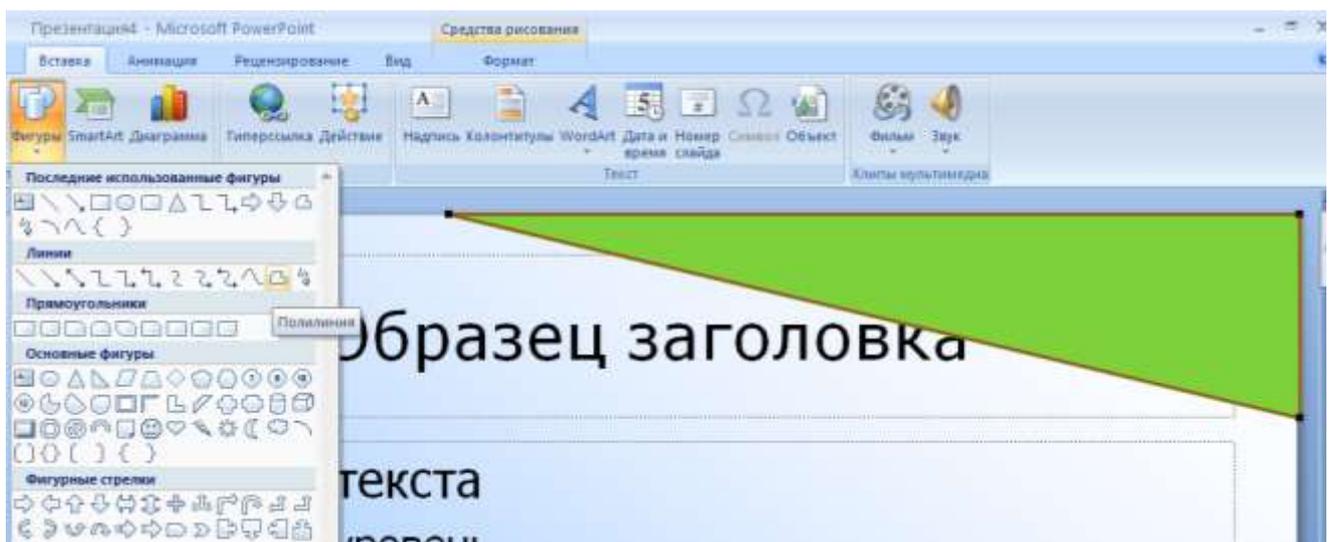
7. Для области вставки номера слайда <#> задайте шрифт размером 18pt.



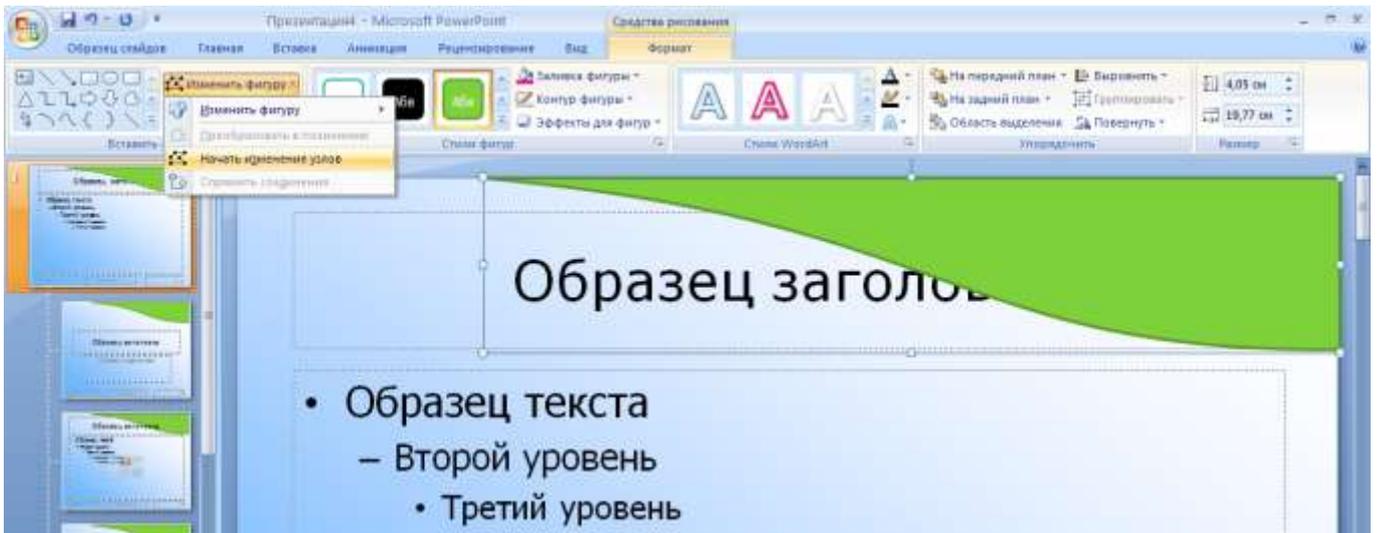
8. Перейдите во вкладку образец слайдов и задайте Стили фона/Формат фона заливку Градиентная заливка, тип Линейная, Направление -135, Точка 1 – белый цвет, Точка 2 – голубой.



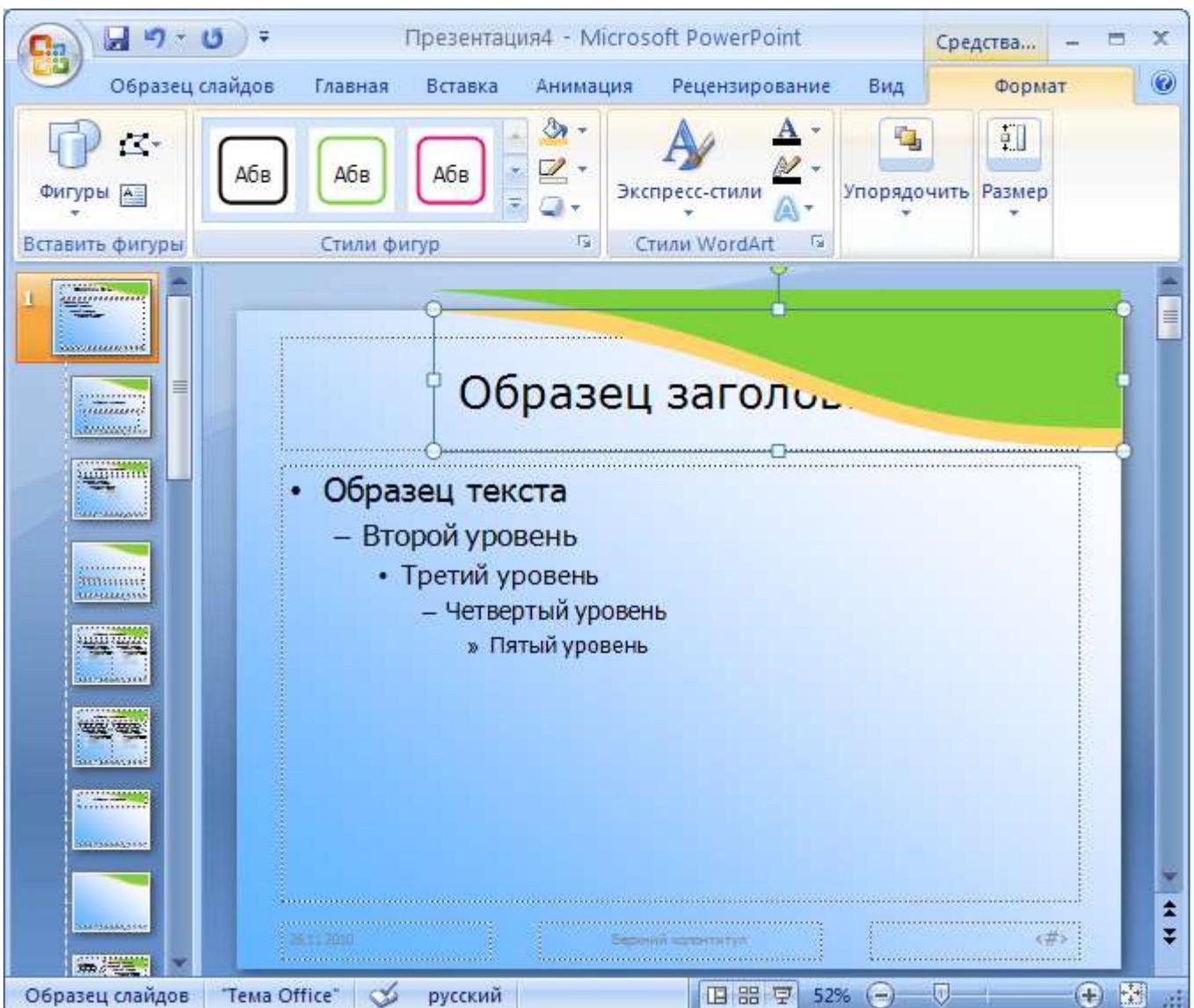
9. Переходим во вкладку Вставка, там выбираем Фигура / Линия / Полилиния и рисуем треугольник в правом верхнем углу (когда Вы щелкаете мышкой образуется узел, два щелчка – закончить полилинию, при нажатой мышке полилиния рисуется «от руки»).



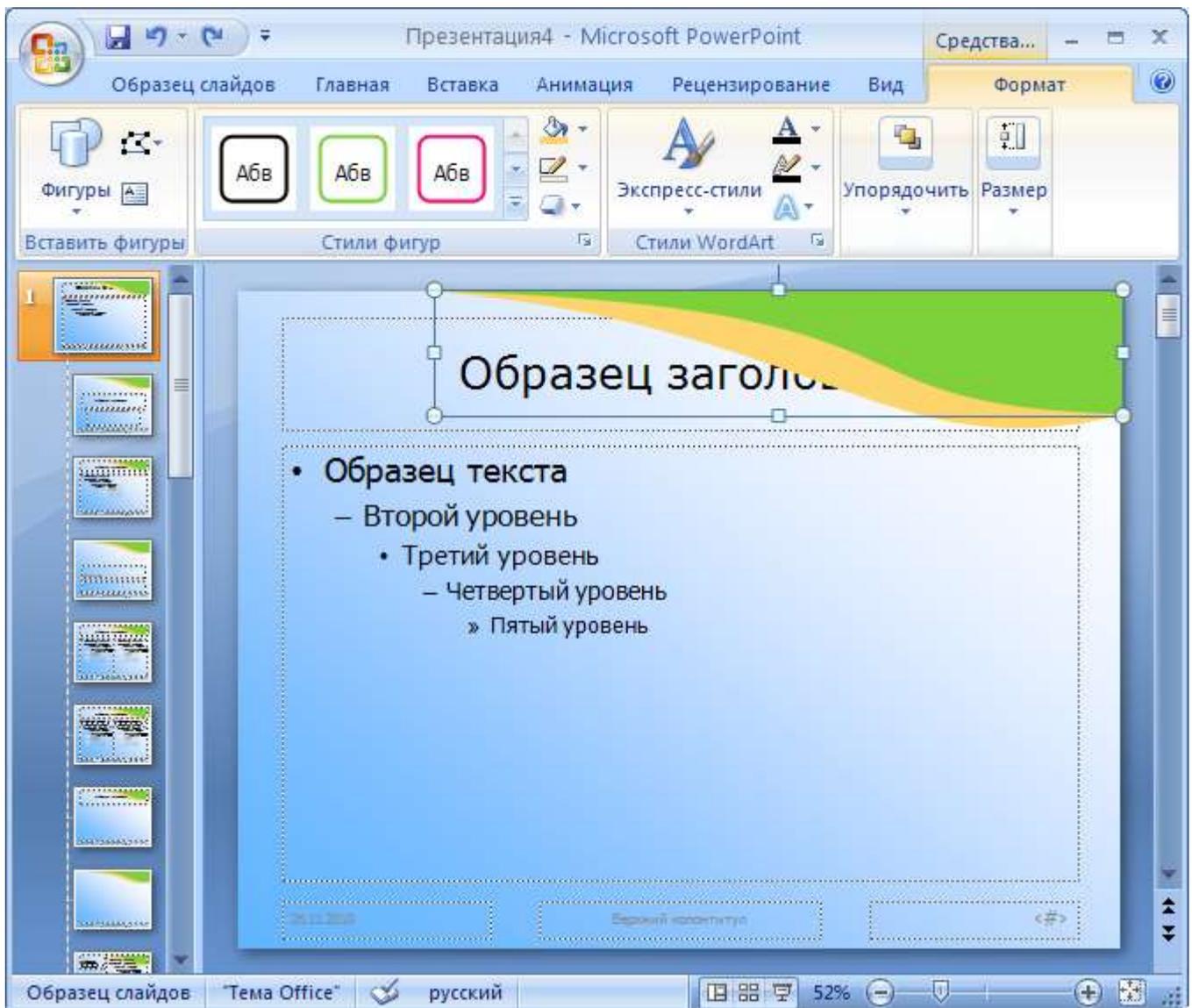
10. Далее выбираем мышью нарисованный треугольник и в появившемся меню Средства рисования / Формат выбираем Начать изменение узлов. Выделяя мышкой узел, можно менять его направляющие (можно и добавлять узлы щелчком мыши). Искривляем треугольник как на рисунке показанном ниже.



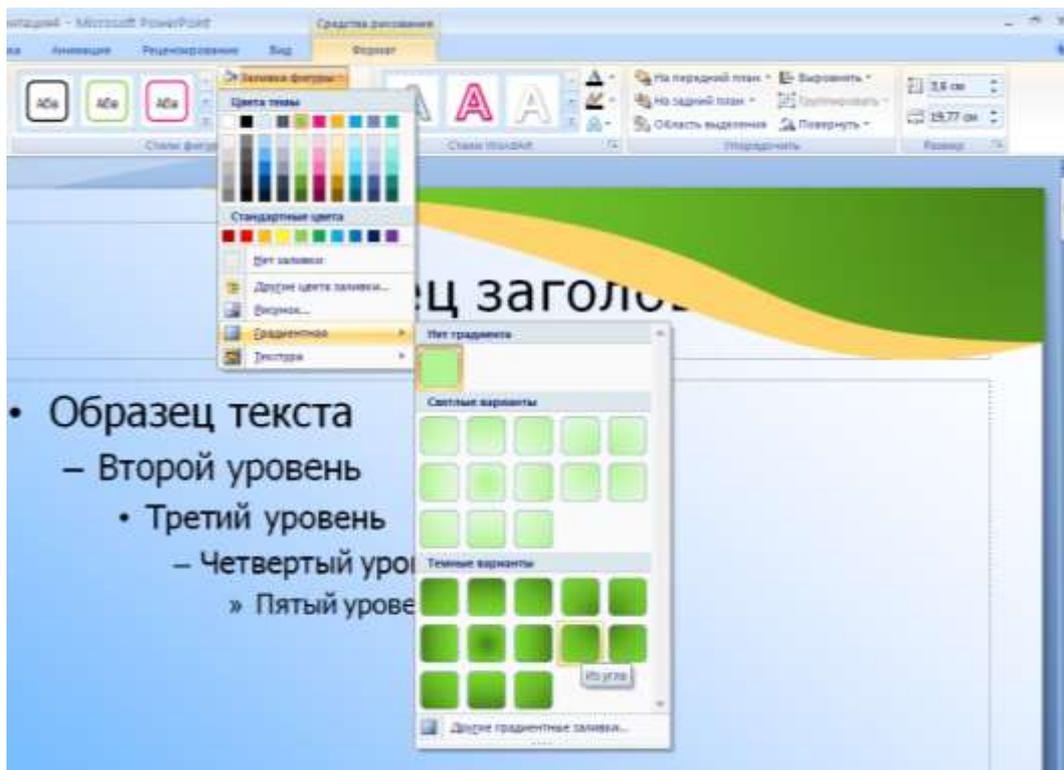
11. Убираем контур у фигуры (если он есть). Формат / Контур фигуры / Нет контура. Откопируем нарисованный «треугольник», можно путем Копировать (Ctrl+C) / Вставить (Ctrl+V), или сдвинув мышью зажав клавишу Ctrl. Задайте нижнему «треугольнику» персиковый цвет.



