

Министерство науки и высшего образования Российской Федерации

Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им Н.Э. Баумана)

Н.А. Борсук

УЧЕБНЫЙ ПРАКТИКУМ ПО ДИСЦИПЛИНЕ

«УЧЕБНАЯ ПРАКТИКА. УЧЕБНО-ТЕХНОЛОГИЧЕСКИЙ ПРАКТИКУМ»

Калуга, 2019

Практические задания разработаны в соответствии с учебным планом КФ МГТУ им. Н.Э. Баумана по направлению подготовки 09.03.01 «Информатика и вычислительная техника» бакалавра (профиль «Вычислительные машины, комплексы, системы и сети»)

– кафедрой «Информационные системы и сети» ИУ2-КФ
протокол № _____ от «____» _____ 201__ г.

– методической комиссией факультета ИУ-КФ
 протокол № _____ от « _____ » _____ 201 ____ г.

– методической комиссией Калужского филиала МГТУ им. Н.Э. Баумана
протокол № _____ от « ____ » _____ 201 ____ г.

Рецензент:
к.т. н., доцент кафедры ИУ6 - КФ _____ А.Б. Лачихина

Аннотация

Приведены теоретические положения и задания на практику.

© Борсук Н.А.

СОДЕРЖАНИЕ

1. ЦЕЛЬ И ЗАДАЧА УЧЕБНОГО ПРАКТИКУМА.....	4
2. ЗАДАНИЕ И ТРЕБОВАНИЯ К СОДЕРЖАНИЮ ОТЧЕТА.....	4
3. ПРАКТИЧЕСКАЯ РАБОТА № 1. ОСНОВЫ РАБОТЫ В HTML. РАБОТА С ТЕКСТОМ.....	4
3.1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ К ПРАКТИЧЕСКОЙ РАБОТЕ №1.....	4
3.2. ЗАДАНИЕ НА ПРАКТИЧЕСКУЮ РАБОТУ №1.....	20
3.3. ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ И ПОДГОТОВКИ К ЗАЩИТЕ ПРАКТИЧЕСКОЙ РАБОТЫ №1.....	20
4. ПРАКТИЧЕСКАЯ РАБОТА №2. РАБОТА С ТАБЛИЦАМИ В HTML.....	21
4.1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ К ПРАКТИЧЕСКОЙ РАБОТЕ №2.....	21
4.2. ЗАДАНИЕ НА ПРАКТИЧЕСКУЮ РАБОТУ №2.....	32
4.3. ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ И ПОДГОТОВКИ К ЗАЩИТЕ ПРАКТИЧЕСКОЙ РАБОТЫ №2.....	33
5. ПРАКТИЧЕСКАЯ РАБОТА №3. ОСНОВЫ РАБОТЫ В HTML. РАБОТА СО СПИСКАМИ И ИЗОБРАЖЕНИЯМИ.....	33
5.1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ К ПРАКТИЧЕСКОЙ РАБОТЕ №3.....	33
5.2. ЗАДАНИЕ НА ПРАКТИЧЕСКУЮ РАБОТУ №3.....	43
5.3. ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ И ПОДГОТОВКИ К ЗАЩИТЕ ПРАКТИЧЕСКОЙ РАБОТЫ №3.....	44
6. ПРАКТИЧЕСКАЯ РАБОТА №4. ОСНОВЫ РАБОТЫ В HTML. РАБОТА С ГИПЕРССЫЛКАМИ.....	45
6.1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ К ПРАКТИЧЕСКОЙ РАБОТЕ №4.....	45
6.2. ЗАДАНИЕ НА ПРАКТИЧЕСКУЮ РАБОТУ №4.....	53
6.3. ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ И ПОДГОТОВКИ К ЗАЩИТЕ ПРАКТИЧЕСКОЙ РАБОТЫ №4.....	53
7. ПРАКТИЧЕСКАЯ РАБОТА №5. ОСНОВЫ РАБОТЫ В HTML. РАБОТА С ФОРМАМИ.....	54
7.1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ К ПРАКТИЧЕСКОЙ РАБОТЕ №5.....	54
7.2. ЗАДАНИЕ НА ПРАКТИЧЕСКУЮ РАБОТУ №5.....	71
7.3. ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ И ПОДГОТОВКИ К ЗАЩИТЕ ПРАКТИЧЕСКОЙ РАБОТЫ №5.....	71
8. ПРАКТИЧЕСКАЯ РАБОТА №6. ОСНОВЫ РАБОТЫ В HTML. ФРЕЙМЫ.....	72
8.1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ К ПРАКТИЧЕСКОЙ РАБОТЕ №6.....	72
8.2. ЗАДАНИЕ НА ПРАКТИЧЕСКУЮ РАБОТУ №6.....	78
8.3. ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ И ПОДГОТОВКИ К ЗАЩИТЕ ПРАКТИЧЕСКОЙ РАБОТЫ №6.....	79
9. ЛИТЕРАТУРА.....	79

1. ЦЕЛЬ И ЗАДАЧА УЧЕБНОГО ПРАКТИКУМА

Цель методического указания: формирование у обучающихся навыков разметки Web-страниц.

Задача методического указания: освоить и закрепить работу с объектами при разработке Интернет-ресурсов.

2. ЗАДАНИЕ И ТРЕБОВАНИЯ К СОДЕРЖАНИЮ ОТЧЕТА

Задание: изучить теоретические сведения, выполнить все примеры и задания по практике, продемонстрировать их преподавателю, оформить результаты выполнения в виде отчёта.

Содержание отчёта: сформулировать цель работы, теоретические сведения, привести текст заданий и примеров, скриншоты страниц разработанного ресурса, сделать выводы, привести список используемой литературы. Пример оформления титульного листа отчета - стандартный.

Необходимые приборы и оборудование:

- Компьютер, совместимый с IBM PC, 1 Гб ОЗУ и более.
- Операционная система WINDOWS *.

3. ПРАКТИЧЕСКАЯ РАБОТА № 1. ОСНОВЫ РАБОТЫ В HTML. РАБОТА С ТЕКСТОМ

3.1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ К ПРАКТИЧЕСКОЙ РАБОТЕ №1

HTML (*HyperText Markup Language*) - язык разметки гипертекста - предназначен для создания Web-страниц.

Под **гипертекстом** в этом случае понимается текст, связанный с другими текстами указателями-ссылками.

HTML представляет собой простой набор кодов, которые описывают структуру документа. HTML позволяет выделить в тексте отдельные логические части (заголовки, абзацы, списки и т.д.), поместить на Web-страницу подготовленную фотографию или картинку, организовать на странице ссылки для связи с другими документами.

HTML не задает конкретные и точные атрибуты форматирования документа. Конкретный вид документа окончательно определяет только **программа-браузер** на компьютере пользователя Интернета.

Основными компонентами HTML являются:

- **Тег (tag).** Тег HTML - это компонент, который определяет Web- браузеру выполнить определенную задачу типа создания абзаца или вставки изображения.
- **Атрибут (или аргумент).** Атрибут HTML изменяет тег. Например, можно выравнивать абзац или изображение внутри тега.
- **Значение.** Значения присваиваются атрибутам и определяют вносимые изменения. Например, если для тега работы с текстом используется атрибут выравнивания, то

можно указать значение этого атрибута. Значения могут быть типа *left, right, center* или *justify* (задается только через использование Каскадных Таблиц Стилей).

Теги представляют собой зарезервированные последовательности символов, начинающиеся с < (знака меньше) и заканчивающиеся > (знаком больше).

Закрытие тега отличается от открытия только наличием символа '/'.

HTML- программа должна начинаться тегом **<HTML>** и заканчиваться тегом **</HTML>**

```
<HTML>
```

```
..... (здесь будут другие теги программы)
```

```
</HTML>
```

HTML- программы состоят из двух основных частей: заголовка и тела. Заголовок ограничивается парой тегов **<HEAD>** и **</HEAD>**, а тело - парой тегов **<BODY>** и **</BODY>**.

В результате HTML- программа выглядит следующим образом:

```
<HTML>
```

```
  <HEAD>
```

```
    ... (здесь будет заголовок)
```

```
  </HEAD>
```

```
  <BODY>
```

```
    .... (здесь будут другие теги тела программы)
```

```
  </BODY>
```

```
</HTML>
```

Кроме того, каждая HTML- программа имеет заголовок, который помещается в заголовок окна браузера. Заголовок окна браузера создается при помощи двух тегов **<TITLE>** и **</TITLE>** и содержится между тегами **<HEAD>** и **</HEAD>**. Тогда программа принимает следующий вид:

```
<HTML>
```

```
  <HEAD>
```

```
    <TITLE> Основы HTML </TITLE>
```

```
  </HEAD>
```

```
  <BODY>
```

```
    .... (здесь будут другие теги тела программы)
```

```
  </BODY>
```

```
</HTML>
```

На рисунке 1 приведена простейшая структура разметки Web-страницы.

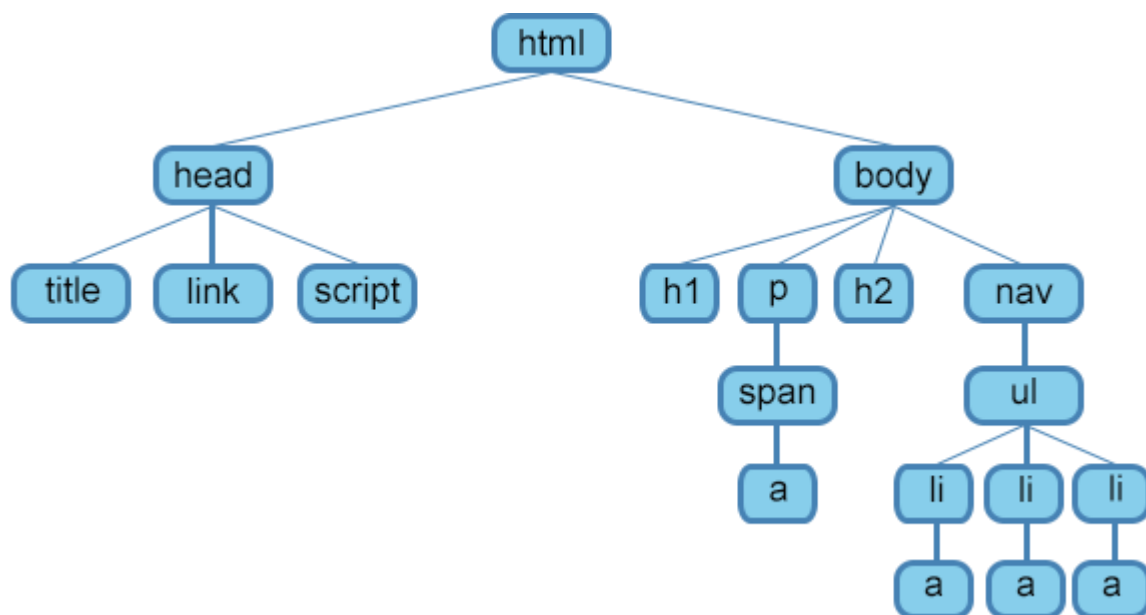


Рис.1. Простейшая структура Web-страницы.

Общая структура HTML-файла

Таким образом, общая структура HTML-файла:

```

<HTML>
<HEAD>
<Мета-теги>
<TITLE>Заголовок документа</TITLE>
</HEAD>
<BODY>
Основная часть документа
</BODY>
</HTML>
  
```

Немного расширим описание HTML-документа.

Тег <DOCTYPE> отвечает за корректное отображение веб-страницы браузером. DOCTYPE определяет не только версию HTML (например, html), но и соответствующий DTD-файл в Интернете.

```

<!DOCTYPE html> <!-- Объявление формата документа -->
<html>
<head> <!-- Техническая информация о документе -->
<meta charset="UTF-8"> <!-- Определяем кодировку символов документа -->
<title>...</title> <!-- Задаем заголовок документа -->
<link rel="stylesheet" type="text/css" href="style.css"> <!-- Подключаем внешнюю
таблицу стилей -->
<script src="script.js"></script> <!-- Подключаем сценарии -->
</head>
<body> <!-- Основная часть документа -->
</body>
  
```

`</html>`

Элементы, находящиеся внутри тега `<html>`, образуют дерево документа, так называемую объектную модель документа, DOM (document object model). При этом элемент `<html>` является корневым элементом.

Чтобы разобраться во взаимодействии элементов веб-страницы, необходимо рассмотреть так называемые «родственные отношения» между элементами. Отношения между множественными вложенными элементами подразделяются на родительские, дочерние и сестринские.

Предок — элемент, который включает в себе другие элементы. На рисунке 1 предком для всех элементов является `<html>`. В то же время элемент `<body>` является предком для всех содержащихся в нем тегов: `<h1>`, `<p>`, ``, `<nav>` и т.д.

Потомок — элемент, расположенный внутри одного или более типов элементов. Например, `<body>` является потомком `<html>`, а элемент `<p>` является потомком одновременно для `<body>` и `<html>`.

Родительский элемент — элемент, связанный с другими элементами более низкого уровня, и находящийся на дереве выше их. На рисунке 1 `<html>` является родительским только для `<head>` и `<body>`. Тег `<p>` является родительским только для ``.

Дочерний элемент — элемент, непосредственно подчиненный другому элементу более высокого уровня. На рисунке 1 только элементы `<h1>`, `<h2>`, `<p>` и `<nav>` являются дочерними по отношению к `<body>`.

Сестринский элемент — элемент, имеющий общий родительский элемент с рассматриваемым, так называемые элементы одного уровня. На рисунке 1 `<head>` и `<body>` — элементы одного уровня, так же как и элементы `<h1>`, `<h2>` и `<p>` являются между собой сестринскими.

Ниже будут описаны основные теги работы с текстом, например, такие как теги заголовков, теги форматирования текста, собственно тег `font`, средства работы с абзацами, теги оформительского характера и многое другое.

Элемент `<html>`

Является корневым элементом документа. Все остальные элементы содержатся внутри тегов `<html>...</html>`. Все, что находится за пределами тегов, не воспринимается браузером как код HTML и никак им не обрабатывается.

Элемент `<head>`

Раздел `<head>...</head>` содержит техническую информацию о странице: заголовок, описание, ключевые слова для поисковых машин, кодировку и т.д. Введенная в нем информация не отображается в окне браузера, однако содержит данные, которые указывают браузеру, как следует обрабатывать страницу.

Элемент `<title>`

Обязательным тегом раздела <head> является тег <title>. Текст, размещенный внутри этого тега, отображается в строке заголовка веб-браузера. Длина заголовка должна быть не более 60 символов, чтобы полностью поместиться в заголовке. Текст заголовка должен содержать максимально полное описание содержимого веб-страницы.

Элемент <meta>

Необязательным тегом раздела <head> является одинарный тег <meta>. С его помощью можно задать описание содержимого страницы и ключевые слова для поисковых машин, автора HTML-документа и прочие свойства метаданных. Элемент <head> может содержать несколько элементов <meta>, потому что в зависимости от используемых атрибутов они несут различную информацию.

```
<meta name="description" content="Описание содержимого страницы">
```

```
<meta name="keywords" content="Ключевые слова через запятую">
```

Описание содержимого страницы и ключевые слова одновременно можно указывать на нескольких языках, например, на русском и английском:

```
<meta name="description" lang="ru" content="Описание содержимого страницы">
```

```
<meta name="description" lang="en" content="Description">
```

```
<meta name="keywords" lang="ru" content="Ключевые слова через запятую">
```

```
<meta name="keywords" lang="en" content="Keywords">
```

С помощью тега <meta> можно запретить или разрешить индексацию веб-страницы поисковыми машинами:

Индексация и переход по ссылкам разрешены:

```
<meta name="robots" content="index, follow">
```

Индексация разрешена, переход по ссылкам запрещен:

```
<meta name="robots" content="index, nofollow">
```

Индексация и переход по ссылкам запрещены:


```
<meta name="robots" content="noindex, nofollow">
```

Для автоматической перезагрузки страницы через заданный промежуток времени нужно воспользоваться значением refresh:

```
<meta http-equiv="refresh" content="30">
```

Страница будет перезагружена через 30 секунд. Чтобы перебросить посетителя на другую страницу, нужно указать URL-адрес в параметре url:

```
<meta http-equiv="refresh" content="0; url=http://yandex.ru/">
```

АТТРИБУТЫ ТЕГА <META>	
Атрибут	Описание, принимаемое значение
charset	Указывает кодировку символов для текущего HTML-документа: <code><meta charset="UTF-8"></code>
content	Содержит произвольный текст, который определяет значение, ассоциируемое с атрибутом http-equiv или name, в зависимости от их значения.
http-equiv	<p>Контролирует действия браузера на данной веб-странице (эквивалент HTTP заголовков). При отображении страницы браузер будет следовать инструкциям, заданным в атрибуте:</p> <p>default-style указывает предпочтительный стиль для использования на странице. Атрибут content должен содержать идентификатор элемента <code><link></code>, который ссылается на таблицу стилей CSS, или идентификатор элемента <code><style></code>, содержащего таблицу стилей.</p> <p>refresh указывает время в секундах до перезагрузки страницы или время до перенаправления на другую страницу, если в атрибуте content после указания времени идет строка "url=адрес_страницы".</p> <p>Автоматическая перезагрузка страницы через заданный промежуток времени, в данном примере, через 30 секунд:</p> <pre><meta http-equiv="refresh" content="30"></pre> <p>Если необходимо сразу перебросить посетителя на другую страницу, то можно указать URL-адрес в параметре url:</p> <pre><meta http-equiv="refresh" content="0; url=http://mail.ru/"></pre>
name	Ассоциируется со значением, содержащимся в атрибуте content. Не должен использоваться в случае, если для элемента уже заданы атрибуты http-equiv, charset или itemprop.