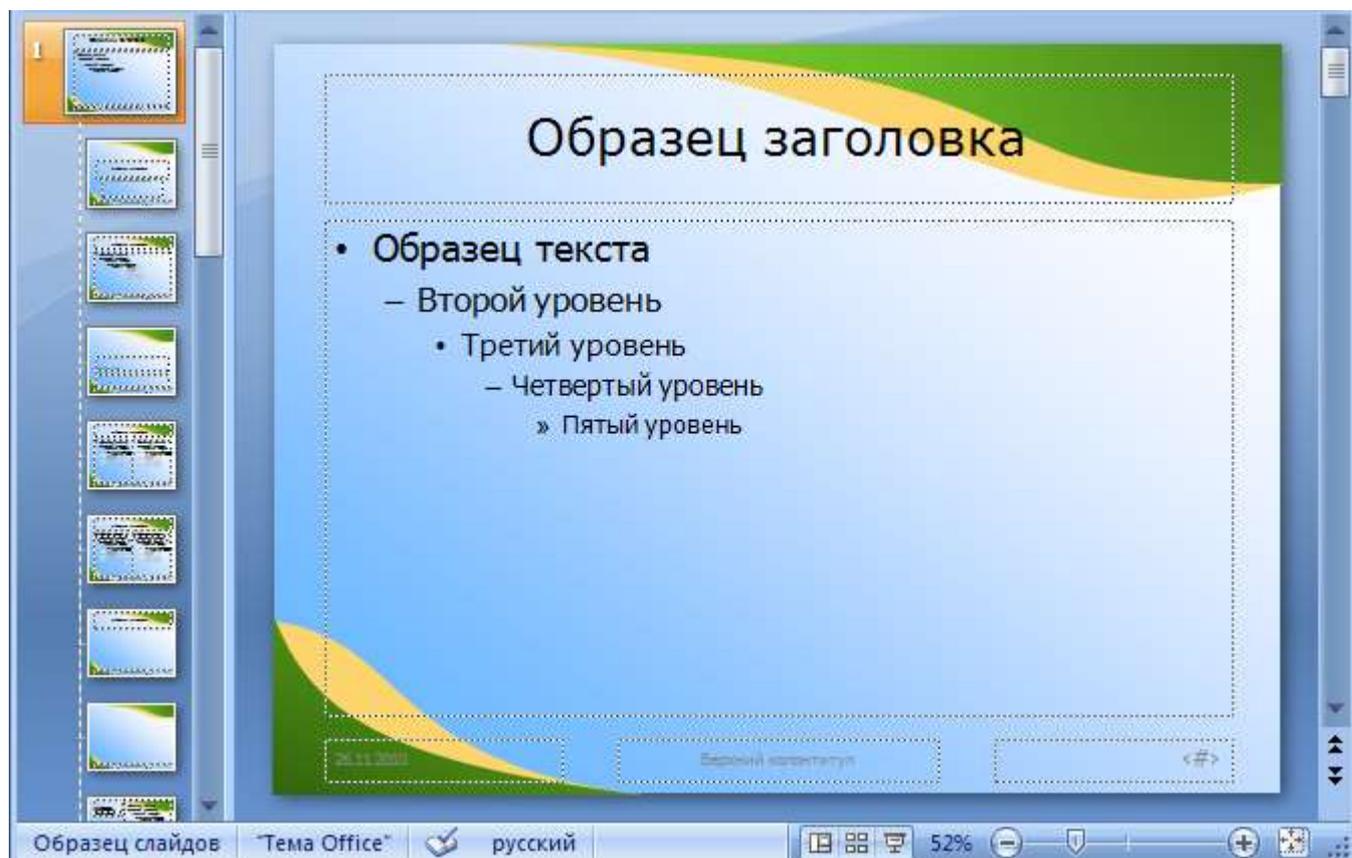
12. Верхний «треугольник» переместите и трансформируйте путем изменения узлов как на рисунке ниже.



13. Задайте верхнему «треугольнику» Заливку фигуры / Градиентная / Из угла.

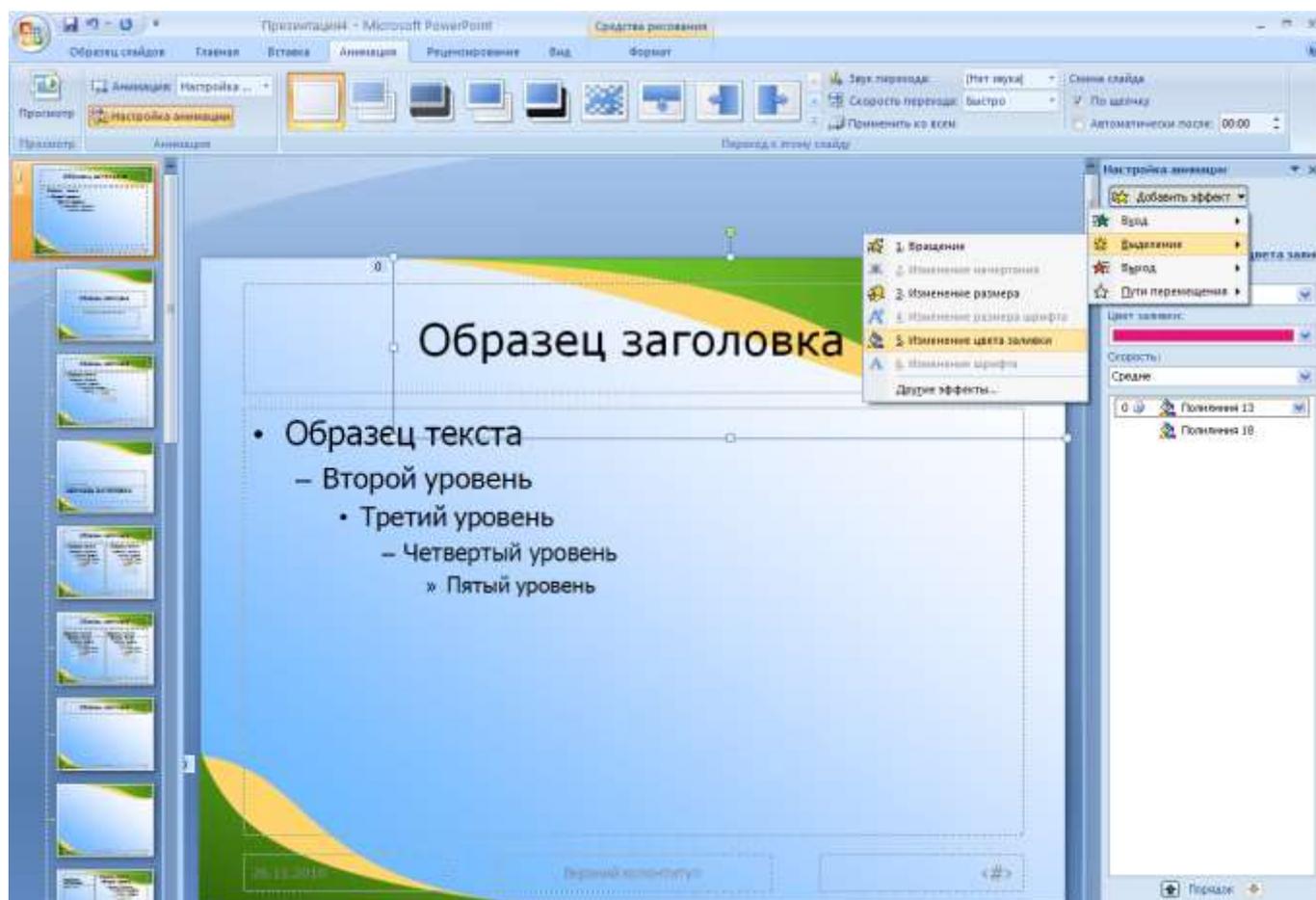


14. Отправьте оба «треугольника» На задний план (команда на панели Формат или в контекстном меню). Таким же способом создайте два «треугольника» в левом нижнем углу.



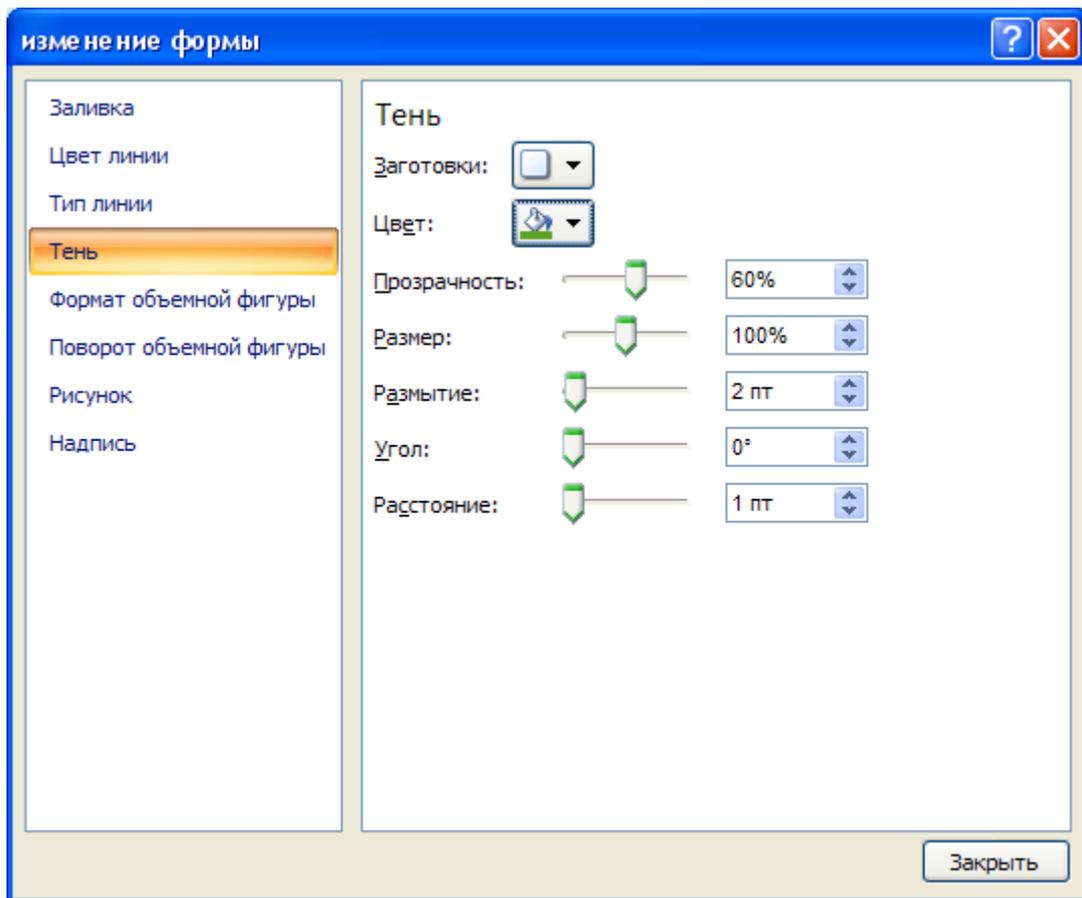
15. Во вкладке Анимация щелкните на команду Настройка анимации и для двух «треугольников» персикового цвета, верхнего и нижнего, задайте Добавит эффект / Выделение / Изменение цвета

заливки. Задайте для первого (любого) Начало / После предыдущего, а для второго С предыдущим (можно поменять Цвет заливки).

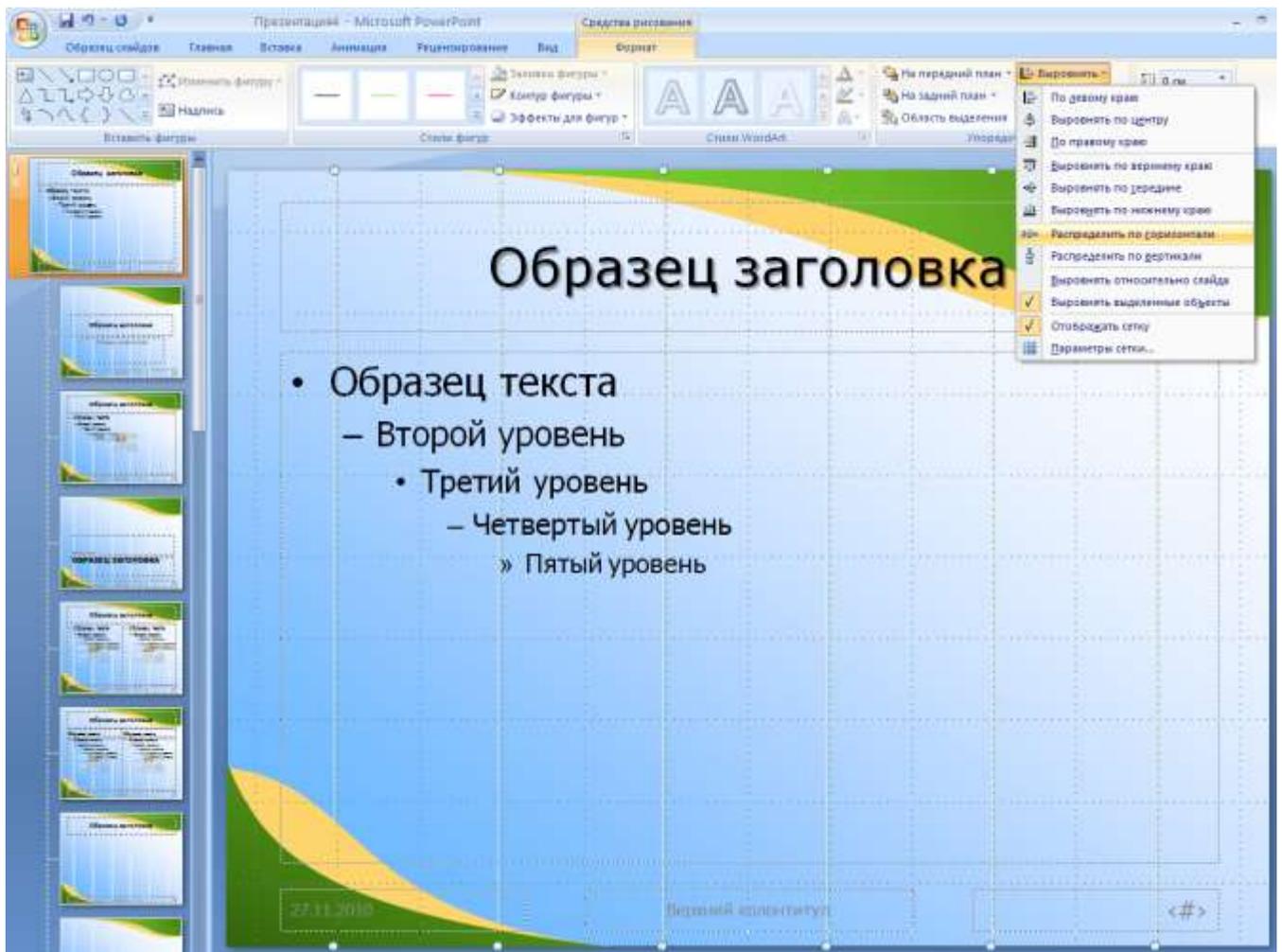


16. Клавишами Shift+F9 включите отображение сетки, или на панели Вид поставить галочку Сетка.

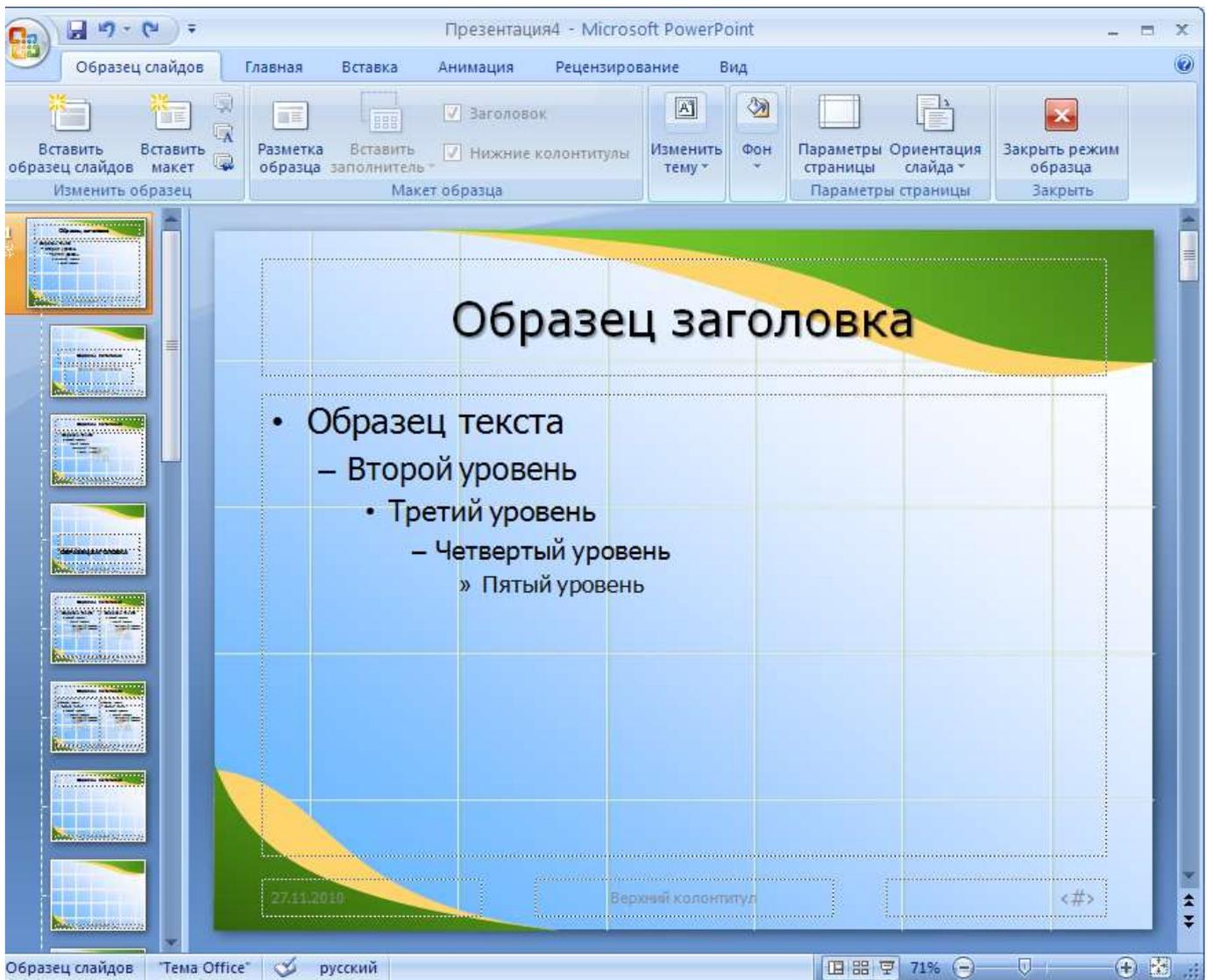
17. Вставкой Фигуры / Линия нарисуйте вертикальную прямую (зажимайте Shift). Во вкладке Средства рисования / Формат / Контур фигуры задайте ей: Цвет – белый, Толщина – 1 pt. Задайте ей Эффекты для фигур / Тень / С права со смещением, а в Вариантах тени задайте Размытие поставьте 2 pt, а Расстояние – 1 pt, Цвет - зеленый.



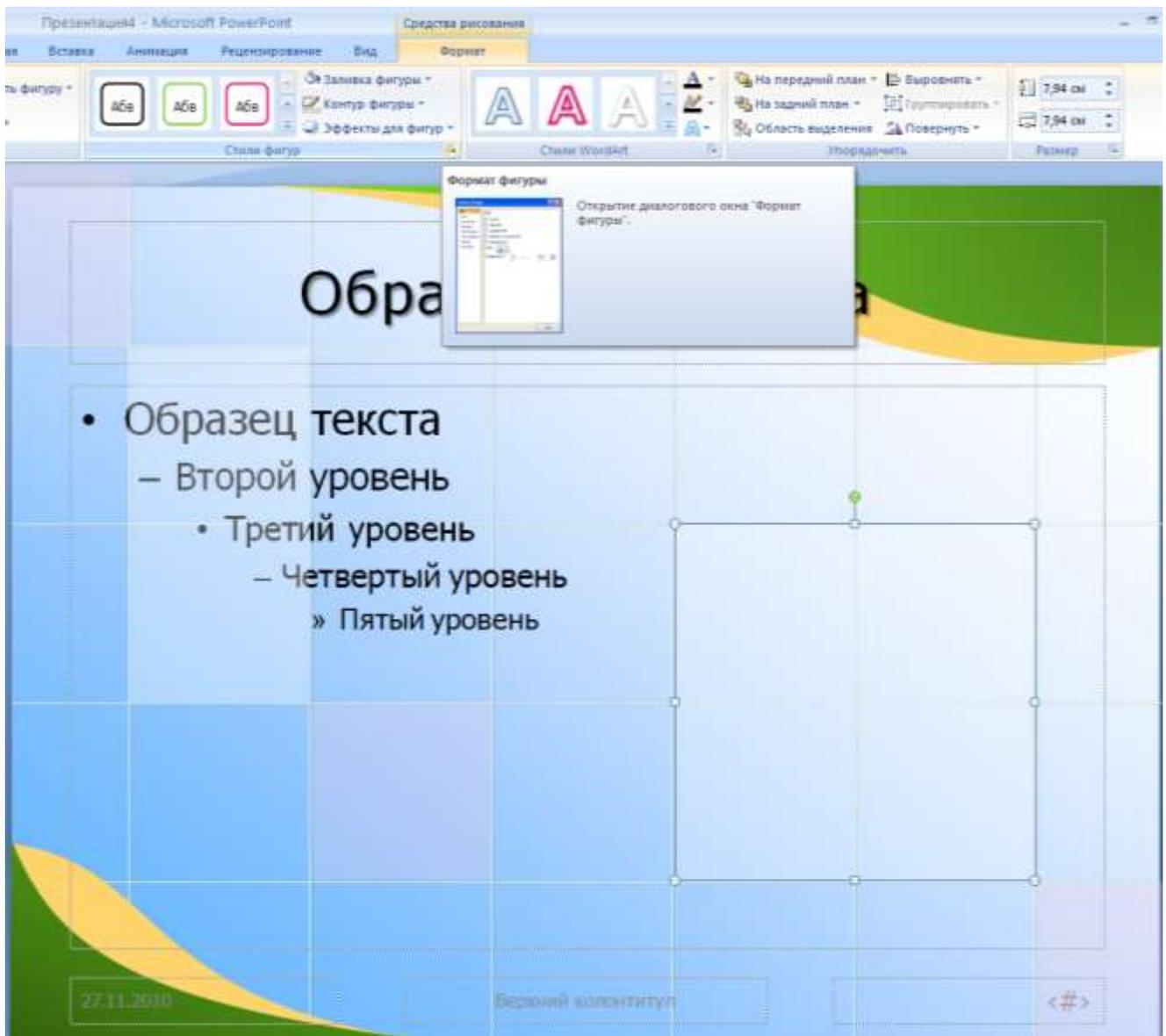
16. Откопируйте линию по сетке через одну клетку. Если получилось не через равные расстояния, то выделите мышкой, зажав Shift, все линии, а затем на панели Формат задайте Выровнять / Распределить по горизонтали.



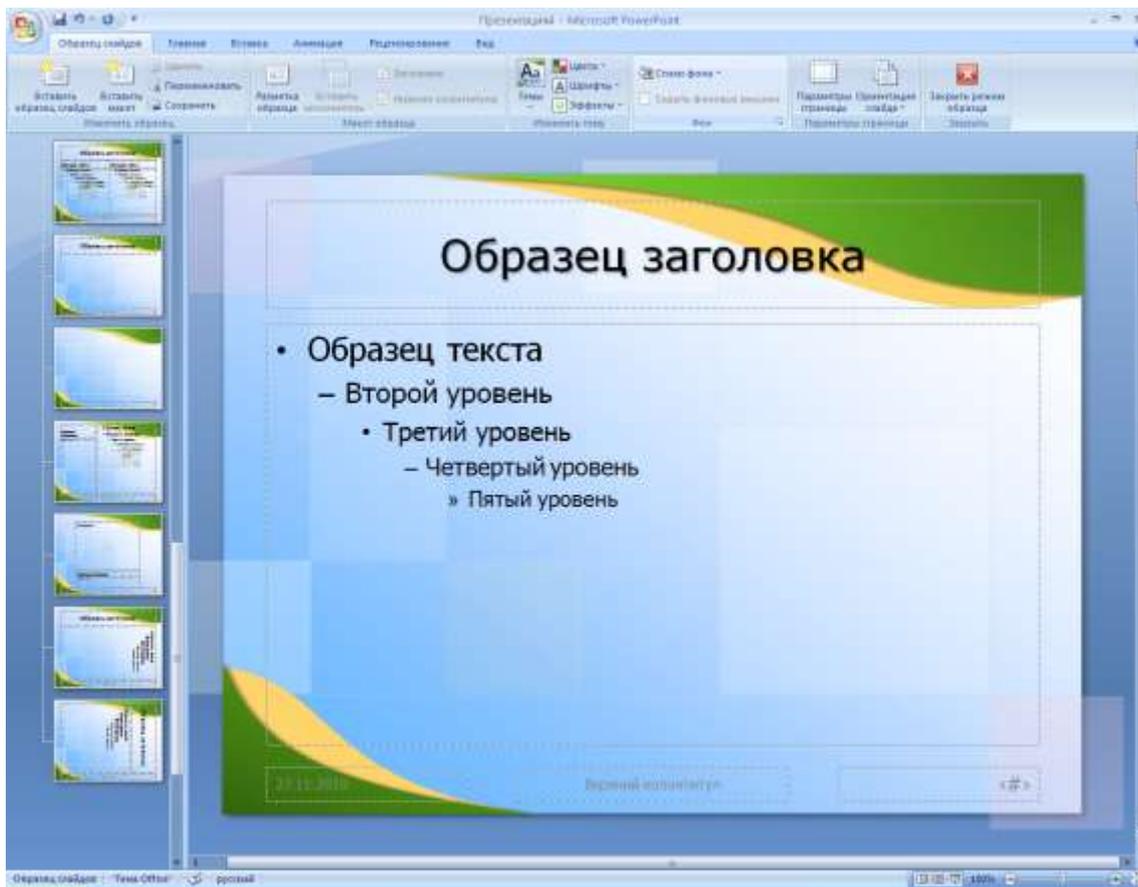
17. Таким же образом нарисуйте горизонтальную сетку. После чего выделите линии сетки и отправьте их На задний план.



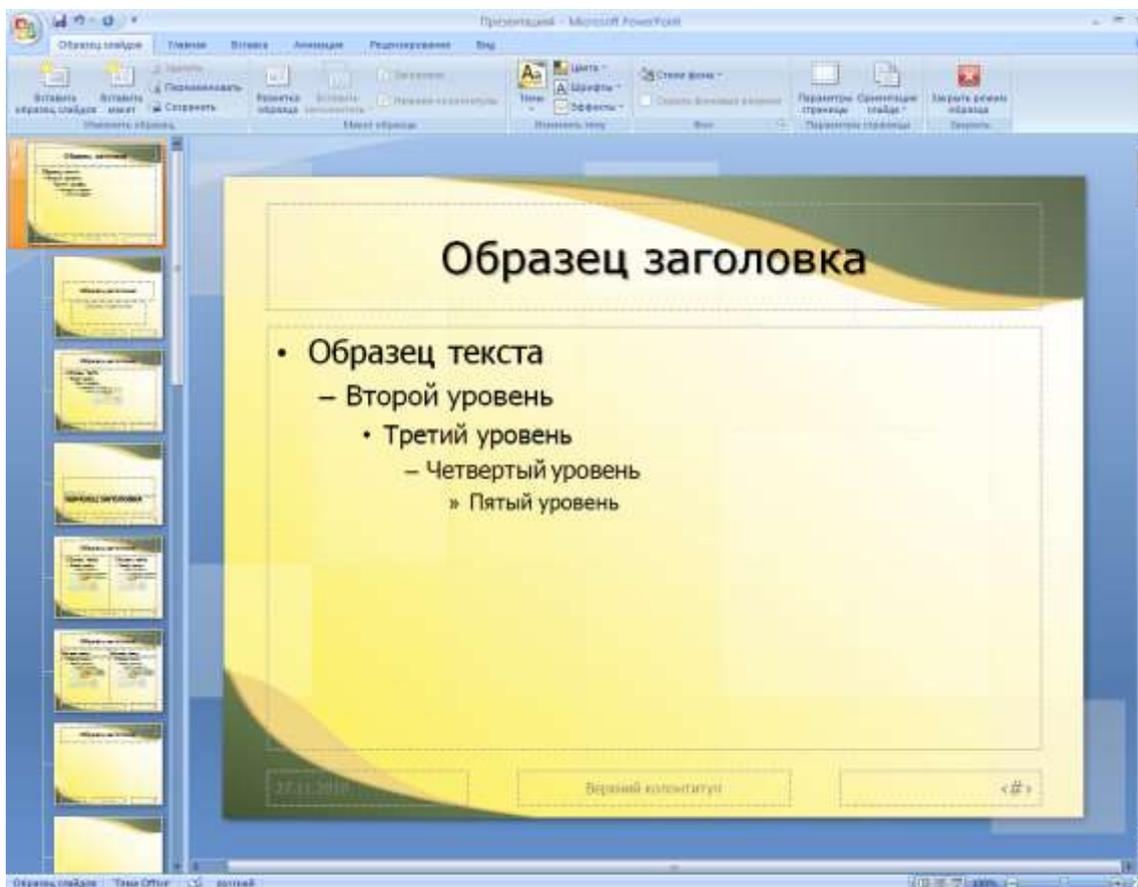
18. Нарисуйте несколько квадратов или прямоугольников по сетке. Задайте им различный цвет и прозрачность (выберите Стили фигур / Формат фигуры). Перемещать фигуры на малый шаг можно стрелочками на клавиатуре зажав клавишу Ctrl.



20. Отрегулируйте прозрачность линий нарисованной сетки (их можно выделить удерживая Shift и сгруппировать для удобства) и фигур, чтобы они были мало заметны. Отправьте квадраты и прямоугольники на задний план. Задайте верхнему и нижнему зеленому «треугольнику» тень.

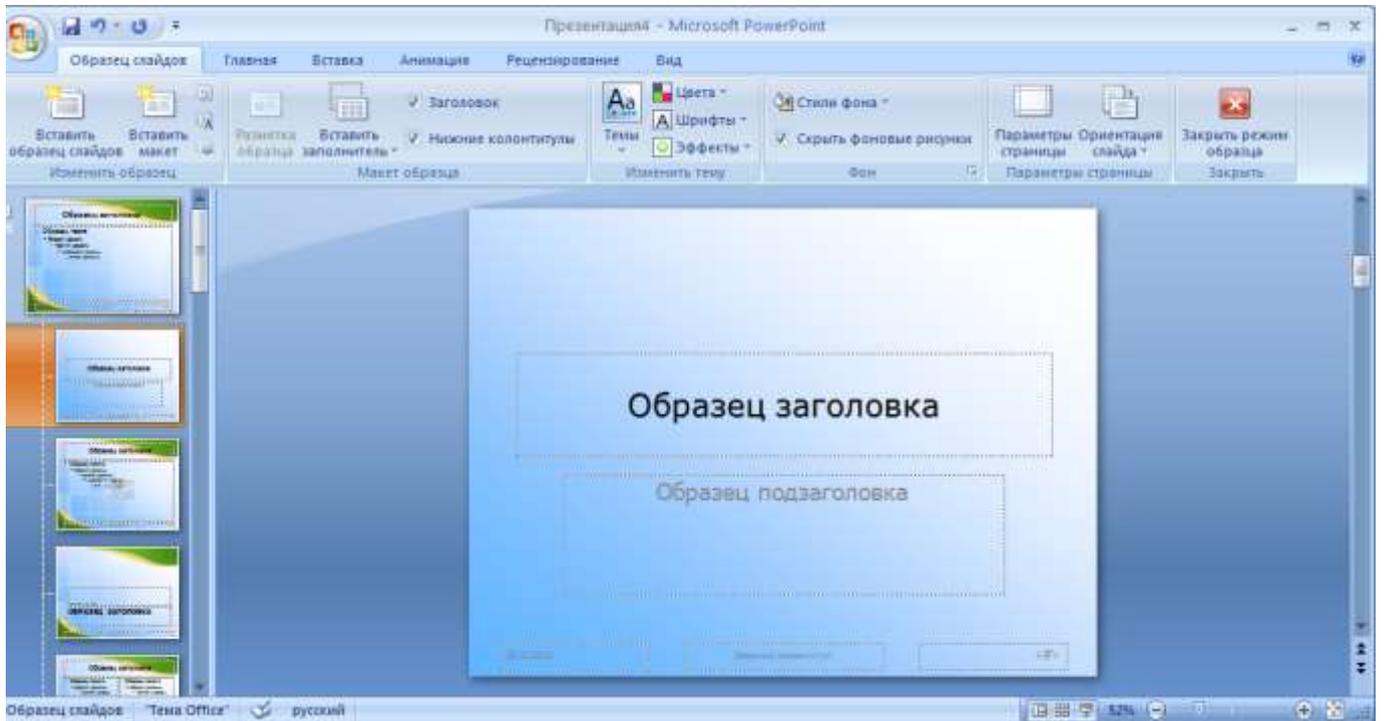


21. Во вкладке Образец слайдов в Цвета можно поменять цветовую схему.