	<p>application-name указывает название веб-приложения, используемого на странице.</p> <p>author указывает имя автора документа в свободном формате.</p> <p>description определяет краткое описание к содержимому страницы, например:</p> <pre><meta name="description" content="Описание содержимого страницы"></pre> <p>generator указывает один из пакетов программного обеспечения, используемого для создания документа, например:</p> <pre><meta name="generator" content="WordPress 4.0"></pre> <p>keywords содержит список ключевых слов, разделенных запятыми, соответствующих содержимому страницы, например:</p> <pre><meta name="keywords" content="Ключевые слова через запятую"></pre> <p>Также атрибут name может принимать следующие значения из расширенной спецификации, такие как creator, googlebot, publisher, robots, slurp, viewport, хотя ни одно из них еще не было официально принято.</p>
--	---

Элемент <style>

Внутри этого элемента задаются стили, которые используются на странице. Для задания стилей в HTML-документе используется язык CSS. Таких элементов на странице может быть несколько.

Для элемента доступны атрибуты media, scoped, type, а также глобальные атрибуты.

Внутри этого элемента можно записывать код форматирования как самих элементов веб-страницы, так и веб-страницы целиком.

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

.paper {

width: 200px;

height: 300px;

background-color: #ef4444;

color: #666666;

}
```

Чтобы подключить к элементу заданный стиль, необходимо через атрибут class(или id) присвоить элементу соответствующее название:

```
<div class="paper">
```

...

</div>

CSS-код можно встраивать непосредственно в элемент разметки в виде значения атрибута style, например:

<p style="color: #666666; background-color: #ef4444; padding: 20px;">

АТТРИБУТЫ ТЕГА <STYLE>	
Атрибут	Описание, принимаемое значение
media	Определяет, для какого типа устройства предназначены те или иные каскадные таблицы стилей. Внутри таблицы стилей CSS специфичные правила для определенных устройств прописываются с помощью media query.
nonce	Генерируемая случайным образом на сервере строковая переменная, которая устанавливает правила использования встроенных стилей с целью защиты контента. Значение атрибута — строка текста.
type	Определяет MIME-тип подключаемого файла со стилями, в данном случае <style type="text/css">.

Теги заголовков

Заголовок первого уровня

Заголовок второго уровня

Заголовок третьего уровня

Заголовок четвертого уровня

Заголовок пятого уровня

Заголовок шестого уровня

Заголовки являются важными элементами веб-страницы, они упорядочивают текст, формируя его визуальную структуру. Теги <h1>...<h6> должны использоваться только для выделения заголовков нового раздела или подраздела. При использовании заголовков

необходимо учитывать их иерархию, т.е. за <h1> должен следовать <h2> и т.д. Также не допускается вложение других тегов в теги <h1>...<h6>.

Тег <h1>

Заголовок самого верхнего уровня, на странице рекомендуется использовать только один раз, по возможности частично дублируя заглавие страницы. Тег <h1> должен быть уникальным для каждой страницы сайта. Рекомендуется прописывать тег в начале статьи, используя ключевое слово в тексте заголовка. Размер шрифта в браузере равен 2em, верхний и нижний отступ по умолчанию 0.67em.

Тег <h2>

Им обозначаются подзаголовки тега <h1>. Размер шрифта в браузере равен 1.5em, верхний и нижний отступ по умолчанию 0.83em.

Тег <h3>

Показывает подзаголовки тега <h2>. Размер шрифта в браузере равен 1.17em, верхний и нижний отступ по умолчанию 1em.

Теги <h4>, <h5>, <h6>

Обозначают подзаголовки четвертого, пятого и шестого уровня. Размер шрифта в браузере равен 1em / 0.83em / 0.67em, верхний и нижний отступ по умолчанию 1.33em / 1.67em / 2.33em соответственно.

Для многих тегов доступны глобальные атрибуты, применимые для других тегов HTML.

Глобальные атрибуты

Глобальные атрибуты, приведенные в таблице ниже, могут быть использованы для любого HTML-элемента, хотя некоторые из них могут не оказывать на элементы никакого влияния.

ГЛОБАЛЬНЫЕ АТТРИБУТЫ HTML	
Атрибут	Описание, принимаемое значение
accesskey	Генерирует сочетания клавиш для доступа к текущему элементу. Состоит из разделенного пробелами списка символов. Браузер в первую очередь выбирает те клавиши, которые существуют на раскладке клавиатуры. Применяется к следующим элементам: <a>, <area>, <button>, <input>, <label>, <legend>, <textarea>. Принимаемые значения: перечень названий клавиш.
class	Определяет имя класса для элемента (используется для определения класса в таблице стилей). Принимаемые значения: имя класса.
contenteditable	Определяет, может ли пользователь редактировать содержимое (контент). Позволяет преобразовать любое поле HTML в

	редактируемый элемент. Принимаемые значения: true/false.
contextmenu	Добавляет к элементу контекстное меню, заданное тегом <menu>. Принимаемые значения: значение атрибута id элемента <menu>.
dir	Определяет направление текста контента в элементах <bdo> и <bdi>. Принимаемые значения: ltr/rtl/auto.
draggable	Определяет, может ли пользователь перетащить элемент. Принимаемые значения: true/false/auto.
dropzone	Определяет область для приема перемещаемых элементов, сообщая браузеру пользователя, какие действия совершить при перемещении. Принимаемые значения: copy — содержимое перемещаемого элемента будет скопировано в область. move — содержимое перемещаемого элемента будет перемещено в новую область. link — при перемещении будет создана ссылка на первоначальные данные элемента.
hidden	Указывает на то, что элемент должен быть скрыт. Принимаемые значения: hidden.
id	Определяет уникальный идентификатор элемента. Принимаемые значения: id — идентификатор элемента.
lang	Определяет код языка содержимого (контента) в элементе. Принимаемые значения: код языка.
spellcheck	Указывает, подлежит ли содержимое элемента проверке орфографии и грамматики. Принимаемые значения: true/false.
style	Указывает на код CSS, применяемую для оформления элемента. Принимаемые значения: код CSS.
tabindex	Определяет порядок перехода к элементу при помощи клавиши TAB. Принимаемые значения: порядковый номер.
title	Определяет дополнительную информацию об элементе, задавая всплывающую подсказку для страницы. Принимаемые значения: текст.
translate	Разрешает или запрещает перевод текста внутри элемента. Принимаемые значения: yes/no.

Теги для форматирования текста

Тег

Задаёт полужирное начертания шрифта. Выделяет текст без акцента на его важность.

Тег

Отображает шрифт курсивом, придавая тексту значимость.

Тег <i>

Отображает шрифт курсивом.

Тег <small>

Уменьшает размер шрифта на единицу по отношению к обычному тексту.

**Тег **

Задаёт полужирное начертание шрифта, относится к тегам логической разметки, указывая браузеру на важность текста.

Тег <sub>

Используется для создания нижних индексов. Сдвигает текст ниже уровня строки, уменьшая его размер.

Тег <sup>

Используется для создания степеней. Сдвигает текст выше уровня строки, уменьшая его размер (рис.2).

Например,

```
<html>
```

```
<body>
```

```
<hr align="center" size="5" width="50%" color="#3399ff">
```

```
<hr align="left" size="5" width="50%" color="lime">
```

```
<hr align="center" size="5" width="50%" color="red">
```

```
<hr align="right" size="5" width="50%" color="000000">
```

```
<table bgcolor="yellow">
```

```
<tr><td>
```

Н₂О – формула воды

Мой адрес: 1 <u>^я</u> Калужская улица

</td></tr>

</table>

</body>

</html>

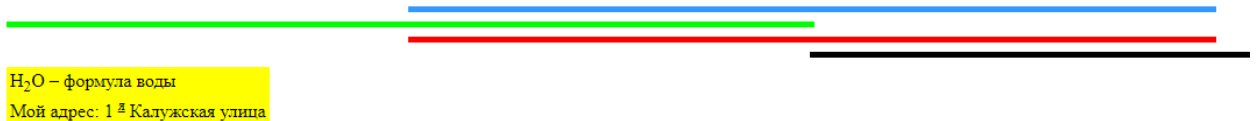


Рис. 2. Пример применения тегов верхнего и нижнего подиндекса

Тег <u>

Используется для подчеркивания текста.

Тег <ins>

Выделяет текст в новой версии документа, подчёркивая его.
Для тега доступны следующие атрибуты: cite, datetime.

Тег

Перечёркивает текст. Используется для выделения текста, удаленного из документа.
Для тега доступны следующие атрибуты: cite, datetime.

Теги для ввода «компьютерного» текста

Тег <code>

Служит для выделения фрагментов программного кода. Отображается моноширинным шрифтом.
Для тега доступны глобальные атрибуты.

Тег <kbd>

Отмечает фрагмент как вводимый пользователем с клавиатуры. Отображается моноширинным шрифтом.
Для тега доступны глобальные атрибуты.

Тег <samp>

Применяется для выделения результата, полученного в ходе выполнения программы.
Отображается моноширинным шрифтом.

Тег <var>

Выделяет имена переменных, отображая курсивом.

Тег <pre>

Позволяет вывести текст на экран, сохранив изначальное форматирование. Пробелы и переносы строк при этом не удаляются.

Теги для оформления цитат и определений

Тег <abbr>

Применяется для форматирования аббревиатур. Браузером обычно подчеркивается пунктирной линией. Расшифровка сокращения осуществляется с помощью атрибута title, она появляется при наведении курсора мыши на текст.

Тег <bdo>

Используется для замещения текущего направления текста, т.е. текст в теге отображается зеркально.

Для тега доступен атрибут dir.

Тег <blockquote>

Выделяет цитаты внутри документа, выделяя его отступами и переносами строк.

Для тега доступен атрибут cite.

Тег <q>

Используется для выделения коротких цитат. Браузерами заключается в кавычки.

Для тега доступен атрибут cite.

Тег <cite>

Применяется для выделения цитат, названий произведений, сносок на другие документы.

Тег <dfn>

Позволяет выделить текст как определение. Несмотря на наличие специального тега, рекомендуется выделять текст силами CSS.

Для тега доступен атрибут title.

Абзацы, средства переноса текста

Тег <p>

Понятие абзаца в HTML-документе также аналогично понятию абзаца в Microsoft Word. Абзац обозначается в документе парными тегами <P> и </P> . Специфика тега

тега **<P>** заключается в том, что после текста, который находится в его пределах, пустая строка добавляется *автоматически*.

Браузер автоматически добавляет верхний и нижний отступ, равный 1em.

Следует помнить и о другой особенности текстовых абзацев: когда текст достигает правой границы окна Web-браузера переход на новую строку осуществляется автоматически, независимо от расположения тега **<P>**.

Для отдельного абзаца можно указать тип, размер и цвет шрифта отличный от стиля остального документа.

Например:

```
<P ALIGN="CENTER"><B><I><FONT FACE="Arial" SIZE=6  
COLOR="#C07080">Мои первые шаги в HTML</P>
```

Атрибуты:

align - Выравнивает параграф относительно одной из сторон документа.

left - выравнивание по правому краю (По умолчанию).

right - выравнивание по правому краю.

center - выравнивание по центру.

justify - выравнивание по ширине.

Тег

Переносит текст на следующую строку, создавая разрыв строки.

Тег <hr>

Используется для разделения контента на веб-странице (рис.3). Отображается в виде горизонтальной линии.

Тег **<hr>** имеет следующие атрибуты:

size — толщина полосы;

width — ширина полосы;

align — выравнивание. Выравнивает заголовок в соответствии со следующими значениями:

- **center** - по центру.
- **left** - по левому краю.
- **right** - по правому краю.

color — цвет полосы;

noshade — атрибут, не имеющий значений. Если атрибут указан, то создается сплошная чёрная полоса без тени.

Например:

```
<html>  
<body>  
<hr align="center" size="5" width="50%" color="#3399ff">  
<hr align="left" size="5" width="50%" color="lime">  
<hr align="center" size="5" width="50%" color="red">  
<hr align="right" size="5" width="50%" color="000000">  
</body>  
</html>
```

Рис.3. Пример применения тега <hr> с различными атрибутами

Определяющий тег для работы с текстом

Тег определяет цвет, размер и выводимый шрифт.
Закрывающий тег обязателен.

Атрибуты тега:

color - определяет цвет текста. Цвет может быть задан одним из нижеуказанных способов:

- Явное указание цвета текста. Например,
- Используя шестнадцатичное значение цвета. Например,
- С указанием используемой цветовой модели. Например,

face - определяет гарнитуру шрифта. Например,

size — размер текста шрифта. Может быть задан абсолютным (от 1 до 6) либо относительным значением (от +1 до +6 и от -1 до -6). По умолчанию, размер текста равен 3. Если указать < font size="+1">< /font>, то текст, размещённый между тегами < font> и < /font>, будет отображаться увеличенным на 1 по сравнению с другим текстом. Аналогично и с -1.

Пример:

.

.

Тег бегущей строки <marquee>

Тег <marquee> заставляет текст (как и любой элемент Web-страницы) перемещаться из стороны в сторону.

Закрывающий тег </marquee> обязателен.

Атрибуты:

behavior - определяет вид движения.

- **alternate** - Колебательные движения налево и направо.
- **scroll** - Перемещение текста в направлении, указанном в direction. Достигнув края экрана, надпись появляется снова с противоположной стороны.
- **slide** - Схоже с scroll, но текст перемещается только один раз и останавливается.

direction - определяет направление движения.

- **down** - Движение вниз.
- **left** - Движение справа налево (по умолчанию).
- **right** - Движение слева направо.
- **up** - Движение вверх.

[перейти к содержанию](#)

ПРИМЕР ПРИМЕНЕНИЯ HTML-ТЕГОВ

```
<HTML>
<HEAD>
<TITLE>Изучаем теги HTML</TITLE>
</HEAD>
<BODY>
<CENTER>
<U><SMALL>СТИХОТВОРЕНИЕ</SMALL></U>
  <h1>Первый снежок</h1>
</CENTER>
<FONT COLOR=navy SIZE=4 FACE=Arial>
  <P align=left>
    <u>Рано зима наступила,<BR>
    Замерзли цветы во дворах,<BR>
    И первый снежок, легкий, легкий...<BR>
    Улегся на наших плечах...<BR>

    <P align=center>
      А как ребяшня была рада<BR>
      Первому в зиму снежку!!!<BR>
      Пытались лепить они бабу,<BR>
      Крепили морковь на носу...<BR>

    <P align=right>
      Я тоже во двор тогда вышла,<BR>
      Позвала щенка я с собой...<BR>
      Сметали снежок мы и с вишни,<BR>
      И с яблони варежкой той...<BR>
  </FONT>
  <HR>
<h3>... Продолжение у автора...</h3>