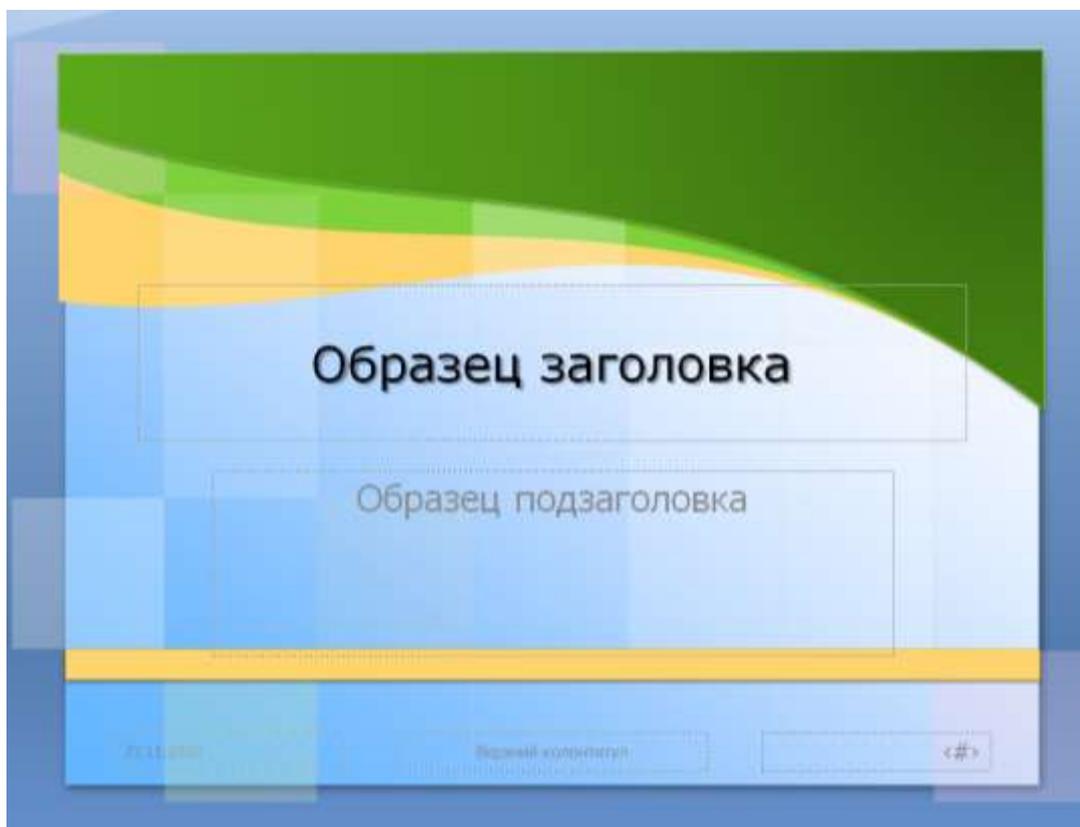


22. Для презентации, необходимо иметь минимум два вида дизайна слайдов – один для титульного и один для остальных слайдов. Если презентация имеет разделы, то желательно и еще один – для слайдов разделов. Перейдите в шаблон титульного слайда (второй).

23. Во вкладке Образец слайдов поставьте галочку напротив Скрыть фоновые рисунки.



24. Нарисуйте фон для слайда как на рисунке ниже.



25. Можете задать анимацию для элементов фона (например, выцветание или изменение прозрачности квадратов или прямоугольников). Сохраните файл и вернитесь из режима Образец слайдов в обычный.

Содержание отчета

Выполненные в ходе работы файлы.

17 Создание презентаций. Графика. Анимация

Цель работы – ознакомление со средством создания графики и добавления анимации.

Задачи работы – получение навыков работы с графическими объектами презентация, получение навыков использования анимации.

Теоретические сведения

Графика и анимация, добавленная в слайды, позволяет оживить презентацию. Основное требование к анимации и графике, вставляемой в презентацию можно сформулировать так: графика и анимация должна подкреплять доклад, акцентировать внимание, «оживлять доклад», а не отвлекать.

Анимация может применяться как ко всему слайду, так и отдельным элементам слайда. Смену слайдов лучше делать типовую для всех слайдов презентации, так она не отвлекает внимание и «объединяет» презентацию. Для деловых презентаций не рекомендуется использовать анимацию, которая нарушает целостность текста, изображения (шашки, жалюзи).

Для добавления анимации, в программе имеется группа команд на вкладке Анимация.

Оборудование, программное обеспечение

Оборудование – ПК с установленной графической операционной системой (Windows или Linux).

Программное обеспечение – MS PowerPoint или Impress (LibreOffice) 97-2003 или MS PowerPoint или Impress (LibreOffice) 2007.

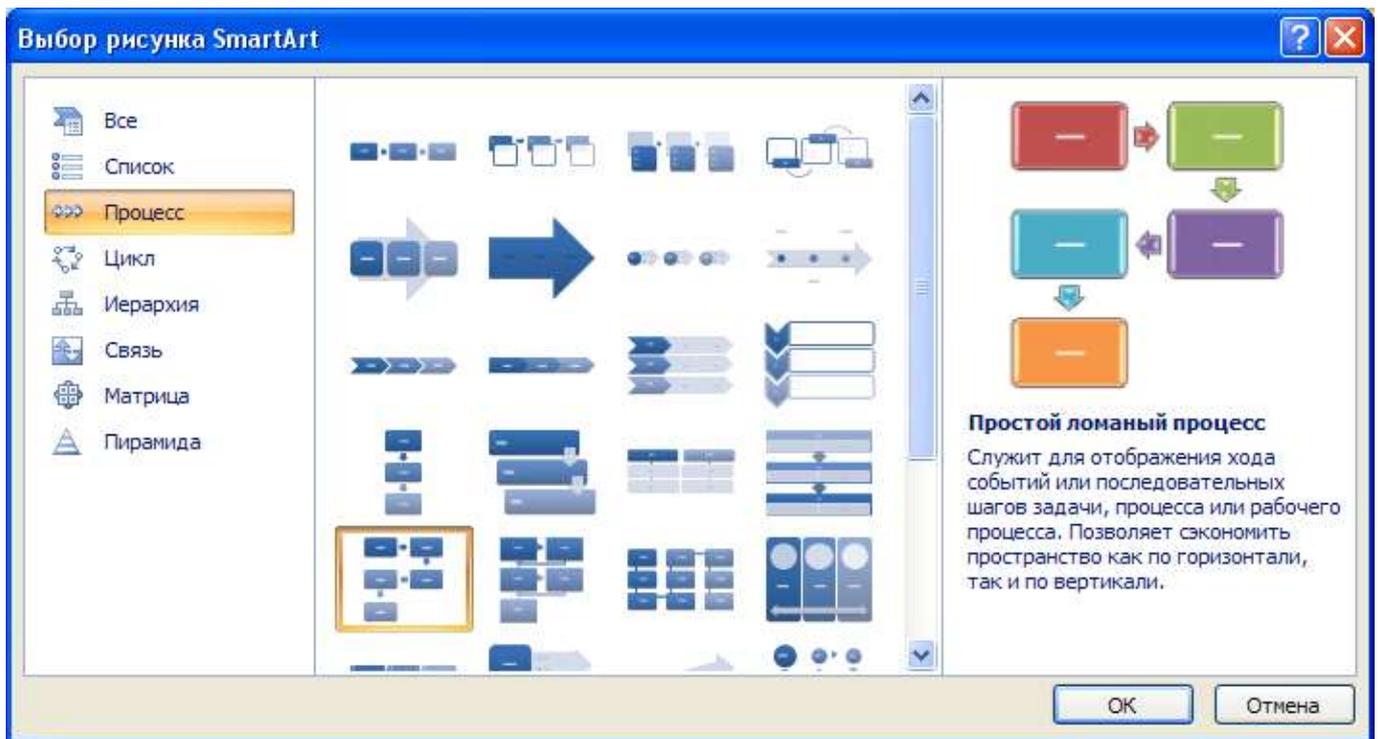
Задание на работу

Создать схему процесса создания сайта, используя:

- 1) умные графические объекты SmartArt;
- 2) средства рисования.

Ход работы

1. Откроем презентацию, созданную в задании создание шаблона. На второй слайд добавим схему «Этапы разработки сайта». На вкладке Вставка выберите команду SmartArt, Простой ломанный процесс.



2. Заполните этапы разработки сайта:

- 1 Составление технического задания
- 2 Разработка дизайна
- 3 Вёрстка
- 4 Программирование
- 5 Тестирование
- 6 Наполнение сайта информацией
- 7 Размещение в сети
- 8 SEO - оптимизация
- 9 Сдача проекта
- 10 Сопровождение и поддержка

Для редактирования удобно пользоваться Областью текста.

Презентация - Microsoft PowerPoint

Главная Вставка Детали Анимация Показ слайдов Редизайнирование Вид Конструктор Формат

Добавить маркер Справа налево Добавить фигуру Масштаб Создать рисунок

Повысить яркость Понижить уровень Область текста

Мастри

Измнить цвета

Стиль SmartArt

Восстановить рисунок Сброс

Заголовок слайда

1 Составление технического задания

2 Разработка дизайна

3 Верстка

4 Программирование

5 Тестирование

6 Наполнение сайта информацией

7 Размещение в сети

8 SEO - оптимизация

9 Сдача проекта

10 сопровождение и поддержка

Введите текст

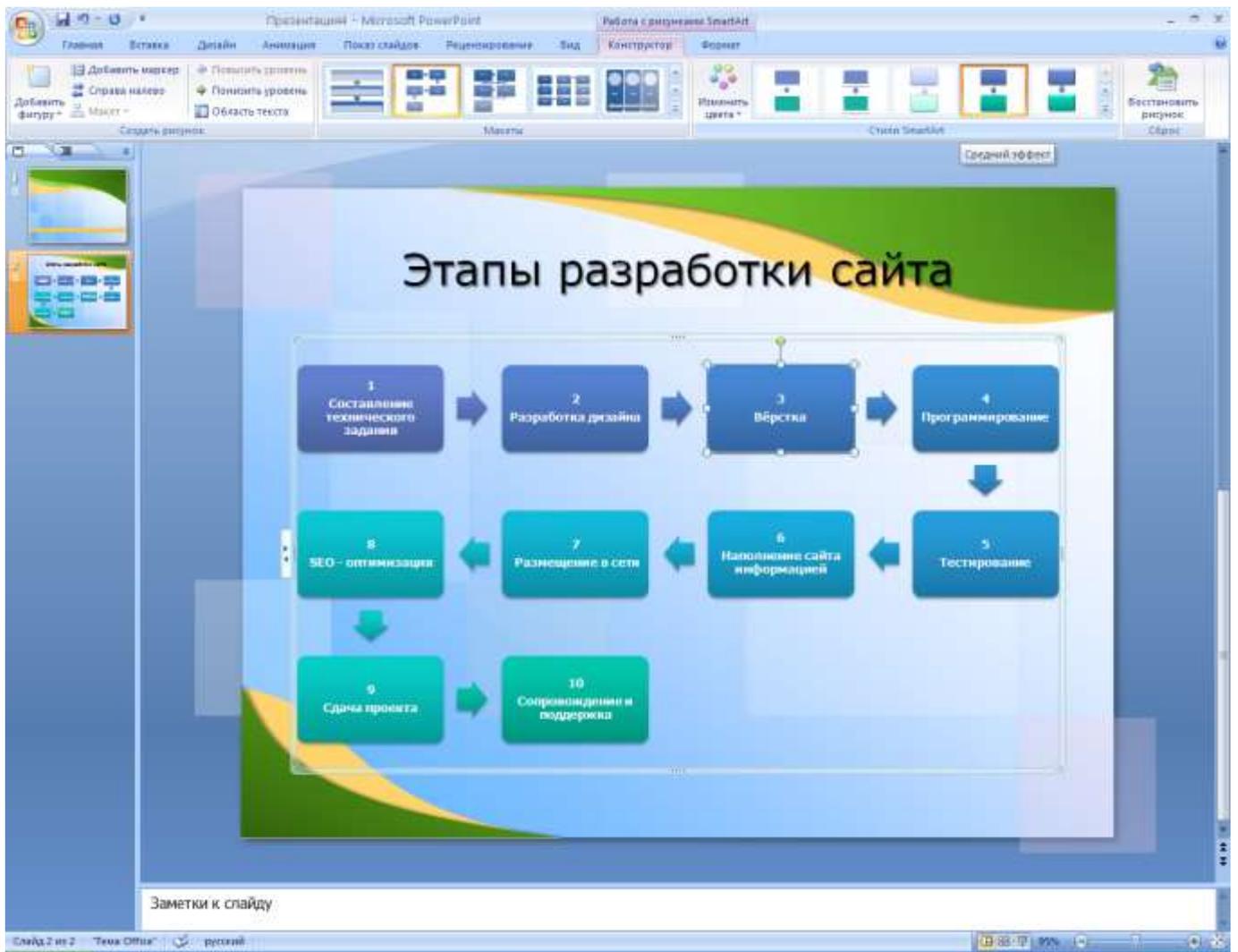
- 1 Составление технического задания
- 2 Разработка дизайна
- 3 Верстка
- 4 Программирование
- 5 Тестирование
- 6 Наполнение сайта информацией
- 7

Простой линейный процесс:
Служит для отображения хода событий или последовательных этапов работы, процесса или рабочего процесса. Позволяет сэкономить пространство как по горизонтали, так и по вертикали. Дополнительные сведения о рисунке, см. рис. 8.1.

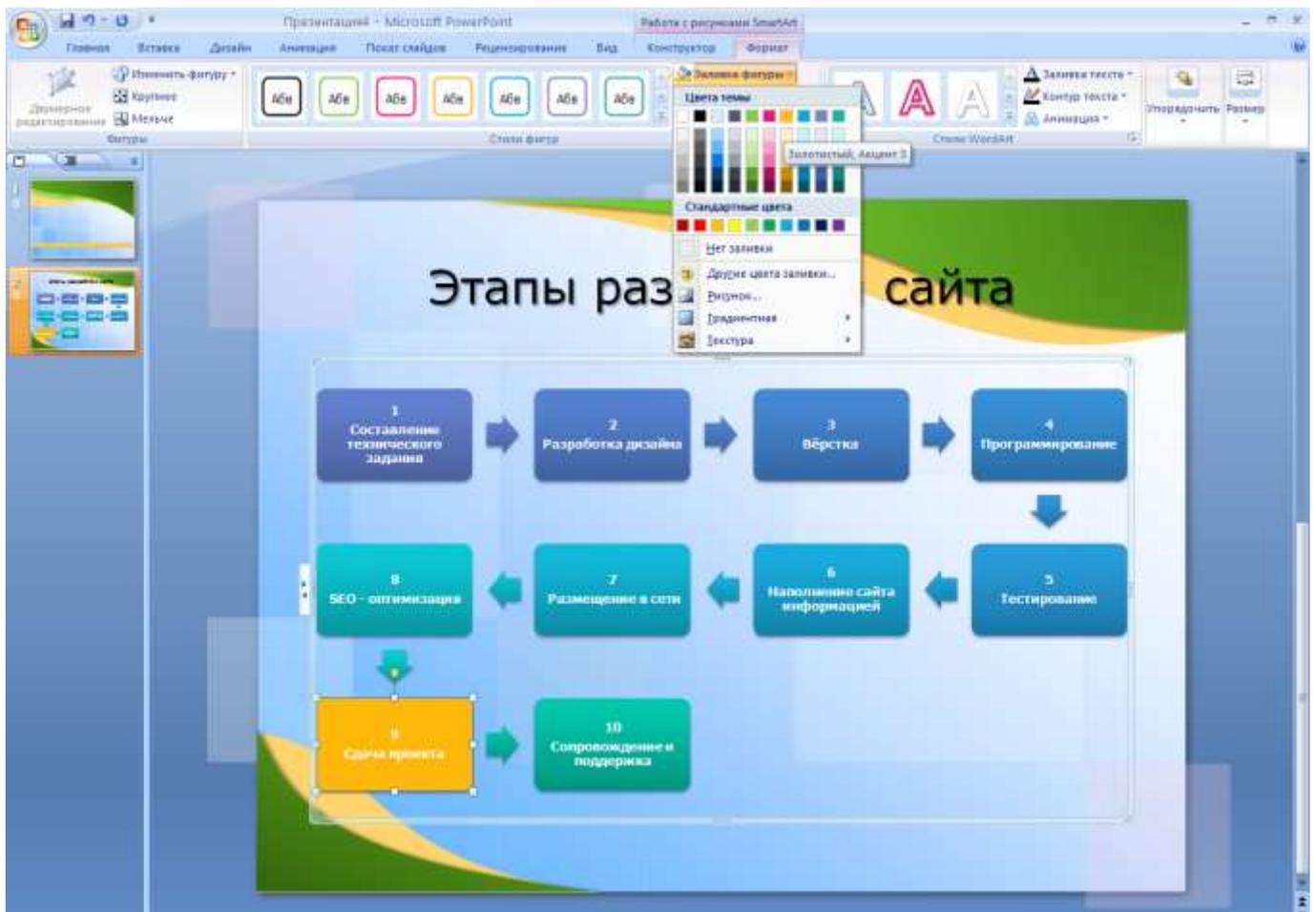
Заметки к слайду

Слайд 2 из 2 Тема Office русский

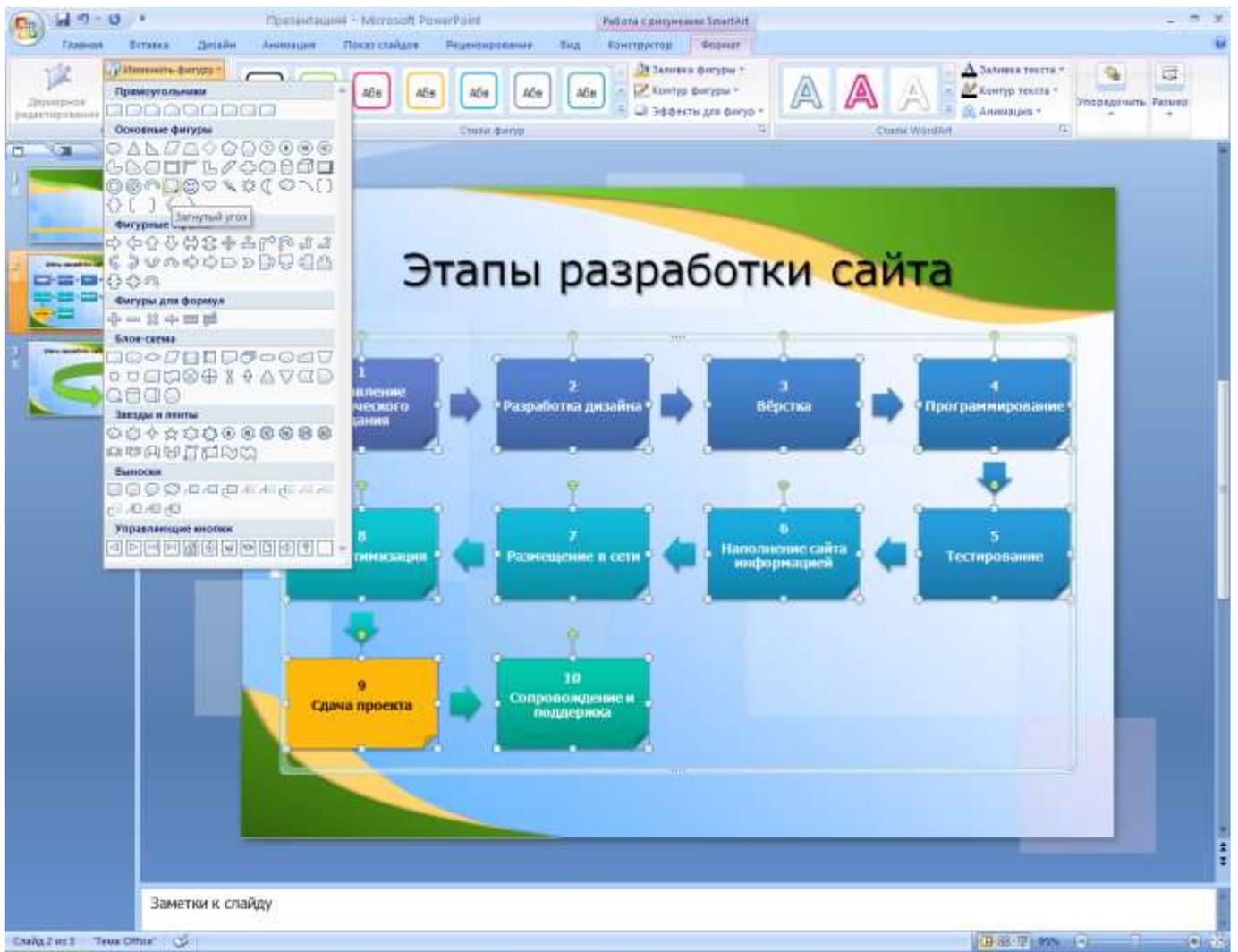
2. В Стилях SmartArt выберите Средний эффект. В Изменить цвета задайте Контрастные цвета 5-6. Добавьте название слайда - Этапы разработки сайта



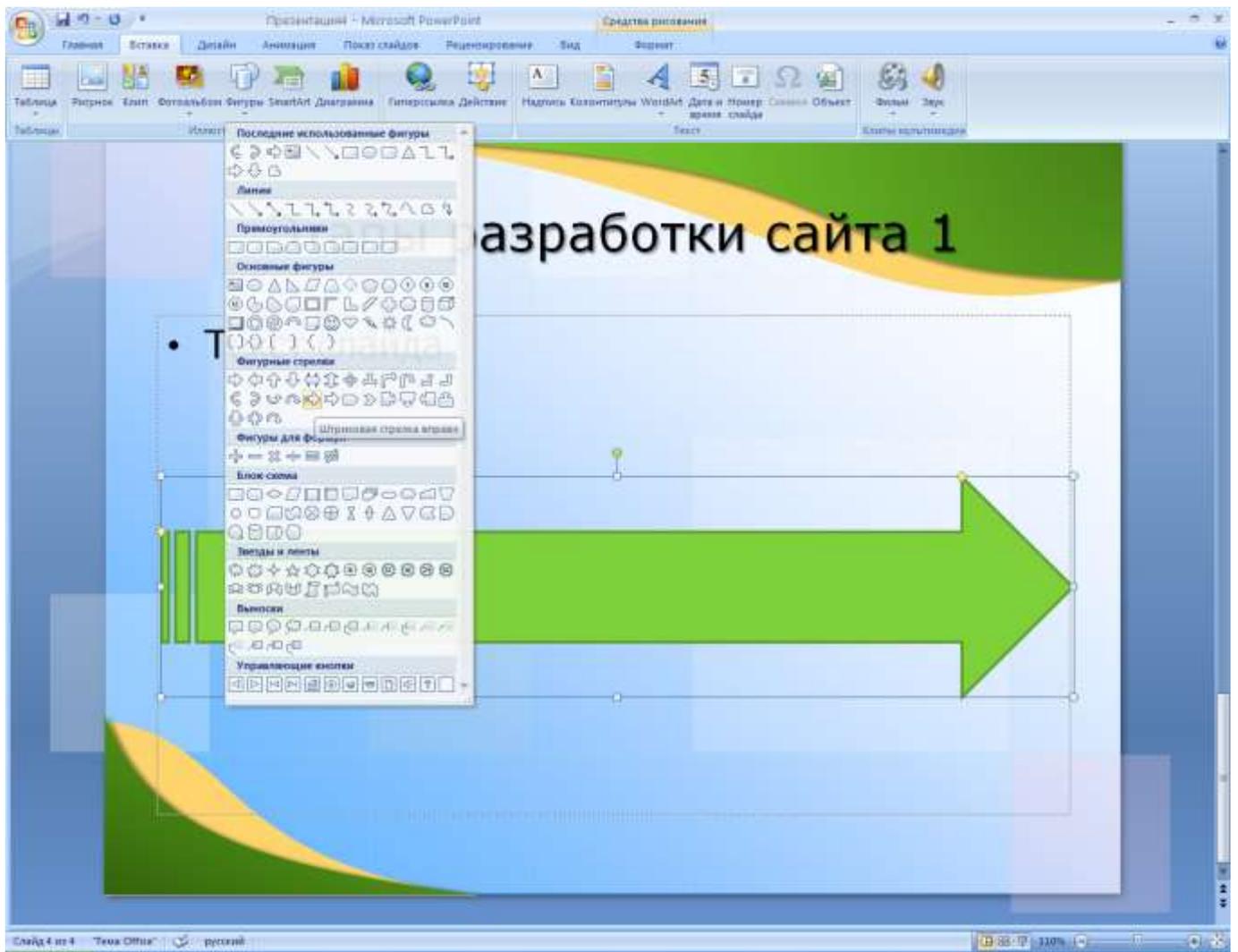
3. Выделите мышкой 9 этап и задайте ему контрастный цвет см. рисунок ниже. Если хотите сохранить эффект выпуклости на фигуре 9 этапа, то отредактируйте Стили фигур / Формат фигуры / Заливка / Градиент.



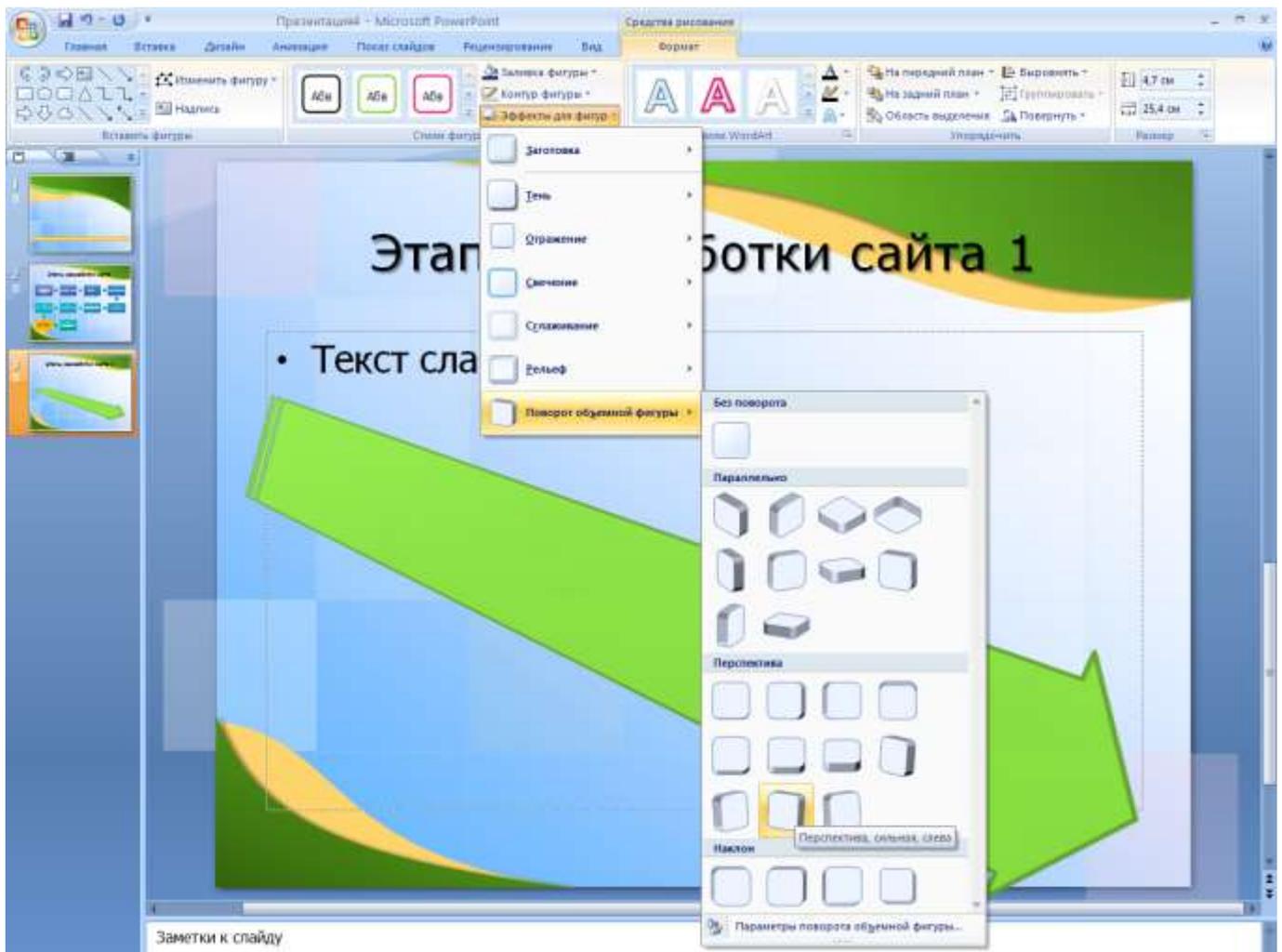
4. Поменяйте для 9 фигуры Заливку текста на черную. Удерживая клавишу Shift выберите мышкой все фигуры этапов и в меню Работа с рисунками SmartArt задайте Изменить фигуру на Загнутый угол.



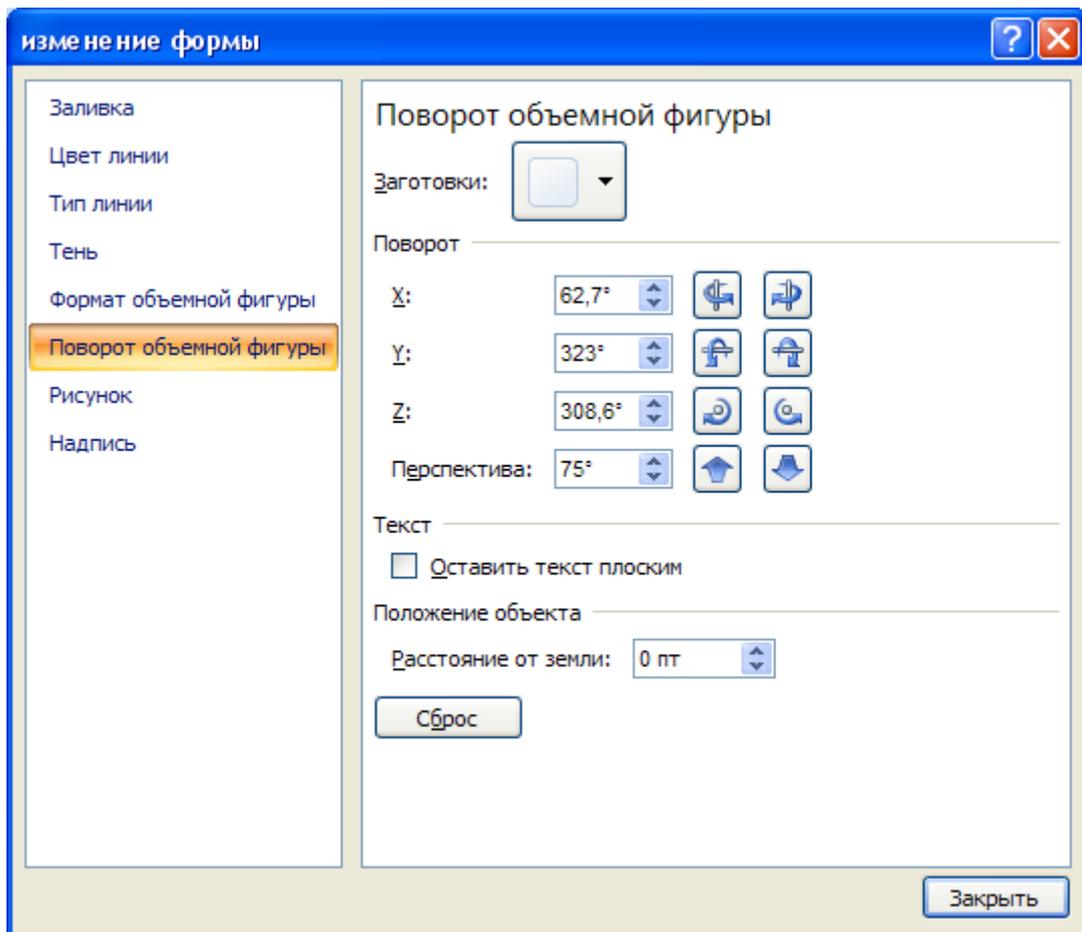
5. Создайте еще один слайд. Назовите Этапы разработки сайта 1. На нем нарисуйте средством Вставка / Фигуры / Штриховая стрелка стрелку.