<ADDRESS>Н.А. Борсук</ADDRESS>
</BODY>
</HTML>
```

Пример реализации приведенного выше кода приведен на рисунке 4.

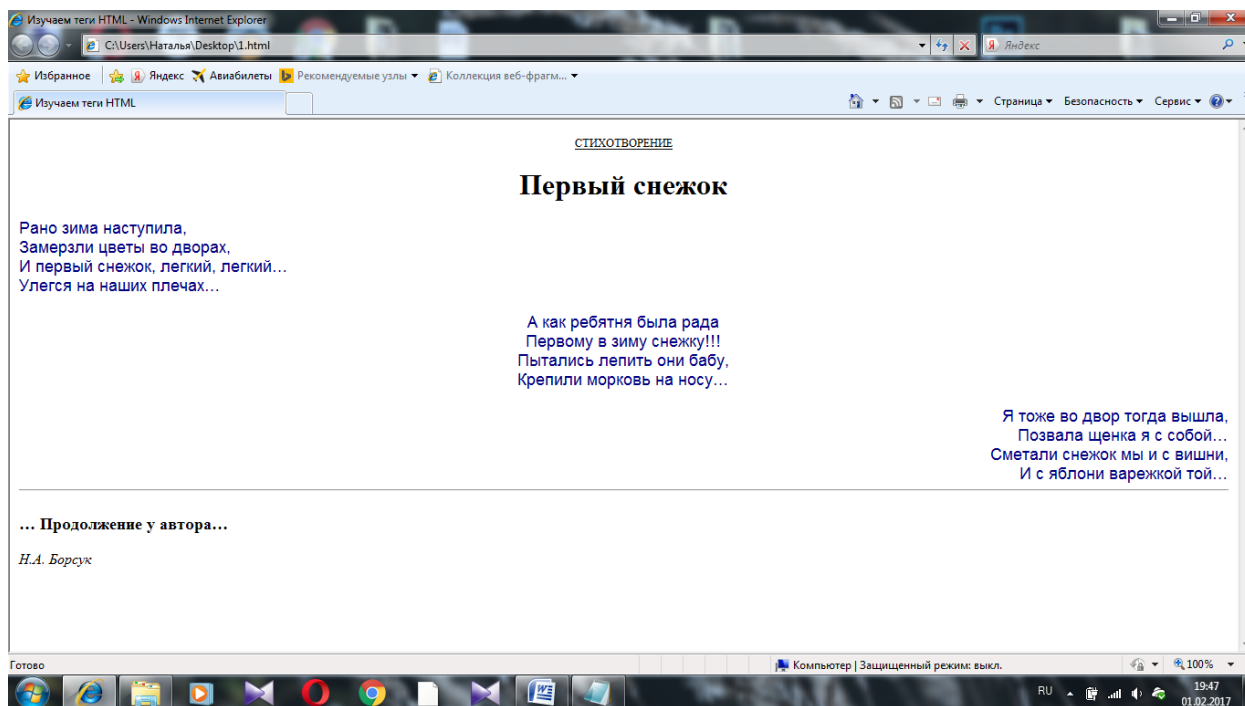


Рис.4. Пример работы с текстом в HTML

3.2. ЗАДАНИЕ НА ПРАКТИЧЕСКУЮ РАБОТУ №1

Написать Web-страницу, используя теги работы с текстом. Необходимо:

1. Проработать различные атрибуты тега FONT;
2. Декорировать страницу, используя возможности тега HR;
3. Использовать на странице различные теги форматирования текста;
4. С помощью тега MARQUEE анимировать текст на Web-странице, показать различные варианты атрибута direction;

3.3. ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ И ПОДГОТОВКИ К ЗАЩИТЕ ПРАКТИЧЕСКОЙ РАБОТЫ №1

1. Опишите атрибуты тега FONT;
2. Продемонстрируйте работу с тегом MARQUEE на Web-странице;
3. Покажите возможности декорирования Интернет-страницы с использованием тега HR;
4. Продемонстрируйте различные способы начертания текста.

[перейти к содержанию](#)

4. ПРАКТИЧЕСКАЯ РАБОТА №2. РАБОТА С ТАБЛИЦАМИ В HTML

4.1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ К ПРАКТИЧЕСКОЙ РАБОТЕ №2

Благодаря универсальности таблиц, большому числу параметров, управляющих их видом, таблицы надолго стали определенным стандартом для верстки Web-страниц. Таблица с невидимой границей представляет собой аналог модульной сетки, в блоках которой удобно размещать элементы Web-страницы.

Таблица состоит из строк и столбцов ячеек, которые могут содержать текст, списки, изображения, бегущие строки с ограниченными размерами ячейки пространством, вложенные таблицы. То есть все возможные элементы Web-страниц могут являться содержимым ячеек таблиц.

Для добавления таблицы на Web-страницу используется тег `<table>`. Этот элемент служит контейнером для элементов, определяющих содержимое таблицы. Любая таблица состоит из строк и ячеек, которые задаются соответственно с помощью тегов `<tr>` и `<td>`. Таблица должна содержать хотя бы одну ячейку. Допускается вместо тега `<td>` использовать тег `<th>`. Текст в ячейке, оформленной с помощью тега `<th>`, отображается браузером шрифтом жирного начертания и выравнивается по центру ячейки. В остальном, разницы между ячейками, созданными через теги `<td>` и `<th>` нет.

Вместе эти теги пишутся так:

```
<table border=2 bordercolor=blue>
<tr>
<td>ячейка</td>
</tr>
</table>
```

Такая запись - это самая маленькая таблица, в ней всего одна строка, содержащая один столбец – ячейку.

ОСНОВНЫЕ ТЕГИ РАБОТЫ С ТАБЛИЦАМИ

Атрибуты тега `<table>`:

- `border` - толщина рамки;
- `bordercolor` - цвет рамки;
- `bordercolordark="цвет"; bordercolorlight="цвет"` – имитация 3D рамки.

Ниже приведен скриншот предыдущего html-кода с обозначенными у таблицы границами и цветовыми решениями (рис. 5).

```

<table border=2 bordercolor=blue>
<tr>
<td>ячейка</td>
</tr>
</table>

```

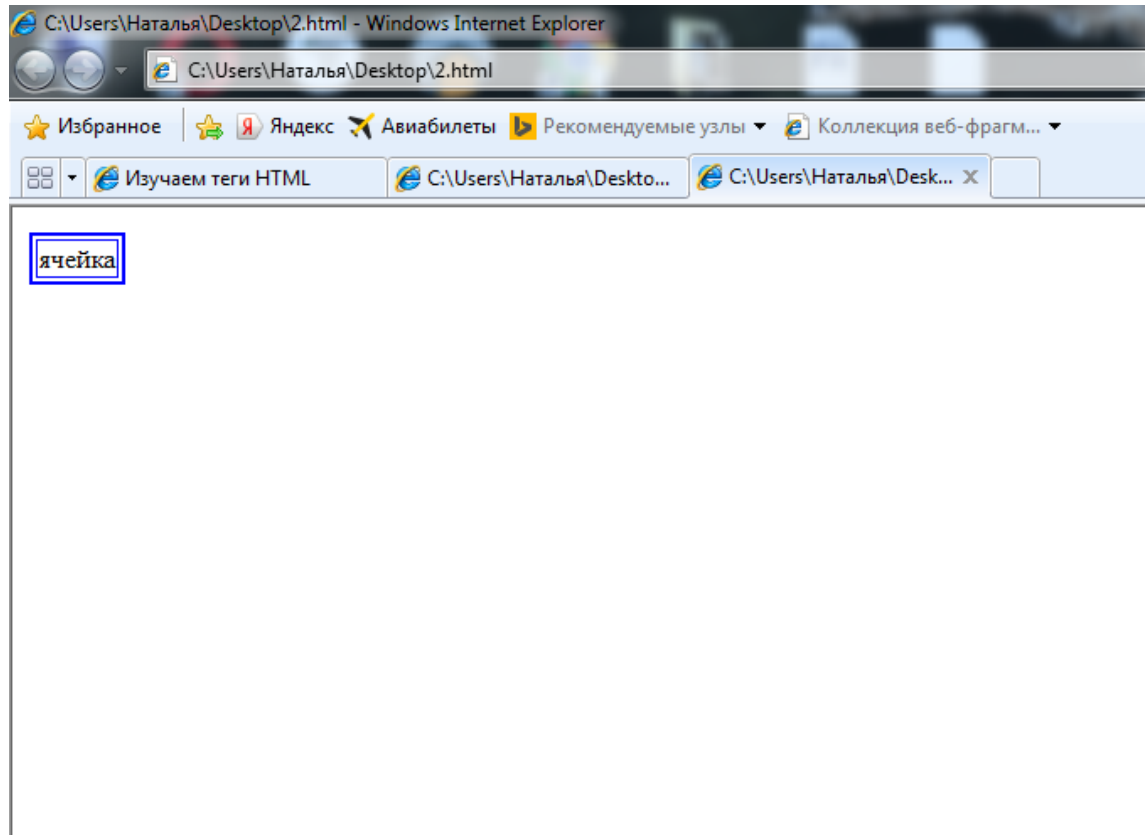


Рис. 5. Пример работы с тегом Table

Фон всей таблицы, отдельных строк или отдельных ячеек задаётся через атрибут bgcolor="цвет" или background="URL файла" в соответствующих тегах.

```

<table bgcolor="red">...</table>
<table background="1.jpg">

```

Пример:

```

<html><body>
  <table border=2 bgcolor=gray bordercolordark=yellow
  bordercolorlight=red >
    <tr> <th>Ф.И.О</th>
    <th>Возраст</th>
    </tr>
    <tr> <td>Иванов</td>
    <td>22</td>
    </tr>
    <tr> <td>Петров</td>
    <td>19</td>
    </tr>
  </table>
</body> </html>

```

Ниже приведен пример таблицы с использованием цветового оформления (рис.6)

Ф.И.О	Возраст
Иванов	22
Петров	19

Рис.6 . Пример работы с цветовым декорированием таблиц

Для выравнивания содержимого ячеек используются атрибуты:

Выравнивание по горизонтали:

Align="left" - слева

Align="center" – по центру

Align="right" - справа

Выравнивание по вертикали:

Valign="top" – сверху

Valign="middle" – посередине

Valign="bottom" – снизу

ОРГАНИЗАЦИЯ СЛОЖНОЙ СТРУКТУРЫ ТАБЛИЦ

Возможно объединение нескольких ячеек в одну строку:

<td rowspan="число строк">

Для объединения нескольких столбцов в одну ячейку:

<td colspan="число столбцов">

Для определения отступов содержимого ячейки от ее границ в тегах table, th, td, tr соответственно задаётся атрибут cellpadding="число пикселей".

Для задания отступов между ячейками прописывается атрибут cellspacing="число пикселей".

Для указания ширины таблицы используется атрибут [width](#), значение атрибута задаётся в пикселях или в процентах от ширины экрана

Для указания высоты – атрибут [height](#) (значение определяется аналогично вышесказанному).

Пример таблицы с приведенными выше атрибутами таблиц показан на рисунке 7, а ниже приведен ее код.

Личные данные		служебная информация
фамилия	возраст	

Рис. 7. Пример таблицы

```

<html>
<head><title> Таблица </title></head>
  <body>
    <table border=2 bordercolor="green" width=40%>
      <tr><th colspan=2> Личные данные</th>
        <th rowspan=2> служебная информация</th></tr>
      <tr><td>фамилия</td>
        <td>возраст</td>
      </tr>
      <tr><td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
      </tr>
      <tr><td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
      </tr>
      <tr bgcolor="green"> <td> &nbsp;</td>
        <td> &nbsp;</td>
        <td> &nbsp;</td>
      </tr>
    </table>
  </body>
</html>

```

На рисунке 8 приведен пример другой таблицы, а ниже показан код данного примера.

1. раз 2. два 3. три	
	1. четыре 2. пять 3. шесть

Рис. 8. Пример таблицы

```

<html>
<head><title> Таблица </title></head>
  <body text=blue>
    <table border=1 bordercolor="blue" cellspacing="4" align="center"
valign="middle">
      <tr><td><ol><li> раз </li>
        <li> два </li>
        <li> три </li>
      </ol>
      <td background="1.png">
      </td>
    </tr>
  </table>

```



```

        </tr>
        <tr>
            <td background="2.png">
            </td>
            <td><ol><li>четыре </li>
                <li>пять </li>
                <li>шесть </li>
            </ol>
            </td>
        </tr>
    </table>
</body>

</html>

```

На рисунке 9 приведен пример работы с размерами ячеек таблицы, а ниже дан код примера.

строка1 ячейка1	строка1 ячейка2	строка1 ячейка3
строка2 ячейка1	строка2 ячейка2	строка2 ячейка3
строка3 ячейка1	строка3 ячейка2	строка3 ячейка3

Рис. 9. Пример работы с размерами таблицы.

```

<html>
<head>
<title>Таблица</title>

```

```

</head>
<body>
<center>
<table border="1" width="640" height="480">
<tr height="25%">
<td width="15%"> строка1 ячейка1</td>
<td width="25%">строка1 ячейка2</td>
<td width="60%">строка1 ячейка3</td>
</tr>
<tr height="50%">
<td>строка2 ячейка1</td>
<td>строка2 ячейка2</td>
<td>строка2 ячейка3</td>
</tr>
<tr height="25%">
<td>строка3 ячейка1</td>
<td>строка3 ячейка2</td>
<td>строка3 ячейка3</td>
</tr>
</table>
</center>
</body>
</html>

```

Выше было сказано, что содержимым ячейки таблицы может являться таблица. Для этого в тег `<td>` нужно вставить еще один тег `<table>` с соответствующим содержимым и закрыть сначала тег вложенной таблицы `</table>`, затем тег ячейки `</td>` и выполнять при необходимости дальнейшие действия (рис. 10).

Заголовок										
содержание:	<p>В одну ячейку вложена целая таблица!!</p> <table><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td><td>6</td></tr><tr><td>7</td><td>8</td><td>9</td></tr></table>	1	2	3	4	5	6	7	8	9
1	2	3								
4	5	6								
7	8	9								

Рис. 10. Пример вложенности таблиц

ГРУППИРОВКА СТРОК ТАБЛИЦЫ

Для группировки строк используются теги `<thead>`, `<tbody>` и `<tfoot>`. Данные теги являют собой емкость для строк таблицы (тег `<tr>`), позволяя тем самым группировать и логически ими управлять.

- Тег `<thead>` - (От Англ. table head - "голова" таблицы) Предназначен для группировки одной или нескольких верхних строк в таблице, делая тем самым "шапку" таблицы.
- Тег `<tbody>` - (От Англ. table body - "тело" таблицы), Как правило этим тегом группируют центральные строки в таблице с основным содержанием.
- Тег `<tfoot>` - (От Англ. table footer - нижний колонтитул таблицы) Определяет "подвал" таблицы, состоящий из одной или нескольких нижних строк.

Данные теги могут быть расположены только внутри таблицы - контейнера `<table>`.

Теги `<thead>` и `<tfoot>` допускается использовать только один раз для одной таблицы. На тег `<tbody>` данное правило не распространяется.

Структура таблицы при использовании всех трёх данных тегов должна выглядеть вот так:

```
<table>
<thead>
<tr>
<td>голова</td>
</tr>
</thead>
<tbody>
<tr>
<td> тело</td>
</tr>
</tbody>
<tfoot>
<tr>
<td>подвал</td>
</tr>
</tfoot>
</table>
```

То есть сверху всегда должен быть контейнер `<thead>`, потом `<tbody>` и `<tfoot>`. Данные теги не могут пересекаться и быть вложенными друг в друга. А вот тег `<tfoot>` можно поменять местами с `<tbody>`, и что примечательно, строки, взятые в данный тег, всё равно браузером будут отображаться внизу таблицы.

К данным тегам допускается применять атрибуты выравнивания текста в ячейках `align` и `valign`, указывать цвет фона строк, взятых в группу с помощью `bgcolor`, делать всплывающую подсказку с помощью `title`. На рис. 11 приведен пример работы в вышеуказанными тегами группировки строк. А ниже приведен код данного примера.

Энергетическая ценность продуктов питания:				
Продукт	Белки	Жиры	Углеводы	ккал
Хлеб ржаной	4,7	0,7	49,8	214
Молоко	2,8	3,2	4,7	58
Картофель	2	0,1	19,7	83
Свинина	11,4	49,3	9	489
Итого:	20,9	53,3	83,2	844

Рис. 11. Пример работы с логическим разбиением содержимого Web-страницы.