6. В Эффектах фигуры выберите сильную перспективу. После этого там же Параметры поворота фигуры..



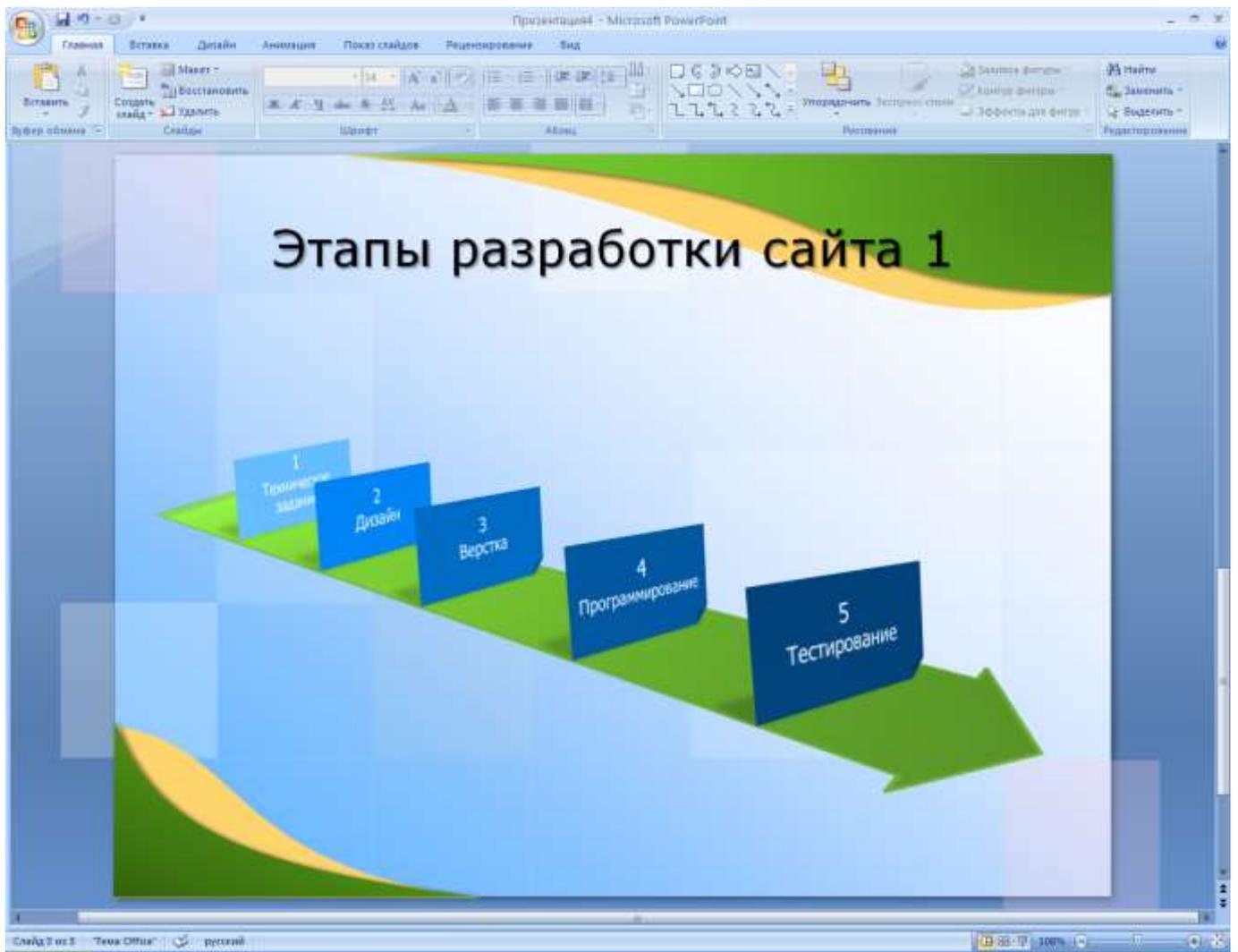
7. В параметрах поворота задайте следующие настройки:



8. Изменяя размер фигуры, добейтесь, чтобы она выглядела примерно так.



9. Нарисуйте прямоугольные блоки с 5 этапами разработки слайда, задайте им перспективу, тень и заливку, как на рисунке ниже.



10. Добавьте анимацию так, чтобы блоки появлялись последовательно.

Содержание отчета

Выполненные в ходе работы файлы.

18 Создание презентаций. Работа со звуком, включение видео.

Теоретические сведения

Общие сведения о видеофрагментах и анимированных изображениях формата GIF

Видеофрагментом называется видео файл в формате AVI и MPEG с расширением AVI, MOV, MPG и MPEG. Обычно видеофрагмент включает речевое сопровождение, например – речь руководителя, который не может лично присутствовать на презентации. Видеофрагмент можно использовать для демонстрации или обучения.

Анимированный файл формата GIF (GIF. Формат графических файлов (с расширением .gif в Microsoft Windows), используемый для отображения в Интернете рисунков с индексированным набором цветов. Поддерживает до 256 цветов и использует сжатие без потерь, то есть сжатие файла без потери данных.) имеет расширение GIF и включает в себя движущееся изображение. Технически анимированные изображения формата GIF не являются фильмами – они только содержат несколько изображений, которые, сменяя друг друга, создают эффект анимации. Файлы этого формата часто используются для создания контрастных элементов оформления и на веб-узлах. Средство «Клип» пакета Microsoft Office распознает файлы формата GIF как видео клипы, однако они не являются цифровыми видео файлами, поэтому не все параметры цифрового видео применимы к анимированным GIF-файлам.

Видеофрагменты и анимированные изображения формата GIF можно добавлять на слайды из файлового архива на компьютере, из организатора клипов (Microsoft), из Интернета или из локальной сети. Чтобы добавить в презентацию видеофрагмент или анимированное изображение формата GIF, вставьте его на отдельный слайд. Существует несколько способов запуска воспроизведения видеофрагмента или GIF-файла: автоматическое воспроизведение при показе слайда, воспроизведение по щелчку мыши и воспроизведение с задержкой в соответствии с установленными значениями интервала времени. Видеофрагмент также можно воспроизводить при показе нескольких слайдов или непрерывно в ходе всей презентации. Можно также установить параметры видеофрагмента, например скрыть рамку или изменить ее размеры.

В отличие от изображений или рисунков, файлы видеофрагментов всегда связаны (Связанный объект. Объект, созданный в файле-источнике и вставленный в файл назначения с поддержанием связи между этими двумя файлами. Связанный объект в файле назначения может быть обновлен при обновлении файла-источника.) с презентацией, а не внедрены (Внедренный объект. Данные, содержащиеся в файле-источнике и вставленные в файл назначения в виде объекта. После внедрения объект становится частью файла назначения. Изменения, вносимые во внедренный объект, отражаются в файле назначения.) в нее. При вставке связанного файла приложение PowerPoint или Impress (LibreOffice) создает на него ссылку. Если затем файл видеофрагмента будет перемещен, PowerPoint или Impress (LibreOffice) не сможет найти путь к нему для воспроизведения. Чтобы приложение PowerPoint или Impress (LibreOffice) могло всегда найти путь к файлам видеофрагмента, перед вставкой видеофрагментов в презентацию скопируйте их в ту же папку, где находится презентация – PowerPoint или Impress (LibreOffice) создаст ссылку и сможет всегда найти видеофрагмент, если его файл остается в папке презентации, даже в случае

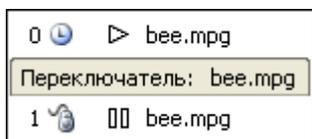
копирования или перемещения папки на другой компьютер. Чтобы гарантировать сохранение пути к связанным файлам презентации, можно воспользоваться средством «Упаковка для записи на компакт-диск». Это средство копирует все файлы презентации в одно расположение (на компакт-диск или в папку) и автоматически обновляет ссылки на файлы видеофрагментов. Если презентацию планируется перенести на другой компьютер или отправить по электронной почте, вместе с ней должны быть скопированы или отправлены и все связанные файлы.

Во время вставки видеофрагмента, вместе с ним в презентацию добавляется эффект триггера паузы. Триггер запускает воспроизведение видеофрагмента по щелчку мыши. Чтобы остановить или запустить воспроизведение видеофрагмента, щелкните область видеофрагмента в презентации.

Эффекты воспроизведения и паузы в режиме автоматического воспроизведения видеофрагментов

Если в презентацию вставить видеофрагмент и затем выбрать параметр **Автоматически**, то в область задач (Область задач. Область в окне приложения Microsoft Office, содержащая часто используемые команды. Ее расположение и небольшие размеры позволяют использовать эти команды, не прекращая работу над файлами.) **Настройка анимации** добавляются два эффекта: пауза и воспроизведение. Эффект паузы позволяет останавливать и запускать воспроизведение видеофрагмента из того же места при повторном щелчке мыши – без этого эффекта видеофрагмент будет по щелчку мыши запускаться сначала.

После вставки видеофрагмента область задач **Настройка анимации** примет вид, как показано на следующем рисунке.



В этом примере первая строка (с цифрой «0») представляет эффект автоматического запуска воспроизведения. Символ часов отображает параметр запуска воспроизведения **Запускать после предыдущего**. Этот параметр позволяет начать воспроизведение видеофрагмента автоматически в момент показа слайда или по завершении другого эффекта (если он есть). Значок треугольника (похожий на символ кнопки воспроизведения на видеомагнитофонах и DVD-проигрывателях) является индикатором эффекта воспроизведения.

Вторая строка является панелью триггера, ниже нее находится строка эффекта паузы (строка с цифрой «1»). Здесь изображен значок мыши и символ из двух полос (похожий на символ кнопки паузы на видеомагнитофонах и DVD-проигрывателях). Этот эффект добавляется при автоматическом воспроизведении видеофрагмента или воспроизведении по щелчку мыши. Присутствие этой строки эффекта паузы под панелью триггеров означает, что для запуска видеофрагмента необходимо щелкнуть в области видеофрагмента, а не какое-либо другое место на слайде.

Эффект паузы и воспроизведение видеофрагментов по щелчку мыши

После вставки видеофрагмента, область задач **Настройка анимации** примет вид, как показано на следующем рисунке.



Независимо от того, какие эффекты добавляются для автоматически запускаемых видеофрагментов, эффект паузы – строка со значком мыши и двухполосным символом – является единственным эффектом, применимым для видеофрагмента, запускаемого по щелчку мыши.

Добавление видеофрагмента в презентацию

Чтобы избежать проблем с неработающими ссылками, рекомендуется, перед добавлением видеофрагментов, копировать их в ту же папку, где находится презентация.

В обычном режиме просмотра щелкните слайд, на который нужно добавить видеофрагмент или анимированный GIF-файл.

На вкладке **Вставка** в группе **Клипы мультимедиа** щелкните стрелку в разделе **Фильм**.



Выполните одно из следующих действий:

Щелкните **Фильм из файла**, выберите папку, в которой находится файл, затем дважды щелкните файл, который нужно добавить.

Щелкните **Фильм из организатора клипов**, выберите нужный клип в области задач (Область задач. Область в окне приложения Microsoft Office, содержащая часто используемые команды. Ее расположение и небольшие размеры позволяют использовать эти команды, не прекращая работу над файлами.) **Клип**, затем щелкните его для добавления на слайд.

СОВЕТ. Перед добавлением клипа в презентацию его можно предварительно просмотреть. В области задач **Клип** в поле **Результаты**, в котором перечислены доступные клипы, наведите указатель мыши на эскиз клипа, щелкните отобразившуюся стрелку и выберите **Просмотр и свойства**.

Выбор между автоматическим воспроизведением или воспроизведением по щелчку мыши

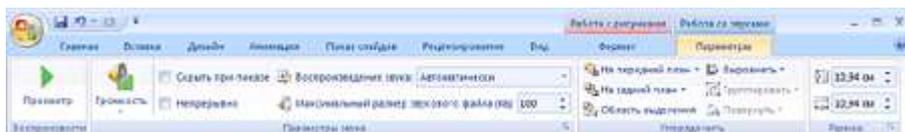
Во время вставки видеофрагмента приложение PowerPoint или Impress (LibreOffice) предложит выбрать способ воспроизведения: автоматический (**Автоматически**) или по щелчку мыши (**По щелчку**).

Чтобы воспроизведение видеофрагмента начиналось автоматически во время показа слайда, выберите **Автоматически**. Чтобы включить паузу, щелкните видеофрагмент. Чтобы продолжить воспроизведение, щелкните его снова.

Чтобы начать воспроизведение видеофрагмента вручную по щелчку мыши на слайде, выберите **По щелчку**.

Во время вставки видефрагмента в презентацию добавляется эффект триггера паузы. Триггер запускает воспроизведение видефрагмента по щелчку мыши в области слайда. Например, щелкните в презентации рамку видефрагмента, чтобы включить паузу, щелкните еще раз, чтобы продолжить воспроизведение.

ПРИМЕЧАНИЕ. Этот параметр в любой момент можно изменить. Щелкните видефрагмент, затем в разделе **Работа с фильмами** щелкните вкладку **Параметры**. В группе **Параметры фильма** выберите нужный параметр из списка **Фильм**.



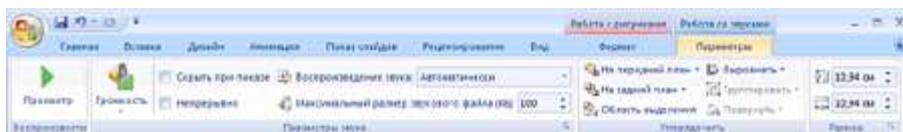
Воспроизведение видефрагмента в полноэкранном режиме

В ходе презентации воспроизведение видефрагмента может выполняться в двух режимах – как часть слайда, и во весь экран. В этом случае воспроизведение называется полноэкранным. В зависимости от разрешения исходного файла видефрагмента, при его увеличении до размеров экрана во время воспроизведения могут возникнуть искажения картинки. Следует всегда предварительно просматривать видефрагмент, чтобы в случае искажений или нечеткости изображения можно было отключить полноэкранный режим воспроизведения. Как правило, видефрагмент с маленькой картинкой будет нечетким в полноэкранном режиме.

При выборе автоматического воспроизведения в полноэкранном режиме можно перетащить рамку видефрагмента из слайда на серую область, чтобы он не был виден и не мелькал на слайде до начала полноэкранного воспроизведения видефрагмента.

В обычном режиме просмотра щелкните рамку видефрагмента на том слайде, где его нужно воспроизвести во весь экран.

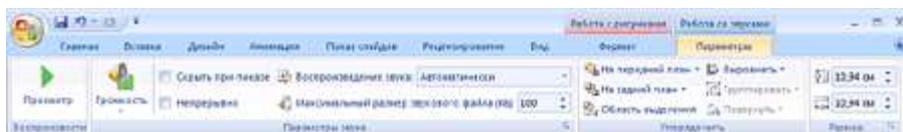
В разделе **Работа с фильмами** на вкладке **Параметры** в группе **Параметры фильма** установите флажок **Воспроизведение в полноэкранном режиме**.



Предварительный просмотр видефрагмента

В обычном режиме просмотра щелкните рамку видефрагмента на том слайде, где его нужно предварительно просмотреть.

В разделе **Работа с фильмами** на вкладке **Параметры** в группе **Воспроизвести** щелкните **Просмотр**.



Советы

Можно также предварительно посмотреть видеофрагмент в обычном режиме просмотра, для этого дважды щелкните его.

Если видеофрагмент или анимированный GIF-файл являются частью пользовательской анимации, для предварительного просмотра щелкните Воспроизвести в области задач (Область задач. Область в окне приложения Microsoft Office, содержащая часто используемые команды. Ее расположение и небольшие размеры позволяют использовать эти команды, не прекращая работу над файлами.) Настройка анимации .

Если презентация состоит из нескольких слайдов, **используйте рамку заголовка** для определения слайда, который содержит рамку видеофрагмента, чтобы понять, какой слайд нужно щелкнуть для воспроизведения.

Если видеофрагмент вставлен как объект, воспроизводимый с помощью проигрывателя Microsoft Windows Media, для управления воспроизведением видеофрагмента используйте кнопки Остановить, Начать и Пауза в проигрывателе Microsoft Windows Media.

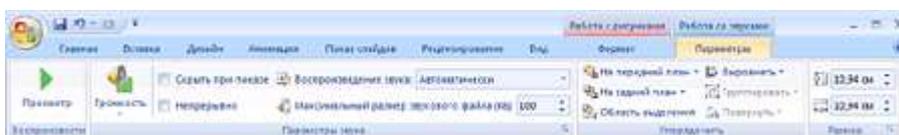
Воспроизведение видеофрагмента во время показа нескольких слайдов

Чтобы вставленный в презентацию видеофрагмент не остановился при следующем щелчке мыши и воспроизведение продолжалось при переходе к следующему слайду, укажите, когда необходимо остановить воспроизведение.

ПРИМЕЧАНИЕ. Для выполнения следующей процедуры необходимо добавить эффект воспроизведения. Если видеофрагмент уже вставлен и выбран режим его воспроизведения **По щелчку**, можно переключиться в режим **Автоматически**, чтобы добавить эффект воспроизведения.

Переключение между автоматическим режимом воспроизведения и по щелчку мыши

- 1 В обычном режиме просмотра щелкните видеофрагмент.
- 2 В разделе **Работа с фильмами** на вкладке **Параметры** в группе **Параметры фильма** в списке **Воспроизвести фильм** щелкните **Автоматически**.

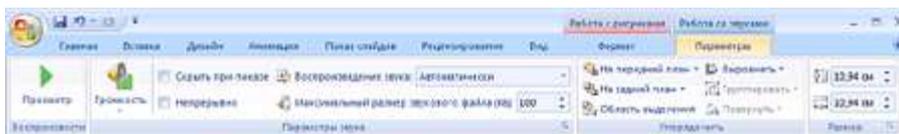


Продолжение воспроизведения видеофрагмента во время показа нескольких слайдов

ПРИМЕЧАНИЕ. Данная процедура запускает демонстрацию видеофрагмента от начала до конца. **Повторное воспроизведение видеофрагмента** при этом невозможно.

В обычном режиме просмотра щелкните рамку видеофрагмента на слайде.

В разделе **Работа с фильмами** на вкладке **Параметры** в группе **Параметры фильма** в списке **Воспроизвести фильм** щелкните **Для всех слайдов**.



На вкладке **Анимация** в группе **Анимация** выберите команду **Настройка анимации**.



В области задач (Область задач. Область в окне приложения Microsoft Office, содержащая часто используемые команды. Ее расположение и небольшие размеры позволяют использовать эти команды, не прекращая работу над файлами.) **Настройка анимации** щелкните строку, отображающую эффект воспроизведения видеофрагмента (строку с треугольником), щелкните стрелку, а затем щелкните **Параметры эффекта**.

Чтобы видеофрагмент воспроизводился во время показа нескольких слайдов, в группе **Закончить** щелкните **После** и укажите общее количество слайдов, во время показа которых должно продолжаться воспроизведение файла.

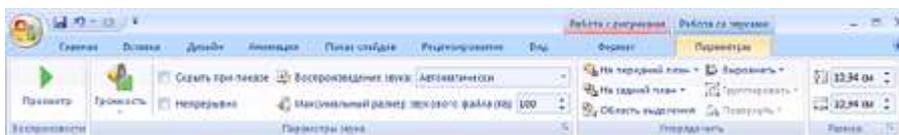
СОВЕТ. Если выбрано максимальное значение 999, и длительность видеофрагмента достаточна, воспроизведение будет выполняться во время всей презентации, при этом нет необходимости изменять это значение в случае добавления или удаления слайдов из презентации.

Воспроизведение видеофрагмента непрерывно в ходе всей презентации

Может потребоваться воспроизведение видеофрагмента в ходе всей презентации или до момента остановки его вручную. Если длительность видеофрагмента меньше, чем презентация, можно установить для него повторное воспроизведение.

В обычном режиме просмотра щелкните рамку видеофрагмента на слайде.

В разделе **Работа с фильмами** на вкладке **Параметры** в группе **Параметры фильма** установите флажок **Непрерывное воспроизведение**.



СОВЕТ. При заиклировании видеофрагмента он будет воспроизводиться повторно до щелчка мыши на нем. Также можно выбрать режим **воспроизведения видеофрагмента во время показа группы слайдов**.

ПРИМЕЧАНИЯ

Если видеофрагмент не воспроизводится, обратитесь к разделу [Видеофрагмент не воспроизводится](#) для решения проблемы.

Можно также [удалить видеофрагмент](#).

19 Создание презентаций. Разработка презентации, управление показом

Способ 1

Для этого способа необходимо иметь микрофон или наушники с микрофоном подключаемые к компьютеру.

1. На первом этапе составляем презентацию в MS PowerPoint или Impress (LibreOffice) и доклад к презентации. Подключаем микрофон к компьютеру. В презентации не устанавливаем время показа слайдов, а устный доклад необходимо отрепетировать.
2. Во вкладке **Показ слайдов** MS PowerPoint или Impress (LibreOffice) выбираем команду **Звукозапись** (см. рисунок 1).

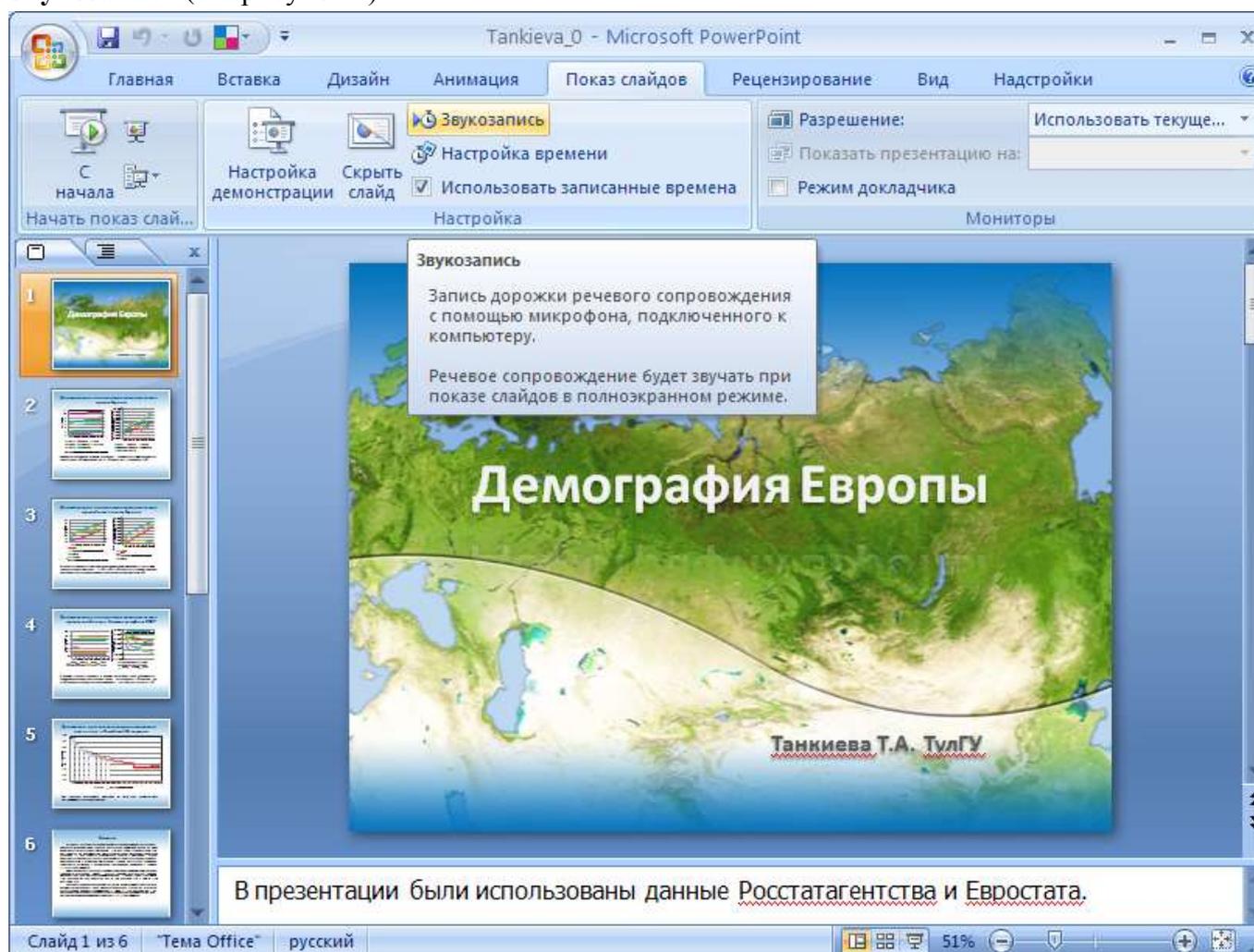


Рисунок 20

3. В появившемся диалоговом (см. рисунок 2) окне можно проверить работу микрофона – команда **Громкость микрофона**. Если к слайдам планируется длительное звуковое сопровождение, то необходимо выбрать опцию **связать речевое сопровождение с:**, по умолчанию указывается та же папка где находится файл презентации. В этом случае звуковое сопровождение будет сохраняться не в презентации, а записываться в отдельные аудиофайлы к каждому слайду в указанную папку. При копировании презентации будет необходимо вместе с презентацией копировать и эти файлы.

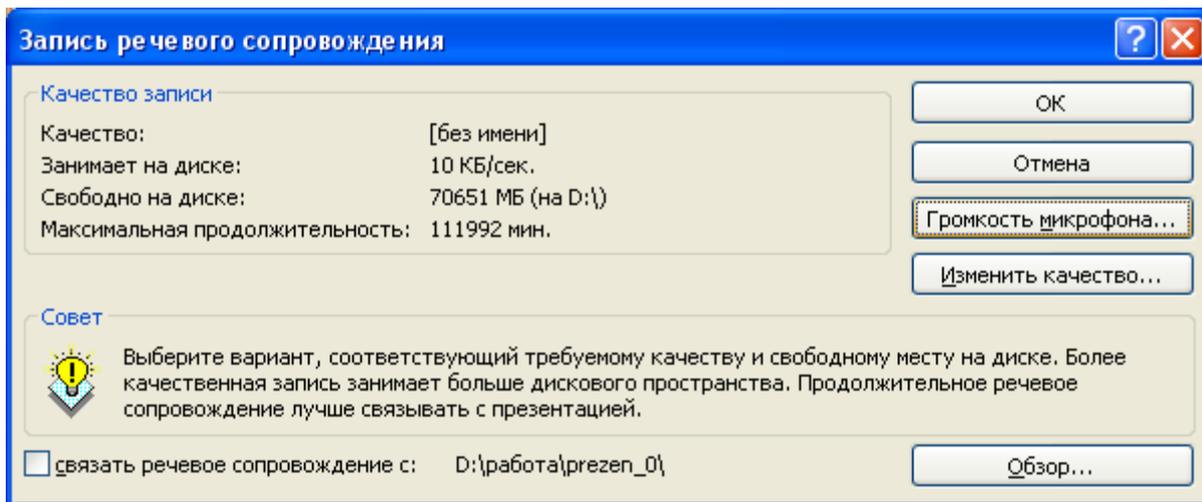


Рисунок 21

4. После того как необходимые опции будут выбраны, нажмите **ОК** – запустится показ слайдов. Проговаривайте доклад в микрофон, по щелчку мыши переключайте слайд в нужное время доклада. По завершению презентации, появится диалоговое окно запрашивающее сохранение времени показа слайдов. Чтобы слайды автоматически переключались под звуковое сопровождение ответьте **Сохранить** (см. рисунок 3).

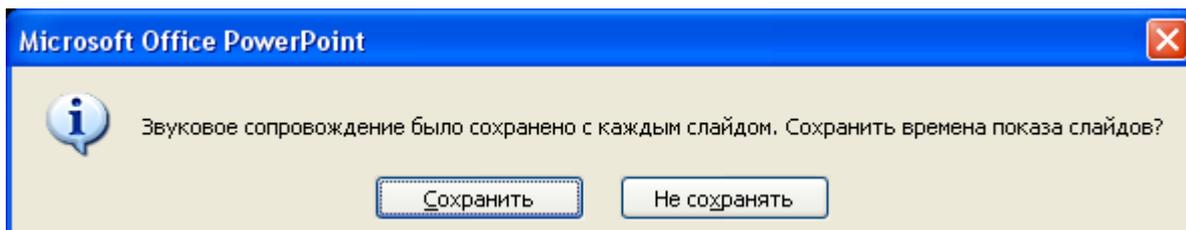


Рисунок 22

5. После откроется презентация в режиме **Сортировщик слайдов**, рядом с каждым слайдом будет записано время его показа. В меню вкладки **Вид** Вы можете переключиться в **Обычный режим**.

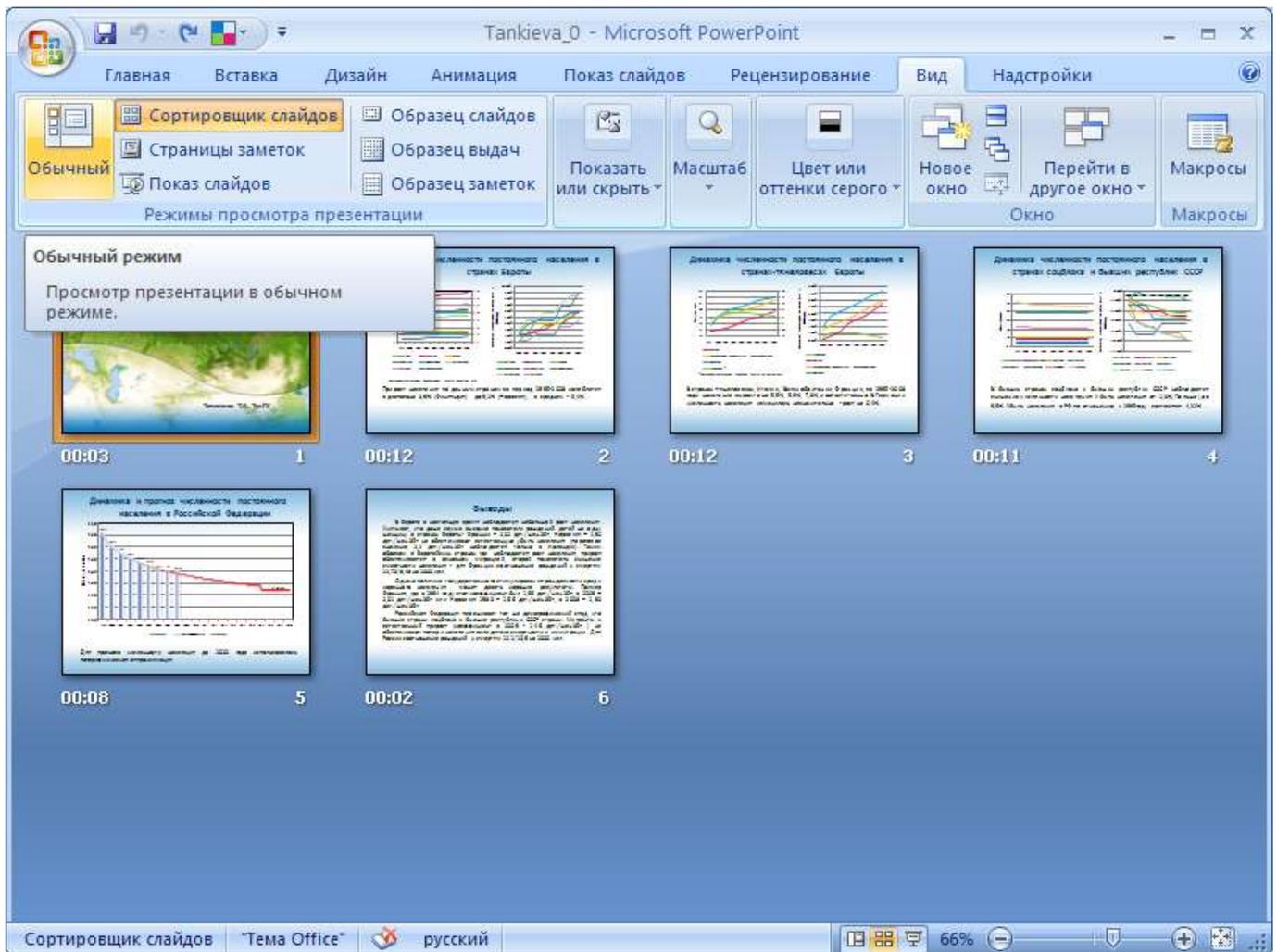


Рисунок 23

6. Обратите внимание на каждый слайд презентации добавился значок-мегафон с которым связан звук. Запустите показ презентации (клавиша F5). Если качество доклада удовлетворительно – сохраните презентацию, если нет – отмените вставку звука (клавиши Ctrl Z) и повторите Звукозапись вновь (этапы 2-6).