```

<html>
<head>
<title>Группировка строк</title>
</head>
<body>
<table border="1" width="400" cellpadding="3" cellspacing="0">
<caption><b>Энергетическая ценность продуктов питания:</b></caption>
<thead bgcolor = "#deb887" title="Шапка">
<tr>
<th>Продукт</th><th>Белки</th><th>Жиры</th><th>Углеводы</th><th>ккал</th>
</tr>
</thead>
<tbody align="left" bgcolor = "#faebd7" title="Энергетическая ценность продуктов
питания">
<tr>
<th>Хлеб ржаной</th><td>4,7</td><td>0,7</td><td>49,8</td><td>214</td>
</tr>
<tr>
<th>Молоко</th><td>2,8</td><td>3,2</td><td>4,7</td><td>58</td>
</tr>
<tr>
<th>Картофель</th><td>2</td><td>0,1</td><td>19,7</td><td>83</td>
</tr>
<tr>
<th>Свинина</th><td>11,4</td><td>49,3</td><td>9</td><td>489</td>
</tr>
</tbody>
<tfoot align="left" bgcolor = "#f5f5dc" title="Итого:">
<tr>
<th>Итого:</th><td>20,9</td><td>53,3</td><td>83,2</td><td>844</td>
</tr>
</tfoot>
</table>
</body>
</html>

```

Основное предназначение данных тегов - это логическая группировка строк в группы, а это, в свою очередь, облегчает работу с таблицей, например, через стили или скрипты.

Теперь не нужно определять тот же стиль для каждой строчки, а достаточно указать его для всей группы.

ГРУППИРОВКА СТОЛБЦОВ ТАБЛИЦЫ

Ниже приведен пример кода группировки колонок таблицы про энергетическую ценность продуктов питания. Таблица состоит из пяти столбцов, поэтому после тега <table> следует написать:

```
<col width="140" bgcolor="#deb887">
<col width="100" bgcolor="#faebd7">
<col width="100" bgcolor="#faebd7">
<col width="100" bgcolor="#faebd7">
<col width="60" bgcolor="#f5f5dc">
```

Для каждой колонки свой тег <col> с указанными параметрами, где первый тег - это описание левой колонки таблицы, второй тег группирует вторую колонку и далее по порядку.

Исходя из выше представленного кода, в таблице вторая, третья и четвертая колонки имеют одинаковые параметры width="100" bgcolor="#faebd7". Чтобы не прописывать каждую одинаковую колонку отдельно можно воспользоваться атрибутом span, который указывает какое количество последующих колонок имеет идентичные параметры. В этом случае запись приобретёт следующий вид:

```
<col width="140" bgcolor="#deb887">
<col span="3" width="100" bgcolor="#faebd7">
<col width="60" bgcolor="#f5f5dc">
```

Таким образом, было указано, что три колонки, начиная со второй, имеют одинаковую ширину и цвет фона.

Тег <colgroup> может служить емкостью для тегов <col>, конструкция такая:

```
<colgroup>
<col>
</colgroup>
```

Если к тегу <table> применить атрибут rules со значением groups (что будет описано ниже), эффект будет достигнут только при использовании <colgroup>.

Больше никакой разницы между этими тегами нет. На рисунке 12 приведен пример работы с тегом <colgroup>.

```
<html>
<head>
<title>colgroup</title>
</head>
<body>
<table border="1">
```

```

<colgroup bgcolor="#deb887" align="left">
<col span="3" width="20">
</colgroup>
<colgroup bgcolor="#faebd7" align="center">
<col span="2" width="40">
<col width="60">
</colgroup>
<colgroup bgcolor="#f5f5dc" align="right">
<col width="20">
<col width="30">
<col width="50">
</colgroup>
<tr>
<td>20</td><td>20</td><td>20</td><td>40</td><td>40</td><td>60</td><td>20</td><td>30</td><td>50</td>
</tr>
<tr>
<td>20</td><td>20</td><td>20</td><td>40</td><td>40</td><td>60</td><td>20</td><td>30</td><td>50</td>
</tr>
<tr>
<td>20</td><td>20</td><td>20</td><td>40</td><td>40</td><td>60</td><td>20</td><td>30</td><td>50</td>
</tr>
<tr>
<td>20</td><td>20</td><td>20</td><td>40</td><td>40</td><td>60</td><td>20</td><td>30</td><td>50</td>
</tr>
</table>
</body>
</html>

```

20	20	20	40	40	60	20	30	50
20	20	20	40	40	60	20	30	50
20	20	20	40	40	60	20	30	50
20	20	20	40	40	60	20	30	50

Рис. 12. Пример работы с тегом <colgroup>

В этом примере колонки таблицы поделены на три группы с помощью <colgroup>, в этих группах указаны цвет фона и способ выравнивания, а ширина колонок задаётся с помощью тегов <col>, которые находятся внутри контейнеров <colgroup>.

Как и в случае с группировкой строк, колонки группируются с основной целью дальнейшего воздействия на них через стили или скрипты.

ГРАНИЦЫ ТАБЛИЦЫ

У тега <table> есть еще два атрибута, отвечающие за видимость границ таблицы: frame и [rules](#).

Атрибут frame определяет рамку таблицы и может принимать следующие значения:

- void - не отображать границы вокруг таблицы.
- border - отображать границы вокруг таблицы (по умолчанию).
- hside - отображать только горизонтальные границы сверху и снизу.
- vside - отображать только вертикальные границы слева и справа.
- above - отображать только верхнюю границу таблицы.
- below - отображать только нижнюю границу таблицы.
- lhs - отображать только левую границу.
- rhs - отображать только правую границу.

Пример приведен на рисунке 13.

Продукт	Белки	Жиры	Углеводы	ккал
Хлеб ржаной	4,7	0,7	49,8	214
Молоко	2,8	3,2	4,7	58
Картофель	2	0,1	19,7	83
Свинина	11,4	49,3	9	489
Итого:	20,9	53,3	83,2	844

Рис. 13. Пример работы с рамками таблицы

```
<html>
<head>
<title>frame</title>
</head>
<body>
<table border="8" frame="hside" width="400" cellpadding="3" cellspacing="0">
<caption><b>Энергетическая ценность продуктов питания:</b></caption>
<thead bgcolor="#deb887" title="Шапка">
<tr>
<th>Продукт</th><th>Белки</th><th>Жиры</th><th>Углеводы</th><th>ккал</th>
</tr>
</thead>
<tbody align="left" bgcolor="#faebd7" title="Энергетическая ценность продуктов
питания">
<tr>
<th>Хлеб ржаной</th><td>4,7</td><td>0,7</td><td>49,8</td><td>214</td>
</tr>
<tr>
```

```

<th>Молоко</th><td>2,8</td><td>3,2</td><td>4,7</td><td>58</td>
</tr>
<tr>
<th>Картофель</th><td>2</td><td>0,1</td><td>19,7</td><td>83</td>
</tr>
<tr>
<th>Свинина</th><td>11,4</td><td>49,3</td><td>9</td><td>489</td>
</tr>
</tbody>
<tfoot align="left" bgcolor="#f5f5dc" title="Итого:">
<tr>
<th>Итого:</th><td>20,9</td><td>53,3</td><td>83,2</td><td>844</td>
</tr>
</tfoot>
</table>
</body>
</html>

```

Атрибут `rules` в отличие от `frame` указывает на то, как следует рисовать границы между ячейками таблицы.

Атрибут `rules` может принимать следующие значения:

- `all` - отображать все границы между ячейками.
- `none` - не отображать границы между ячейками.
- `rows` - отображать границы только между строками.
- `cols` - отображать границы только между столбцами.
- `groups` - отображать границы только между группами строк и группами столбцов образованными с помощью тегов: `<thead>`, `<tbody>`, `<tfoot>` и `<colgroup>`.

4.2. ЗАДАНИЕ НА ПРАКТИЧЕСКУЮ РАБОТУ №2

Написать Web-страницу, используя теги работы с таблицами. Необходимо:

1. Проработать различные атрибуты тега `TABLE` в плане задания ее размеров, объединения строк и столбцов и расположения на Web-странице;
2. Декорировать страницу, используя возможности атрибутов тега `TABLE`;
3. Вставить в качестве элементов страницы списки, изображения и вложенную таблицу;
4. Проработать уже изученные возможности работы с текстом при работе с тегом `TABLE`.

4.1. ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ И ПОДГОТОВКИ К ЗАЩИТЕ ПРАКТИЧЕСКОЙ РАБОТЫ №2

1. Опишите атрибуты тега TABLE;
2. Продемонстрируйте работу с возможностью создания сложных таблиц на Web-странице;
3. Создайте примеры таблиц с различным декорированием всей таблицы и отдельных ее ячеек;
4. Опишите способы горизонтального и вертикального выравнивания содержимого ячеек таблиц;
5. Продемонстрируйте возможность различного отображения границ таблицы, отдельных ее ячеек и отступов внутри ячеек таблицы.

[перейти к содержанию](#)

5. ПРАКТИЧЕСКАЯ РАБОТА №3.

ОСНОВЫ РАБОТЫ В HTML. РАБОТА СО СПИСКАМИ И ИЗОБРАЖЕНИЯМИ

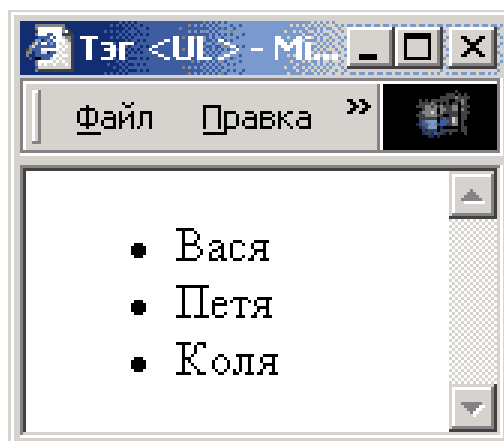
5.1 ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

МАРКИРОВАННЫЕ СПИСКИ

Каждый элемент маркированного списка имеет маркер — [специальный значок слева](#). Список ограничен парным тэгом (от английского *unordered list* — неупорядоченный список). Каждый элемент списка начинается с непарного тэга , после которого можно располагать текст, рисунки, таблицы и т.п. Список заканчивается закрывающим тэгом .

Ниже приведен пример простейшего маркированного списка — слева HTML-код, а справа — изображение в окне браузера.

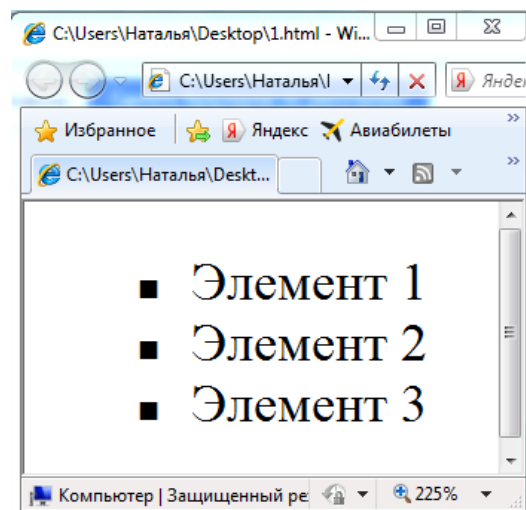
```
<BODY>  
<UL>  
<LI>Вася  
<LI>Петя  
<LI>Коля  
</UL>  
</BODY>
```



В команде **** можно задавать тип маркера, например, для того, чтобы установить в качестве маркера кружок, надо написать **<UL TYPE="disk">**

Кроме этого типа можно также использовать значения **circle** (окружность) и **square** (квадрат).

```
<BODY>
<UL TYPE="SQUARE">
<LI>Элемент 1
<LI>Элемент 2
<LI>Элемент 3
</UL>
</BODY>
```

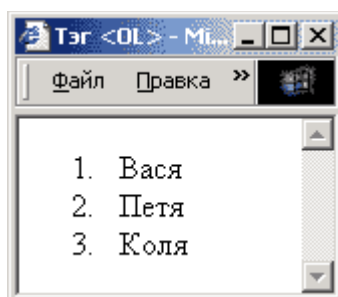


НУМЕРОВАННЫЕ СПИСКИ

Каждый элемент такого списка имеет номер. Нумерация может быть числовой или буквенной. Список ограничен парным тэгом **** (от английского *ordered list* — упорядоченный список). Каждый элемент списка начинается с непарного тэга ****, после которого можно располагать текст, рисунки, таблицы и т.п. Список заканчивается закрывающим тэгом ****.

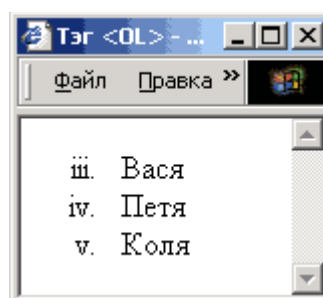
Ниже приведен пример простейшего нумерованного списка — слева **HTML**-код, а справа — изображение в окне браузера.

```
<BODY>
<OL>
<LI>Вася
<LI>Петя
<LI>Коля
</OL>
</BODY>
```



В команде **** можно задавать тип нумерации (в параметре **TYPE**) и начальное значение номера (в параметре **START**). Так в следующем примере задается нумерация маленькими римскими цифрами начиная с **iii**.

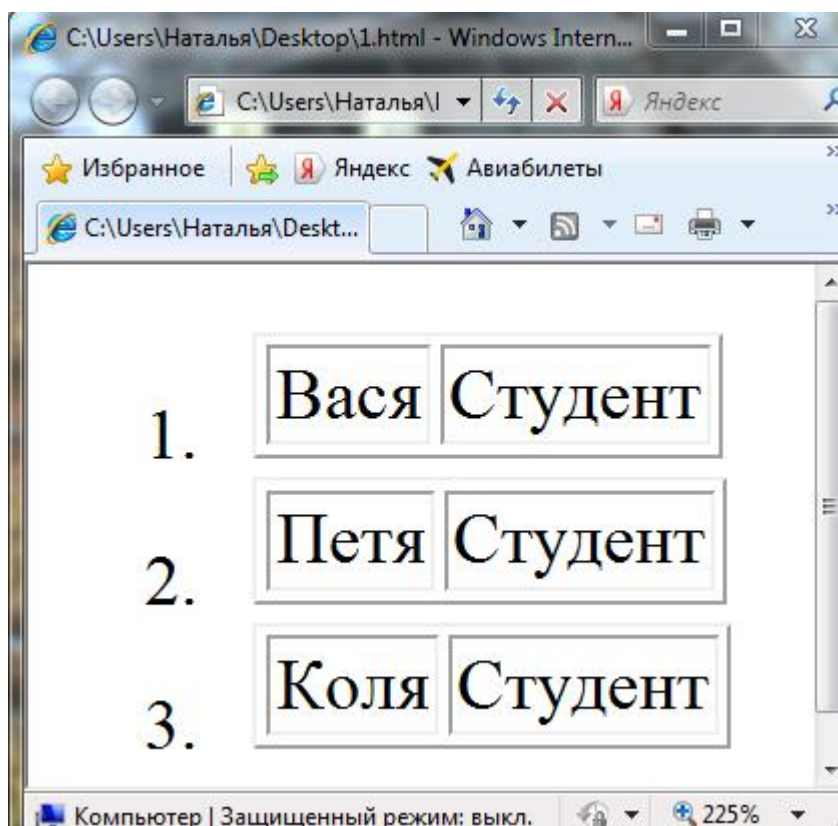
```
<BODY>
<OL TYPE="i" START="3">
<LI>Вася
<LI>Петя
<LI>Коля
</OL>
</BODY>
```



Кроме этого типа можно также использовать значения **A** (заглавные латинские буквы), **a** (строчные латинские буквы), **I** (большие римские цифры), и **1** (арабские цифры) — значение по умолчанию).

Ниже приведен пример маркированного списка, элементами которого являются таблицы. В этом случае список получает более сложный вид, а сам код усложняется.

```
<BODY>
<OL>
<LI>
<TABLE> <TR> <TD>
Вася </TD> <TD>
Студент </TD> </TR>
</TABLE>
<LI>
<TABLE> <TR> <TD>
Петя </TD> <TD>
Студент </TD> </TR>
</TABLE>
<LI>
<TABLE> <TR> <TD>
Коля </TD> <TD>
Студент </TD> </TR>
</TABLE>
</OL>
</BODY>
```



СПИСКИ ОПРЕДЕЛЕНИЙ

Так называются [списки-словари](#), в которых приводятся определения понятий и терминов. Каждый элемент такого списка содержит две записи — сам термин и его толкование. Нумерация отсутствует.

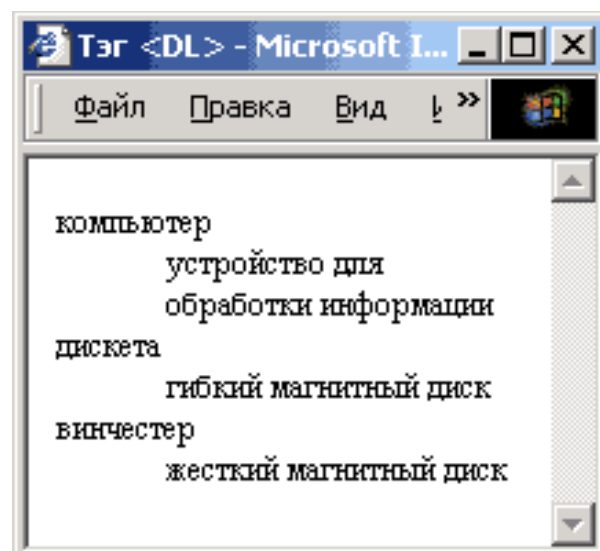
Список ограничен парным тэгом **<DL>** (от английского *definition list* — список определений). Каждый термин списка начинается с непарного тэга **<DT>** (от английского *definition term* — определяемый термин), после которого записывают его название. Вслед за термином записывают его определение, которое начинается с непарного тэга **<DD>** (от английского *definition description* — описание определения). Список заканчивается закрывающим тэгом **</DL>**.

Ниже приведен пример простейшего списка определений — слева **HTML**-код, а справа — изображение в окне браузера.

```

<BODY>
<DL>
<DT>компьютер
<DD>устройство      для
обработки информации
<DT>ОЗУ
<DD>оперативное
запоминающее устройство
<DT>винчестер
<DD>жесткий магнитный
диск
</DL>
</BODY>

```



ВЛОЖЕННЫЕ СПИСКИ

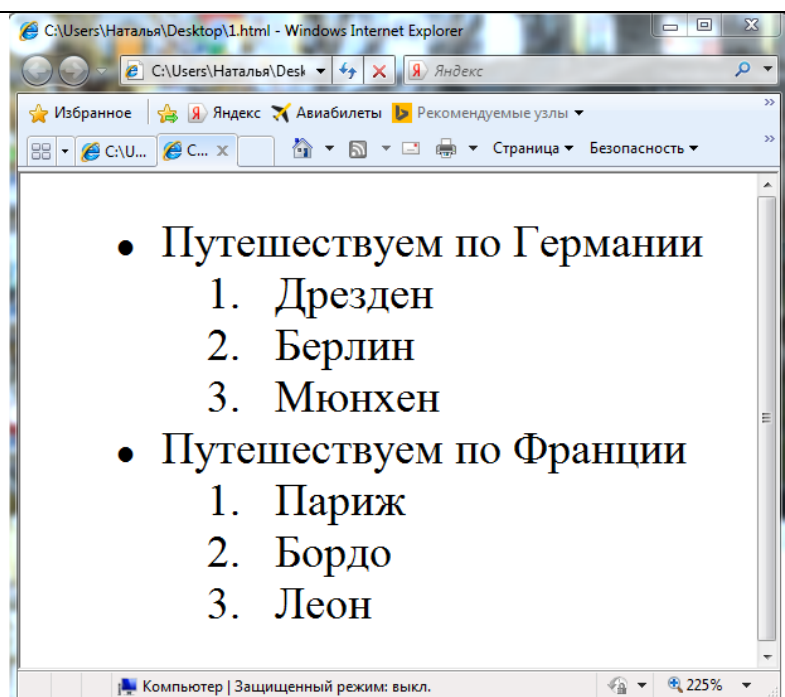
Списки могут быть вложенными, то есть, внутри элемента списка может быть еще один список (любого типа). На каждом уровне могут быть списки любых типов (нумерованные, маркированные, определения).

Ниже приведен пример простейшего вложенного списка — слева **HTML**-код, а справа — изображение в окне браузера.

```

<BODY>
<UL>
<LI>Путешествуем по
Германии
<OL>
<LI>Дрезден
<LI>Берлин
<LI>Мюнхен
</OL>
<LI>Путешествуем по
Франции
<OL>
<LI>Париж
<LI>Бордо
<LI>Леон
</OL>
</UL>

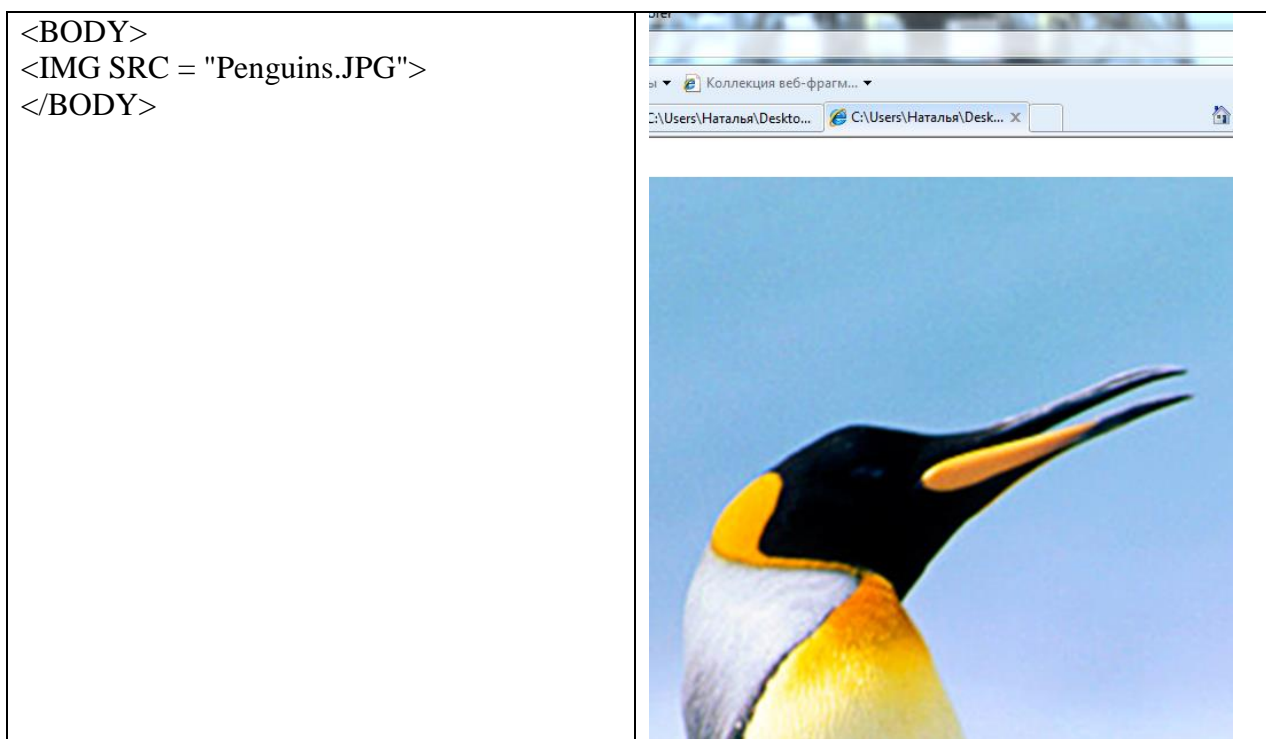
```



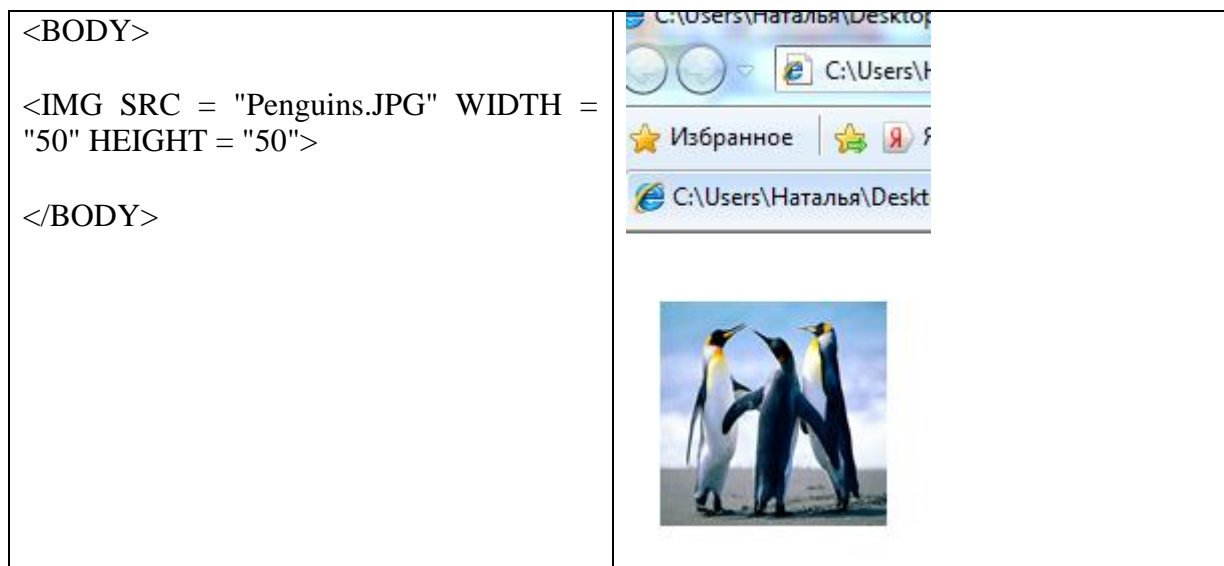
ИЗОБРАЖЕНИЯ

Тег `` предназначен для отображения на веб-странице изображений в графическом формате GIF, JPEG или PNG. Адрес файла с картинкой задаётся через атрибут `src`. Если необходимо, то рисунок можно сделать ссылкой на другой файл, поместив тег `` в контейнер `<a>`. При этом вокруг изображения отображается рамка, которую можно убрать, добавив атрибут `border="0"` в тег ``.

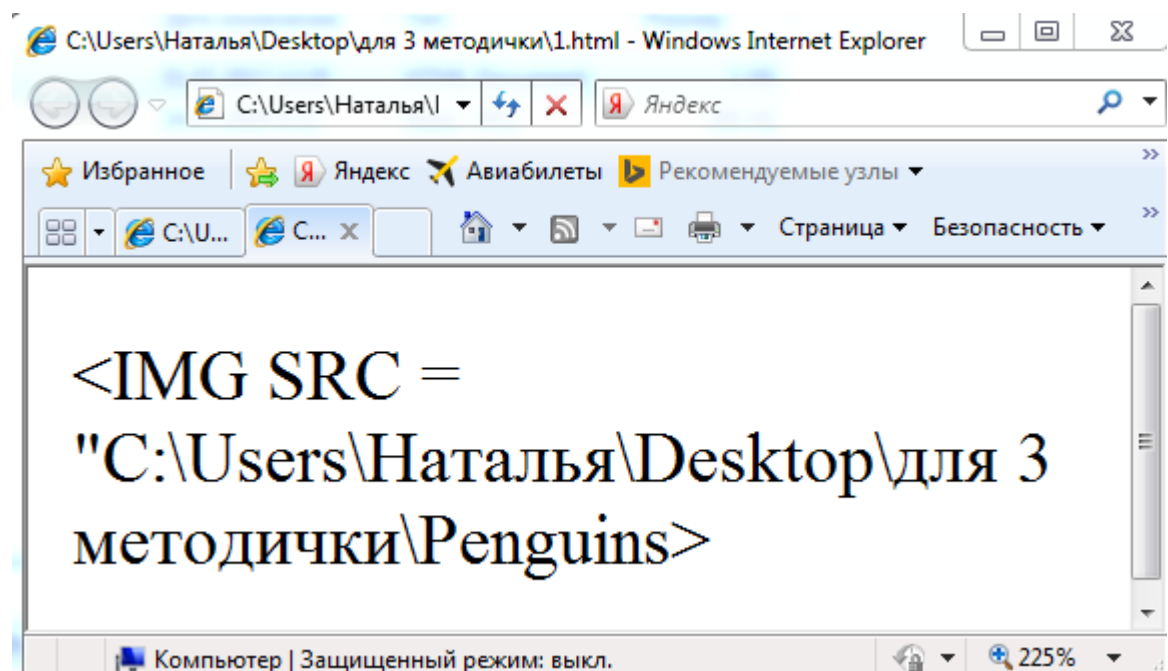
Ниже приведен пример простейшей интеграции изображения на Web-страницу — слева HTML-код, а справа — изображение в окне браузера.

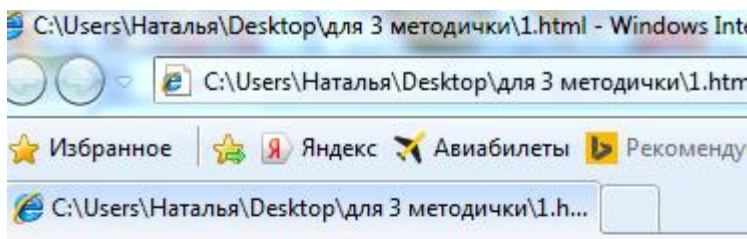


Однако, желательно задавать размеры изображения, чтобы можно было контролировать точность внедрения элемента-изображения на Web-страницу. Полный список атрибутов тега `` приведен далее. Ниже в таблицу сведены код html-разметки и результат этого при задании конкретных размеров изображения.



Возможны ситуации, когда путь к изображению прописан неправильно или не дописан. Примеры этого показаны ниже.





Атрибуты тега

- **ALIGN** - Определяет как рисунок будет выравниваться по краю и способ обтекания текстом.
- **ALT** - Альтернативный текст для изображения.
- **BORDER** - Толщина рамки вокруг изображения.
- **HEIGHT** - Высота изображения.
- **HSPACE** - Горизонтальный отступ от изображения до окружающего контента.
- **ISMAP** - Говорит браузеру, что картинка является серверной картой-изображением. longdesc Указывает адрес документа, где содержится аннотация к картинке.
- **SRC** - Путь к графическому файлу.
- **VSPACE** - Вертикальный отступ от изображения до окружающего контента.
- **WIDTH** - Ширина изображения.
- **USEMAP** - Ссылка на тег <map>, содержащий координаты для клиентской карты-изображения.

Рисунки также могут применяться в качестве карт-изображений, когда картинка содержит активные области, выступающие в качестве ссылок. Такая карта по внешнему виду ничем не отличается от обычного изображения, но при этом она может быть разбита на невидимые зоны разной формы, где каждая из областей служит ссылкой.

В этом случае в теге необходимо прописать [атрибут USEMAP](#), значение которого должно совпадать со значением, прописанным в теге <MAP>. Именно тегом <MAP> описывается изображение-карта, ее сегментирование, формы сегментов и их координатное расположение.