Способ 2

1. С помощью любого доступного устройства (например, диктофона сотового телефона) запишите аудиофайл доклада. Скопируйте записанный аудиофайл в ту же папку, где находится подготовленная Вами презентация MS PowerPoint или Impress (LibreOffice).

2. К первому слайду презентации во вкладке **Вставка** MS PowerPoint или Impress (LibreOffice) выберите команду **Звук из файла...** В появившемся диалоговом окне укажите Ваш аудиофайл с записанным докладом.

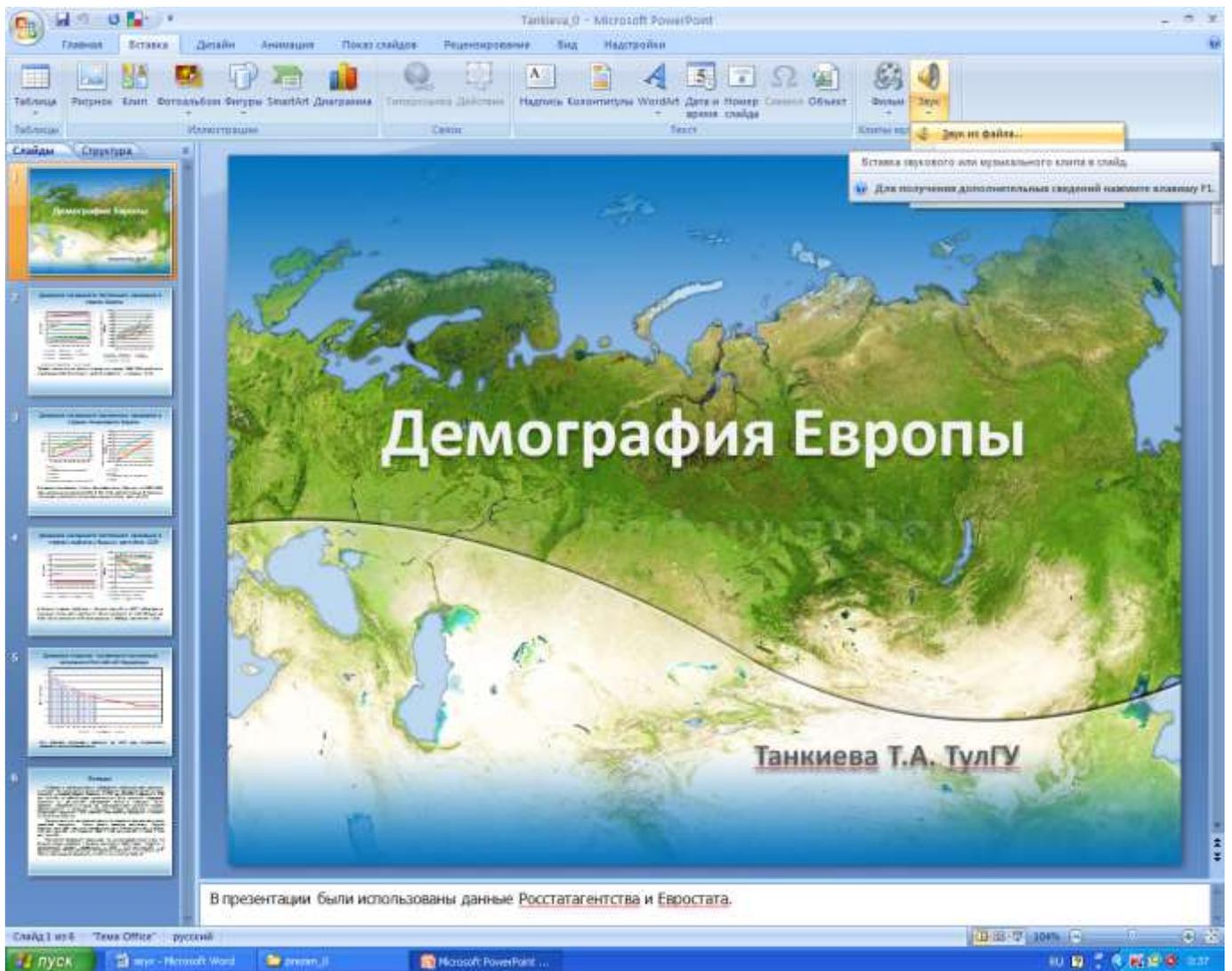


Рисунок 24

3. На вопрос «Воспроизводить звук при показе слайдов?» ответьте **Автоматически**.

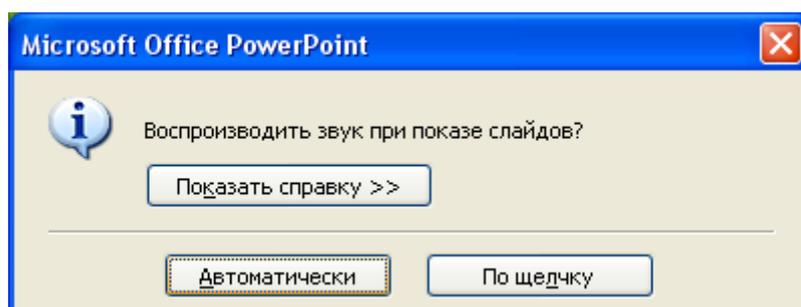


Рисунок 25

4. Обратите внимание, что на первом слайде появился значок-мегафон, с которым связан звук. Во вкладке **Работа со звуком** (эта вкладка активна когда выбран значок-мегафон на слайде) установите **Воспроизведение звука: Для всех слайдов** (если у Вас доклад записан одним аудиофайлом ко всем слайдам презентации). Поставьте галочку **Скрыть при показе**, чтобы значок-мегафон при показе презентации был невидим.

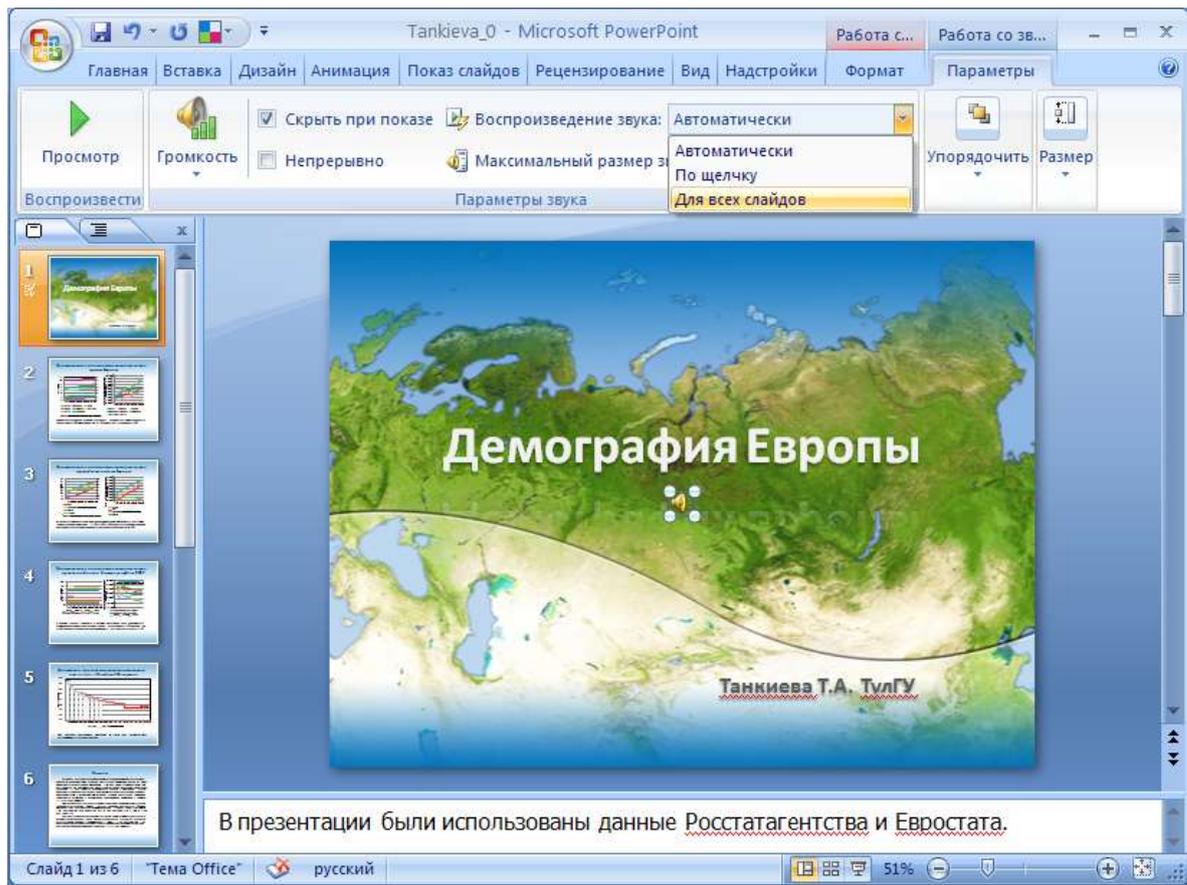


Рисунок 26

5. Теперь необходимо настроить время показа слайдов презентации так чтобы их демонстрация совпадала с аудиосопровождением. Для этого во вкладке **Показ слайдов** выберите команду **Настройка времени** (рисунок 8).

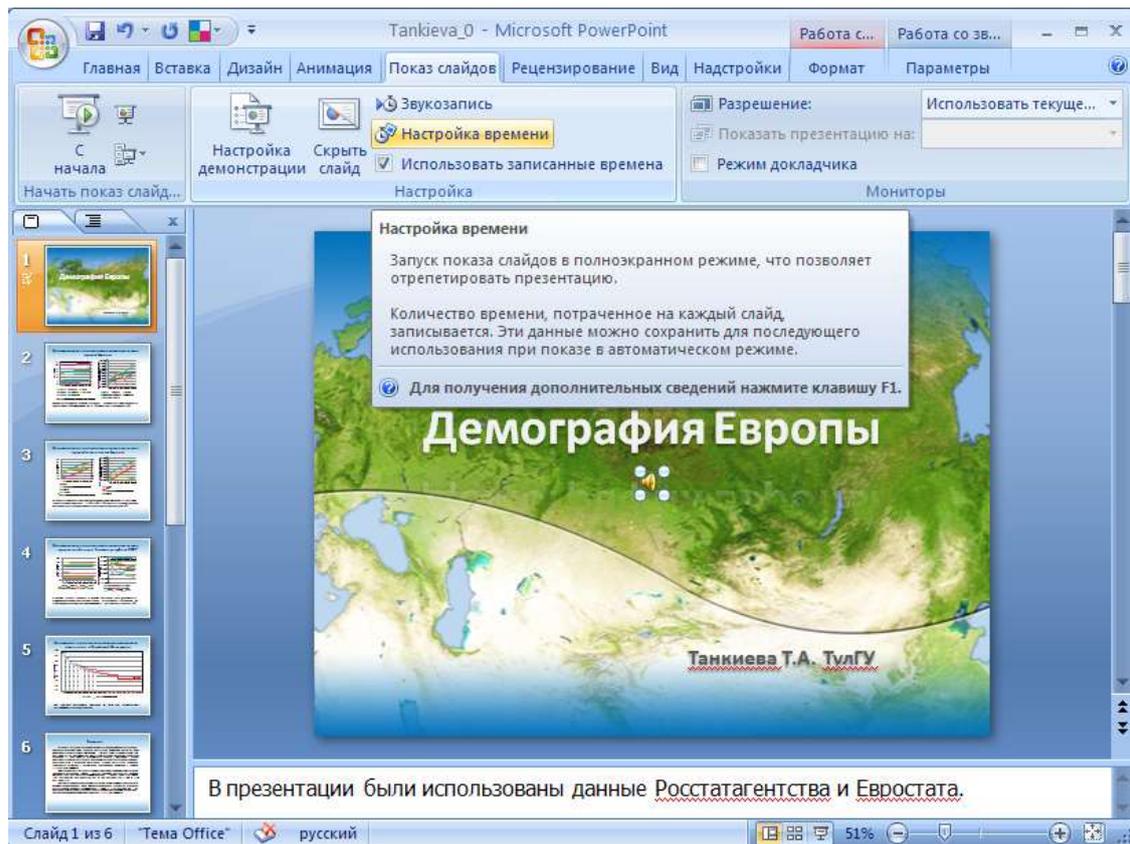


Рисунок 27

6. Запустится показ слайдов, при этом на экране будет находиться панель **Репетиция** (рисунок 9). На панели **Репетиция** нажимайте **Далее** (прямая стрелка) когда по звуковому сопровождению доклада будет требоваться смена слайда. По завершению сохраните время показа слайдов.

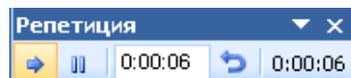


Рисунок 28

Библиографический список рекомендуемой литературы

Основной

1. Баранов, Р. Д. Практические аспекты разработки веб-ресурсов : учебное пособие / Р. Д. Баранов, С. А. Иноземцева, А. А. Рябова. — Саратов : Вузовское образование, 2018. — 121 с. — ISBN 978-5-4487-0263-1. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/75692.html>. — Режим доступа: для авторизир. пользователей. - DOI: <https://doi.org/10.23682/75692>
2. Вагин, Д. В. Современные технологии разработки веб-приложений : учебное пособие / Д. В. Вагин, Р. В. Петров. — Новосибирск : Новосибирский государственный технический университет, 2019. — 52 с. — ISBN 978-5-7782-3939-5. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/98738.html>. — Режим доступа: для авторизир. пользователей
3. Коваленко, Ю. В. Основы работы в Microsoft Office PowerPoint 2007 : учебно-методическое пособие / Ю. В. Коваленко. — Омск : Омская юридическая академия, 2017. — 50 с. — ISBN 978-5-98065-154-1. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/86176.html>. — Режим доступа: для авторизир. пользователей

Дополнительный

1. Макарова, Т. В. Веб-дизайн : учебное пособие / Т. В. Макарова. — Омск : Омский государственный технический университет, 2015. — 148 с. — ISBN 978-5-8149-2075-1. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/58086.html>. — Режим доступа: для авторизир. пользователей
2. Качановский, Ю. П. Технологии создания мультимедийных презентаций в Microsoft PowerPoint : методические указания к проведению лабораторной работы по курсу «Информатика» / Ю. П. Качановский, А. С. Широков. — Липецк : Липецкий государственный технический университет, ЭБС АСВ, 2014. — 38 с. — ISBN 2227-8397. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/55165.html>. — Режим доступа: для авторизир. пользователей
3. Шульгин, В. П. Создание эффектных презентаций с использованием PowerPoint 2013 и других программ / В. П. Шульгин, М. В. Финков, Р. Г. Прокди. — Санкт-Петербург : Наука и Техника, 2015. — 256 с. — ISBN 2227-8397. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/43324.html>. — Режим доступа: для авторизир. пользователей