```
<IMG SRC="1.JPG" USEMAP = "#карта1">
<MAP NAME="карта1">
```

```
<body>
```

```
<div id="title"><br>
```

```
<p><map name="Navigation">
```

```
<area shape="poly" coords="113,24,211,24,233,0,137,0" href="inform.html" alt="Небо">
```

```
<area shape="poly" coords="210,24,233,0,329,0,307,24" href="activity.html"
alt="Мероприятия">
```

```
<area shape="poly" coords="304,24,385,24,407,0,329,0" href="depart.html"
alt="Отделения">
```

```
<area shape="poly" coords="384,24,449,24,473,0,406,0" href="techinfo.html"
alt="Техническая информация">
```

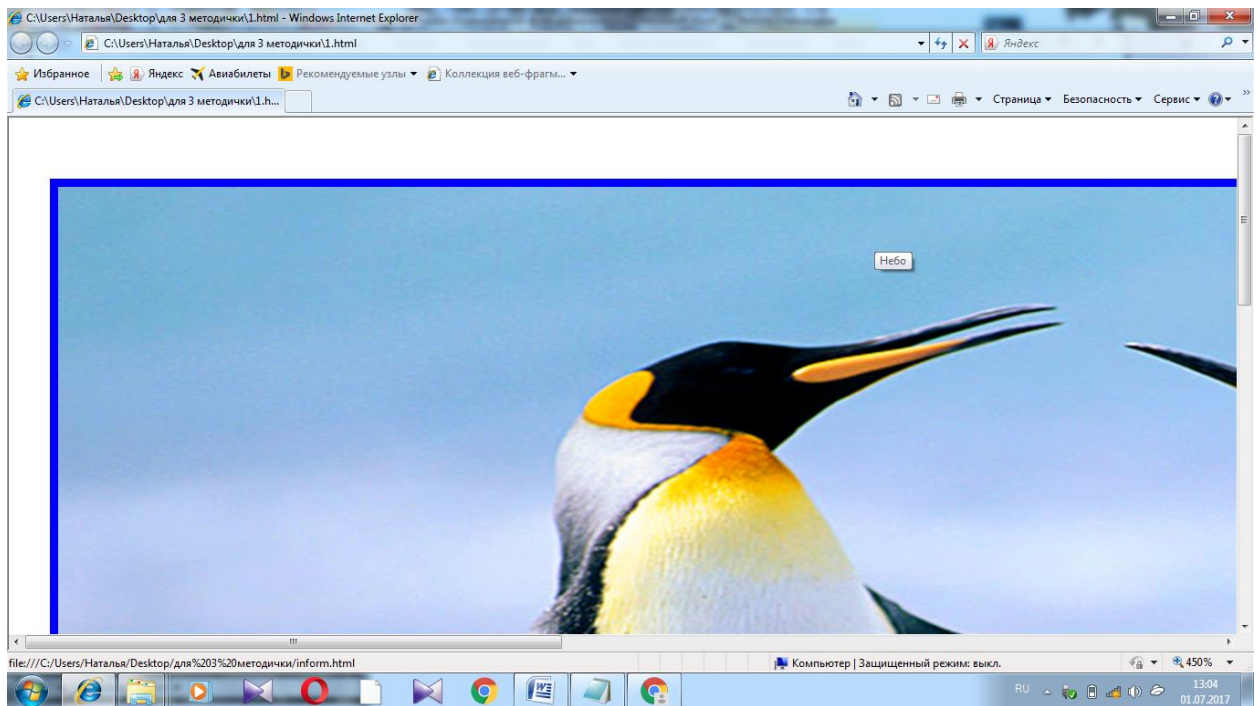
```
<area shape="poly" coords="449,24,501,24,525,0,473,0" href="study.html"
alt="Обучение">
```

```
<area shape="poly" coords="501,24,560,24,583,0,525,0" href="work.html" alt="Работа">
```

```
<area shape="poly" coords="560,24,615,24,639,0,585,0" href="misk.html" alt="Разное">
```

```
</map></p>
```

```
</body>
```

Описание

Атрибут `usemap` связывает между собой картинку и карту-изображение, задаваемую с помощью контейнера `<map>`. В качестве такой связи выступает имя идентификатора, которое указывается в значении атрибута `usemap`, и то же имя, заданное у атрибута `name` тега `<map>`. При этом в теге `` идентификатор должен начинаться с символа решетки (#).

Изображения могут применяться и в качестве фона Web-страницы. Для этого в теге `<BODY>` нужно прописать следующее:

```
<html>
<head>
<title> Изучаем тег img </title>
</head>
<body background="415084.JPG">
<table width=100% height=20 border=0>
<tr> <td> &nbsp; </td> </tr>
</table>
<u> <i> <font color=red size=0>Работаем с изображениями!!! </font> </i> </u>
</body>
</html>
```

Пример фонового изображения на странице, текста с различными атрибутами и пустой таблицы с заданными размерами приведен ниже.



Причем, размер используемого изображения, в данном случае, небольшой, поэтому оно «столбится» на странице. Во избежание этого возможна либо корректировка параметров картинки в любом графическом редакторе, либо через использование CSS (каскадных таблиц стилей) задание возможности/невозможности и направления дублирования изображения.

[перейти к содержанию](#)

ПРИМЕР ПРИМЕНЕНИЯ HTML-ТЕГОВ

```
<html>
<head> <title> Работа с таблицами в HTML </title> </head>
<body background="415084.JPG">
<center><table border="3" bordercolor="brown" cellspacing="2">
<tr bgcolor=grey><th colspan="4">
<div><B><U><font color=white> 1 </font></B></U></div></th><th
colspan="4"><div><B><U><font color=white> 2 </font></B></U></div></th></tr>
<tr bgcolor=yellow><td colspan="4" height = 5% background="415084.JPG"><div><I><font
color=red face=Odessa size=1> <UL>
<LI>Путешествуем по Германии
<OL>
<LI>Дрезден
<LI>Берлин
<LI>Мюнхен
</OL>
<LI>Путешествуем по Франции
<OL>
<LI>Париж
<LI>Бордо
<LI>Леон
</OL>
```

```

</UL> </font><I></div></td><td rowspan="2" colspan="2"><div><I><font color=red
face=Odessa> 4 </font><I></div></td><td rowspan="2" colspan="2"><div><I><font color=red
face=Odessa> 5 </font><I></div></td></tr>
<tr bgcolor=yellow><td colspan="4"><div><I><font color=red face=Odessa> 6
</font><I></div></td></tr>
<tr><td rowspan="2" colspan="4"><font color=blue> 7 </font></td><td colspan="2"><font
color=blue> 8 </font></td><td colspan="2"><font color=blue> 9 </font></td></tr>
<tr><td><font color=blue> 10 </font></td><td><font color=blue> 11 </font></td><td><font
color=blue> 12 </font></td><td><font color=blue> 13 </font></td></tr>
<tr bgcolor=lime><td rowspan="2" colspan="4"><font color=blue> 14 </font></td><td
rowspan="2" colspan="4"><font color=blue> 15 </font></td></tr>
</table></center>
</body>
</html>

```



5.2. ЗАДАНИЕ НА ПРАКТИЧЕСКУЮ РАБОТУ №3

Разработать Web-страницу, содержащую в своем составе текст, декорированный тем или иным способом, списки различных видов и изображения, используемые как в качестве фона на странице, так и собственно как изображения/сегментированные изображения. Необходимо:

1. Проработать различные атрибуты тегов , , <DL>, ;

2. Создать на странице вложенный список, который следует разместить в таблице 3*4 (три строки, четыре ячейки в каждой строке), причем в таблице должна быть реализована возможность объединения строк и ячеек строки;
3. Декорировать страницу, используя возможности тегов <TABLE>, ;
4. Применить различные виды маркеров списков;
5. Использовать возможность применения рамки с различными атрибутами вокруг изображения;
6. Внедрить на страницу сегментированное изображение, разбить его на области различной формы и размеров и реализовать переход на файлы с расширениями *.doc, *.jpg, *.ppt.

5.3. ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ И ПОДГОТОВКИ К ЗАЩИТЕ ПРАКТИЧЕСКОЙ РАБОТЫ №3

1. Опишите атрибуты тега IMG;
2. Продемонстрируйте работу с тегами OL, UL, DL на Web-странице;
3. Приведите пример вложенного списка;
4. Покажите возможности декорирования Интернет-страницы с использованием тега IMG;
5. Продемонстрируйте различные варианты работы с изображением на Web-странице;
6. Приведите возможные ошибки при внедрении изображения на Web-страницу.

[перейти к содержанию](#)

6. ПРАКТИЧЕСКАЯ РАБОТА №4.

ОСНОВЫ РАБОТЫ В HTML. РАБОТА С ГИПЕРССЫЛКАМИ

6.1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Тег <A> - тег гиперссылки

Тег <a> является одним из важных элементов HTML и предназначен для создания ссылок. В зависимости от присутствия [атрибутов name](#) или href тег <a> устанавливает ссылку или якорь. Якорем называется закладка внутри страницы, которую можно указать в качестве цели ссылки. При использовании ссылки, которая указывает на якорь, происходит переход к закладке внутри веб-страницы.

Для создания ссылки необходимо сообщить браузеру, что является ссылкой, а также указать адрес документа, на который следует сделать ссылку. В качестве значения атрибута href используется адрес документа (URL, Universal Resource Locator, универсальный указатель ресурсов), на который происходит переход. Адрес ссылки может быть абсолютным и относительным. Абсолютные адреса работают везде и всюду независимо от имени сайта или веб-страницы, где прописана ссылка. Относительные ссылки, как следует из их названия, построены относительно текущего документа или корня сайта.

```
Мы рады Вас приветствовать на <a href="a1.html">нашей странице </a>  
<a name="#Закладка1">Немного о нас (якорь гиперссылки)</a>
```

В первом случае происходит переход на внешний документ a1.html, а во втором случае – переход внутри текущего документа на закладку с именем Закладка1.

Атрибут href

Задаёт адрес документа, на который следует перейти. Поскольку в качестве адреса ссылки может использоваться документ любого типа, то результат перехода по ссылке зависит от конечного файла. Так, архивы (файлы с расширением zip или rar) будут сохраняться на локальный диск. По умолчанию новый документ загружается в текущее окно браузера, однако это свойство можно изменить с помощью атрибута target.

```
<a href=f1.html> Перейти на главную страницу </a>
```

Атрибут name

Атрибут name используется для определения якоря внутри страницы. Вначале следует задать в соответствующем месте закладку и установить ее имя при помощи атрибута name тега <a>. Имя ссылки на закладку начинается символом #, после чего идет название закладки. Название выбирается любое, соответствующее тематике. Можно также делать ссылку на закладку, находящуюся в другой веб-странице и даже другом сайте. Для этого в адресе ссылки следует указать ее адрес и в конце добавить символ решетки # и имя закладки.

Между тегами `` и `` текст писать не обязательно, так как требуется лишь указать местоположение перехода по ссылке.

Пример (Рис. 1 и Рис. 2)

```
<html>
<head>
  <title>Атрибут Name тега <A></title>
</head>
<body>
<a name=k>
  <p><a href="#doplit">К дополнительной литературе</a></p> </a>
  <p style="height:3000px;">
```

1. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

Фонд оценочных средств приведен в приложении к рабочей программе дисциплины и включает в себя:

.....Много текста.....

2. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Основная литература

.....Много текста.....

```
</p>
<p><a name="doplit">Дополнительная литература</a>
```

```
<p>
.....Много текста.....
```

```
</p>
<a href="#k">Наверх</a>
</body>
</html>
```

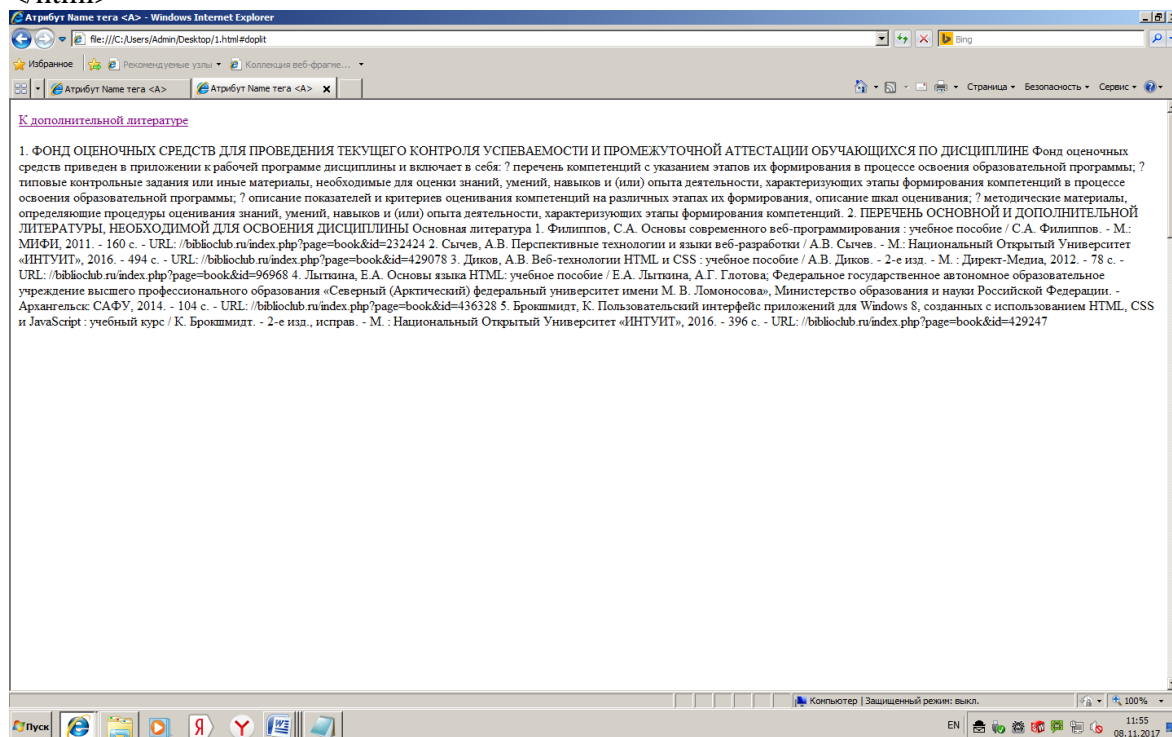


Рис. 1. Гиперссылка перемещения внутри документа

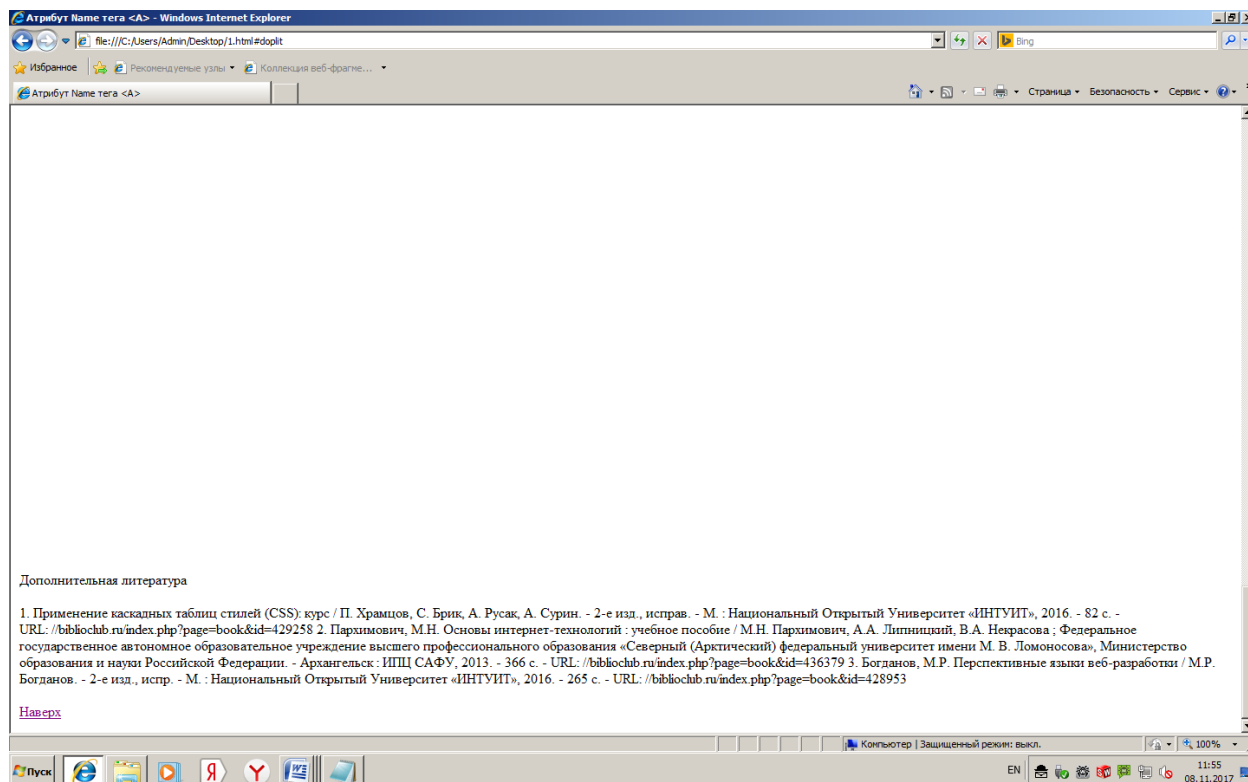


Рис. 2. Результат перемещения внутри документа

Атрибут **tabindex**

Атрибут **tabindex** определяет последовательность перехода между ссылками при нажатии на кнопку Tab.

`...`

В качестве значения используется любое целое положительное число. Значения выстраиваются последовательно и переход между ссылками происходит от меньшего значения к большему. Значения меньше нуля игнорируются и переход к таким ссылкам не происходит.

Пример

```
<html>
<head>
<title>Атрибут tabindex тег <A></title>
</head>
<body>
  <p><a href="f1.html" tabindex="1">Меню 1</a></p>
  <p><a href="f3.html" tabindex="3"> Меню 3</a></p>
</td>
  <p><a href="f2.html" tabindex="2"> Меню 2</a></p>
  <p><a href="f4.html" tabindex="4"> Меню 4</a></p>
</body>
</html>
```

Атрибут **target**

По умолчанию, при переходе по ссылке документ открывается в текущем окне или фрейме. При необходимости, это условие может быть изменено атрибутом `target` тега `<a>`.
`Переход на страницу Личное`

В качестве значения используется имя окна или фрейма, заданное атрибутом `name` для окна или фрейма. Если установлено несуществующее имя, то будет открыто новое окно. В качестве зарезервированных имен применяются следующие.

`_blank` — загружает страницу в новое окно браузера.

`_self` — загружает страницу в текущее окно (это значение задается по умолчанию).

`_parent` — загружает страницу во фрейм-родитель, если фреймов нет, то это значение работает как `_self`.

`_top` — отменяет все фреймы и загружает страницу в полном окне браузера, если фреймов нет, то это значение работает как `_self`.

Ниже приведен пример программного кода и рисунок (Рис. 3).

```
<html>
<head>
  <title>Ссылка в новом окне</title>
</head>
<body>
  <p><a href="new.html" target="_blank">Открыть в новом окне</a></p>
</body>
</html>
```

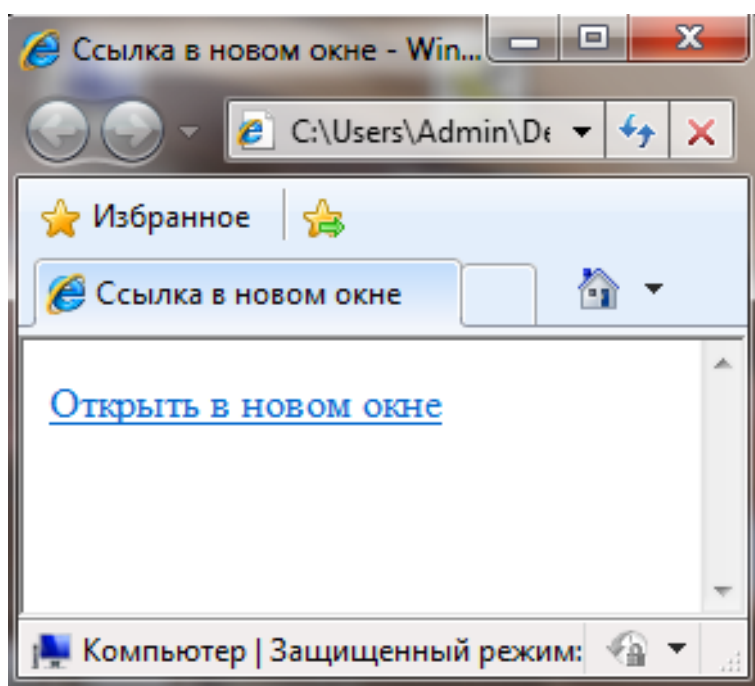


Рис.3. Работа с гиперссылкой

Атрибут `title`

Добавляет поясняющий текст к ссылке в виде всплывающей подсказки. Такая подсказка отображается, когда курсор мыши наводится на ссылку, после чего подсказка через некоторое время пропадает.

`...`

В качестве значения указывается любая текстовая строка. Строка должна заключаться в двойные или одинарные кавычки.

Пример (Рис. 4)

```
<html>
```



```

<head>
  <title>рисунки животных</title>
</head>
<body>
  <p><a href="zoo.html ">Фото цветов</a></p>
  <img src='C:\Users\Admin\Desktop' with="300" height="300" title="хризантема">
</body>
</html>

```

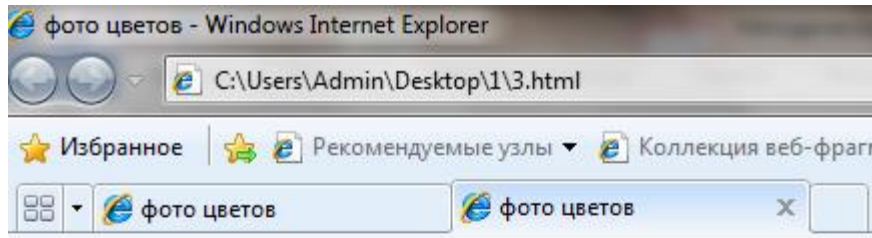


Рис.4. Работа с атрибутом title тега <a>

Цвета и оформления всплывающей подсказки зависят от настроек операционной системы и браузера, и меняться разработчиком не могут.

Атрибуты link, vlink и alink

Атрибуты LINK, ALINK, VLINK задают цвет шрифта гиперссылок.

Атрибут LINK служит для выделения гиперссылок, которые еще не посещались пользователем.

Атрибут VLINK - уже посещенные ссылки.

Атрибут ALINK - выделяет активную гиперссылку.

```
<body link="#008000" alink="#ff0000" vlink="#ffff00">
```

При необходимости можно принудительно выделять цветом как всю ссылку, так и отдельные её части (фразы слова буквы) тегом и его атрибутом color. Это касается не только цвета, так же отдельно можно задать размер, стиль и шрифт текста.

Следует учитывать, что манипуляции с цветом нужно проводить внутри тега <a> , а не вне данного тега, иначе цвет ссылки будет либо по умолчанию, либо так как прописано в теге <body>.

Пример (Рис. 5 и Рис. 6)

Файл index.html:

```
<html>
<head>
<title>Радуга</title>
</head>
<body link="#008000" alink="#ff0000" vlink="#ffff00">
<center>
<a href="primer.html">
<font size="+1" color=#ff0000>P</font>
<font size="+2" color=#ff8c40>A</font>
<font size="+3" color=#ffff00>Д</font>
<font size="+3" color=#008000>У</font>
<font size="+2" color=#0000ff>Г</font>
<font size="+1" color=#800080>А</font>
</a>
</center>
</body>
</html>
```

Файл primer.html:

```
<html>
<head>
<title>Радуга</title>
</head>
<body link="#008000" alink="#ff0000" vlink="#ffff00">
<center>
<font size="+3">
<font color=#ff0000>Каждый</font>
<font color=#ff8c40>охотник</font>
<font color=#ffff00>желает</font>
<font color=#008000>знать</font>
<font color=#40caff>где</font>
<font color=#0000ff>сидит</font>
<font color=#800080>фазан</font>
</font>
<br><br><br>
<a href="index.html">вернуться на главную</a>
</center>
</body>
</html>
```



Рис.5. Пример декорирования текста гиперссылки

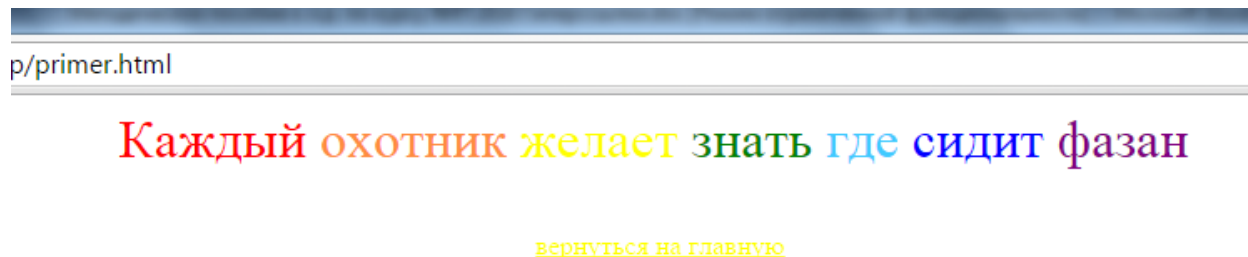


Рис.6. Переход по гиперссылке в соответствии с кодом

Изображение в качестве ссылки

Кроме текста, в качестве ссылок можно использовать и изображения. Изображение должно быть использовано в качестве [якоря гиперссылки](#).

Пример (Рис. 7)

```
<html>
<head>
<title>Ссылка</title>
</head>
<body>
<p><a href="sample.html"><a></p>
</body>
</html>
```

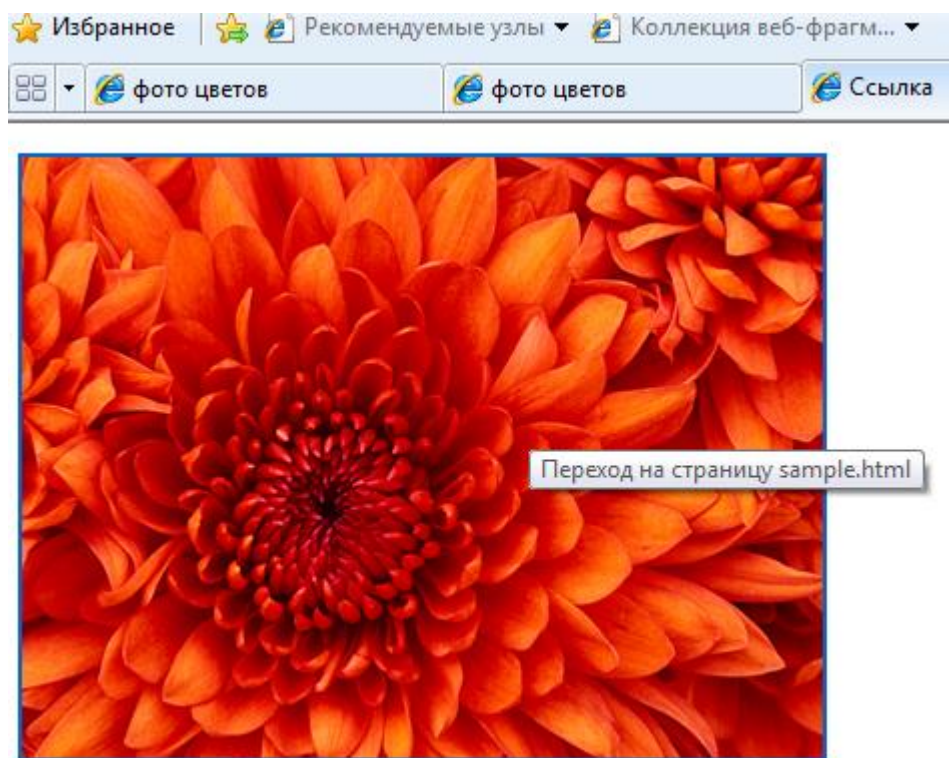


Рис.7. Пример использования изображения в качестве гиперссылки

Атрибут href тега <a> задает путь к документу, на который указывает ссылка, а src тега — путь к графическому файлу.

Вокруг изображения-ссылки автоматически добавляется рамка толщиной один пиксел и цветом, совпадающим с цветом текстовых ссылок.

Чтобы убрать рамку, следует у тега установить атрибут border="0".

Пример

```
<html>
<head>
  <title>Рамка вокруг изображения</title>
</head>
<body>
  <p><a href="sample.html"></a></p>
</body>
</html>
```

Можно также воспользоваться CSS, чтобы убрать рамку для всех изображений, которые являются ссылками. Для этого применяется стилевое свойство border со значением none.

Пример

```
<html>
<head>
  <title>Ссылка</title>
  <style type="text/css">
    A IMG { border: none; }
  </style>
</head>
```

```
<body>
<p><a href="sample.html"></a></p>
</body>
</html>
```

Конструкция <A IMG> определяет контекст применения стилей — только для тега , который находится внутри контейнера <a>. Поэтому изображения в дальнейшем можно использовать как обычно.

6.2. ЗАДАНИЕ НА ПРАКТИЧЕСКУЮ РАБОТУ №4

Разработать Web-страницу, содержащую в своем составе текст, декорированный тем или иным способом, списки различных видов и изображения, используемые как в качестве фона на странице, так и собственно как изображения. Также на Web-странице нужно представить внутренние и внешние гиперссылки.

Необходимо:

1. Проработать различные атрибуты тегов , , , <DL>, , <A>;
2. Декорировать страницу, используя возможности тегов <BODY>, , ;
3. Применить различные виды гиперссылок на Web-странице с различными атрибутами;
4. Реализовать словарь терминов с возможностью возврата в начало страницы после каждого определения;

6.3. ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ И ПОДГОТОВКИ К ЗАЩИТЕ ПРАКТИЧЕСКОЙ РАБОТЫ №4

1. Опишите атрибуты тега <A>;
2. Продемонстрируйте работу с гиперссылками на Web-странице;
3. Приведите пример закладок в пределах одной страницы;
4. Покажите возможности декорирования гиперссылок;
5. Продемонстрируйте различные варианты использования различных элементов Web-страницы в качестве гиперссылок;
6. Приведите возможные ошибки при внедрении гиперссылок на Web-страницу.

[перейти к содержанию](#)

7. ПРАКТИЧЕСКАЯ РАБОТА №5.

ОСНОВЫ РАБОТЫ В HTML. РАБОТА С ФОРМАМИ

7.1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ К ПРАКТИЧЕСКОЙ РАБОТЕ №5

ЭЛЕМЕНТЫ ФОРМ

Текстовое поле

Простое однострочное текстовое поле (Рис. 1), в которое можно вводить и редактировать текст. Задается тегом `<input>`.

```
<html>
<body>
<form name="forma1">
<input type="text" name="text1" size="20" maxlength="50" value="Введите текст">
</form>
</body>
</html>
```

Атрибуты элемента Текстовое поле:

- name - имя элемента;
- type - тип элемента (в данном случае - text);
- size - размер текстового поля в символах, которые одновременно будут видны. При вводе большего количества символов, они будут прокручиваться;
- maxlength - максимальное количество символов, которое можно ввести в поле. Если опустить этот параметр, то число символов будет неограниченным;
- value - текст, который будет отображаться (его можно стереть), при отсутствии этого параметра поле будет пустым.
- disabled - блокирует поле от любых изменений;
- readonly - делает поле доступным только для чтения.

```
<html>
<body>
<form name="forma1">
<input type="text" name="text1" size="20" maxlength="50" value="Введите текст">
<br>
<br>
<input type="text" name="text2" size="10" value="Текстовое поле 2" readonly>
</form>
</body>
</html>
```

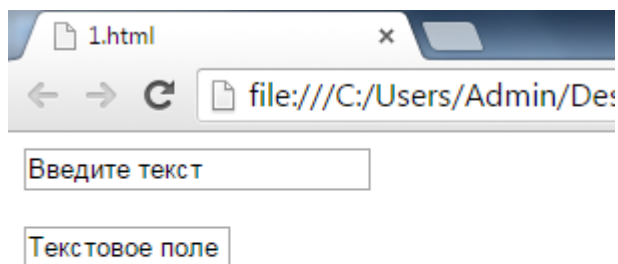


Рис.1 Пример текстовых полей на Web-странице

Текстовое поле для ввода пароля

Это такое же [текстовое поле](#), как и предыдущий элемент. Разница только в том, что вводимый текст не отображается, вместо него появляются специальные символы, например звездочки. Чаще всего используется при вводе паролей. Все параметры такие же, как у простого текстового поля, кроме параметра `type="password"`.

```
<html>
<body>
  <form name="form1">
    <input type="text" name="text1" size="20" maxlength="50" value="Введите текст">
  <br>
  <br>
    <input type="password" name="text2" size="10" value="Текстовое поле 2">
  </form>
</body>
</html>
```

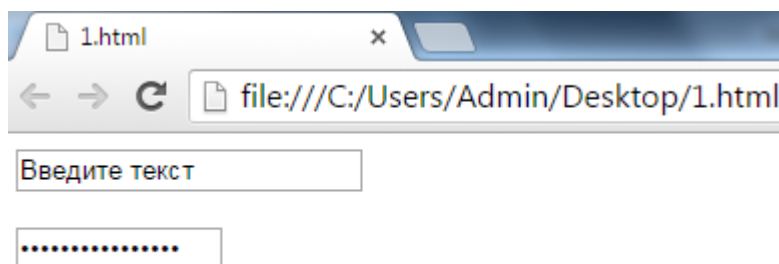


Рис.2. Пример поля для ввода пароля на Web-странице

Флажки (checkbox)

В этом случае значением атрибута `type` тега `<input>` является `checkbox`. При этом нужно прописывать такое количество тегов `<input>`, сколько флажков предполагается на странице (Рис.3).

```
<html>
<body>
  <form name="form1">

    <input type="text" name="text1" size="20" maxlength="50" value="Введите текст">
```

```

<br>
<br>
    <input type="password" name="text2" size="10" value="Текстовое поле 2">
<br>
Ваши увлечения:
<br>
Спорт <input type="checkbox" name="text2" value="Спорт">
<br>
Кулинария <input type="checkbox" name="text3" value="Кулинария">
<br>
Рыбалка <input type="checkbox" name="text4" value="Рыбалка" checked>
    </form>
</body>
</html>

```

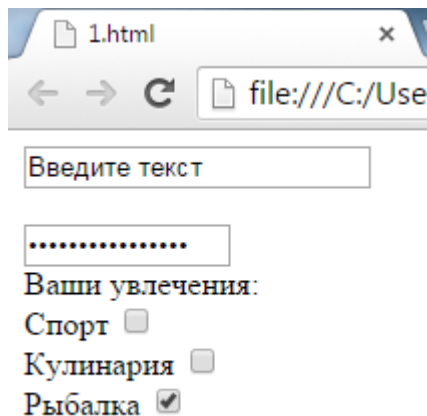


Рис. 3. Пример флажков на Web-странице

Атрибуты тега `<input>` для checkbox:

- `type` - тип элемента (в данном случае - checkbox);
- `name` - имя элемента, указывает программе - обработчику формы, какой пункт выбрал пользователь;
- `value` - значение элемента, указывает программе - обработчику формы значение пункта, который выбрал пользователь.
- `checked` - им обычно помечают наиболее вероятные для выбора пункты, пользователь кликом мыши может выбрать другие пункты.

Переключатели (радиогруппы)

В данном случае в отличие от флажков возможен только один вариант выбора. В связи с этим значения параметра `name` должны быть одинаковы для всех элементов группы. Параметр `type="radio"`. Все остальные такие же, как у флажков (Рис. 4).

```

<html>
<body>
<form name="form1">

    <input type="text" name="text1" size="20" maxlength="50" value="Введите текст">

```


2. `image` - графическая кнопка отправки содержимого формы web-серверу. Для ее использования необходимо подготовить картинку кнопки, а потом использовать ее в виде кнопки. Ее параметры:

- ✓ `type="image"` - тип графической кнопки;
- ✓ `name` - имя кнопки;
- ✓ `src` - адрес картинки для кнопки.

3. `reset` - кнопка, позволяющая восстановить все значения по умолчанию в форме. Ее параметры:

- ✓ `type="reset"` - тип кнопки очищения;
- ✓ `name` - имя кнопки;
- ✓ `value` - надпись на кнопке.

4. `button` - произвольная кнопка, ее действия назначаются разработчиком, т.е. сама она делать ничего не умеет. Ее параметры:

- ✓ `type="button"` - тип произвольной кнопки;
- ✓ `name` - имя кнопки;
- ✓ `value` - надпись на кнопке.
- ✓ `onclick` - указывает, что делать при клике по кнопке.

Если на форме несколько кнопок, то они должны иметь разные названия.

Пример кода:

```
<html>
<body>
<form name="form1">
  <input type="submit" name="submit" value="Отправить">
  <input type="image" name="but_img" src="корова.jpg">
  <input type="reset" name="reset" value="Очистить">
  <input type="button" name="button" value="Отправить">
</form>
</body>
</html>
```

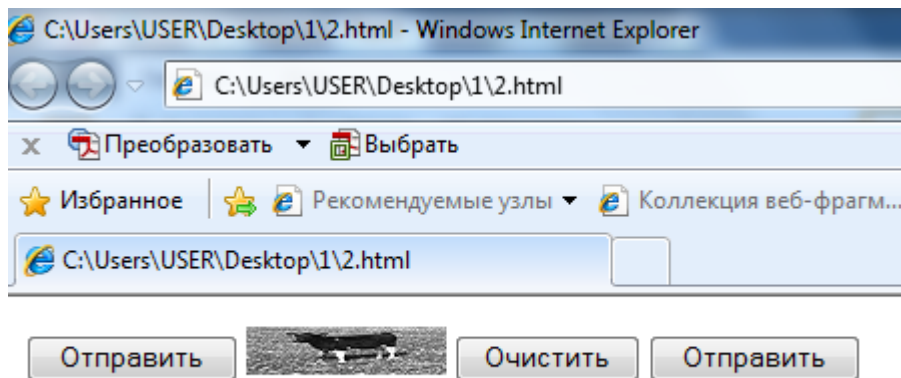


Рис. 5. Пример кнопок на странице

Кнопки можно задавать и при помощи тегов `<button>` `</button>`. Возможности у таких кнопок несколько шире, они могут иметь содержимое в виде текста или картинки. Этот тег имеет следующие параметры:

- ✓ `type` - тип кнопки, может принимать значения:
 - `reset` - кнопка очистки формы,
 - `submit` - кнопка отправки данных,
 - `button` - кнопка произвольного действия.
- ✓ `name` - имя кнопки,
- ✓ `value` - надпись на кнопке.

Пример кода:

```
<html>
<body>
<form name="forma2">
  <input type="submit" name="submit" value="Отправить">
  <input type="image" name="but_img" src="корова.jpg">
  <input type="reset" name="reset" value="Очистить">
  <input type="button" name="button" value="Отправить">
</form>
<form name="forma1">
  <button name="submit" type="submit" value="кнопка">
    
    <font size="4"> Кнопка </font>
  </button>
</form>
</body>
</html>
```

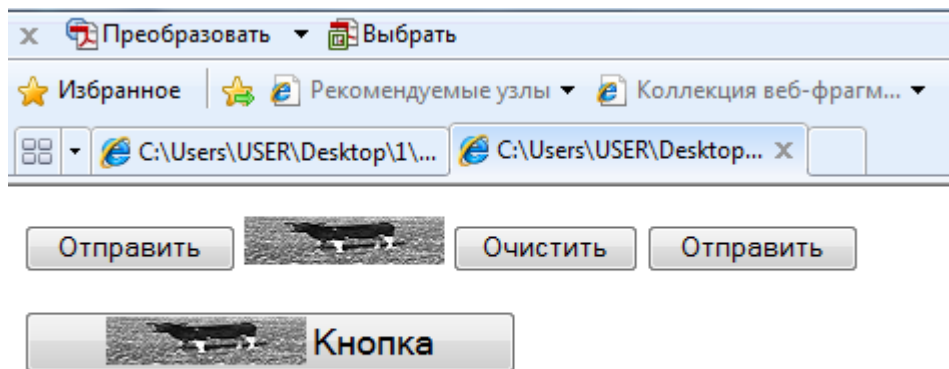


Рис. 6. Пример кнопки с тегом <button>

Поле для ввода имени файла

Это поле для ввода имени файла, сопровождаемое кнопкой Browse (Обзор), при клике по которой открывается окно просмотра дерева папок компьютера, где можно выбрать нужный файл. Выбранный файл присоединяется к содержимому формы при отправке на сервер.

```
<html>
<body>
<form name="forma2">
  <input type="submit" name="submit" value="Отправить">
  <input type="image" name="but_img" src="a11.jpg">
  <input type="reset" name="reset" value="Очистить">
  <input type="button" name="button" value="Отправить">
</form>
<form name="forma1">
  <button name="submit" type="submit" value="кнопка">
    
    <font size="4"> Кнопка </font>
  </button>
<br><br>

  <input type="file" name="load" size="50">

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

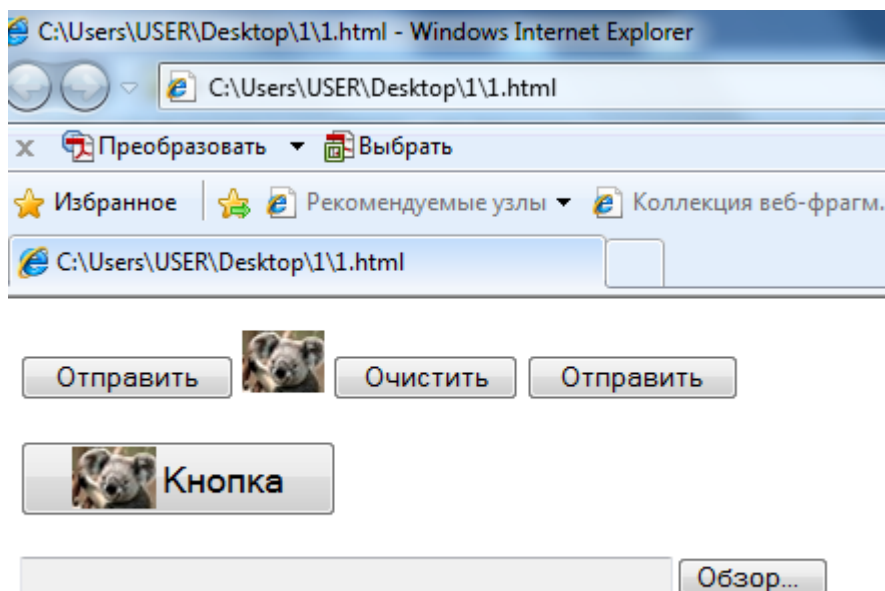


Рис. 7. Пример выбора файла

Многострочное текстовое поле

Данный элемент формы используется, например, для объемных текстов почтовых сообщений, удобно использовать именно этот элемент. Он создается тегами `<textarea>` `</textarea>` и имеет следующие параметры:

- name - имя поля,
- cols - ширина поля в символах,
- rows - количество строк текста, видимых на экране,
- wrap - способ переноса слов:
 - ✓ off - переноса не происходит,
 - ✓ virtual - перенос отображается, но на сервер поступает неделимая строка,
 - ✓ physical - перенос и на экране, и при поступлении на сервер.
- disabled - неактивное поле,
- readonly - разрешено только чтение.

```
<html>
<body>
<form name="forma2">
  <input type="submit" name="submit" value="Отправить">
  <input type="image" name="but_img" src="a11.jpg">
  <input type="reset" name="reset" value="Очистить">
  <input type="button" name="button" value="Отправить">
</form>
<form name="forma1">
```

```

<button name="submit" type="submit" value= "кнопка">
  
  <font size="4"> Кнопка </font>
</button>
<br><br>
<input type="file" name="load" size="50"><br>
<textarea cols="20" rows="3" wrap="off"></textarea><br>
<textarea cols="35" rows="5" wrap="virtual"></textarea><br>
<textarea cols="50" rows="7" wrap="physical"></textarea>
</form>
</body>
</html>

```

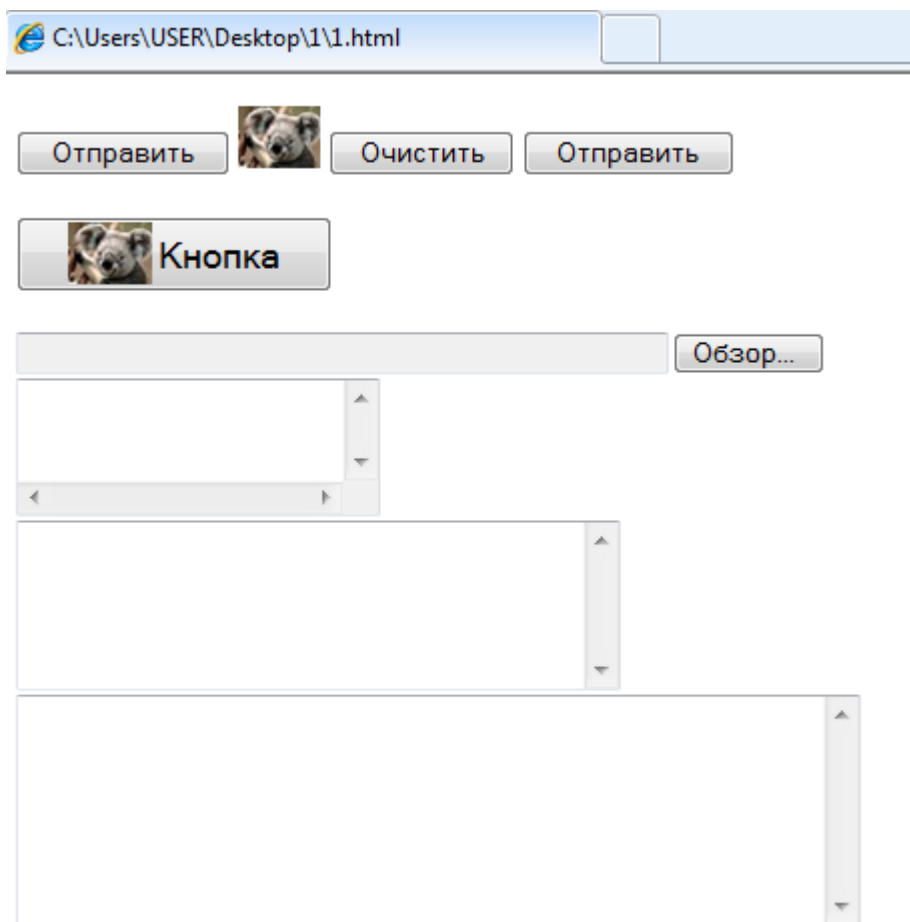


Рис. 8. Интеграция многострочных текстовых полей на странице

Раскрывающиеся списки

Данный вид списка в отличие от [списков](#), задаваемых тегами , и <DL>, позволяет задать раскрывающийся список на Web-странице с возможностью отображения различного количества элементов списка. Также в данном случае возможен выбор нескольких элементов из списка для последующей обработки.

Раскрывающийся список задается с помощью тегов <select> </select>, внутри которых располагаются элементы значений, заданных тегом <option>. Параметры этих тегов:

1. Для тега <select>:
 - ✓ name - имя списка. Каждый выбранный элемент списка при передаче на сервер будет иметь вид: name.value, где значение (value) берется из тега option.

- ✓ size - определяет количество видимых элементов в списке: 1 (значение по умолчанию) - простой раскрывающийся список, число более 1 - список с полосой прокрутки (при этом количество видимых элементов списка задается именно этим значением) .

- ✓ multiple - разрешает выбор нескольких элементов списка.

2. Для тега <option>:

- ✓ selected – означает, что данный вид списка выбран по умолчанию.

- ✓ value - значение, которое будет отправлено серверу, если пункт выбран.

```
<html>
<body>
<form name="form1">
Какой язык Web-программирования Вы хотите изучать:
<select name="language" size="1">
<option selected value="html">html-разметку
<option value="php">php
<option value="java">JavaScript
</select><br><br>
Какое время Вы готовы на это потратить:<br>
<select name="time" size="3">
<option selected value="1">1 месяц
<option value="2">2 месяца
<option value="3">3 месяца
</select><br><br>
Наиболее подходящие дни занятий:<br>
(выбирайте с нажатой клавишей ctrl)<br>
<select name="day" size="7" multiple>
<option selected value="mon">понедельник
<option value="tue">вторник
<option value="wen">среда
<option selected value="thu">четверг
<option value="fri">пятница
<option value="sat">суббота
<option value="san">воскресенье
</select>
</form>
</body>
</html>
```

Какой язык Web-программирования Вы хотите изучать: php

Какое время Вы готовы на это потратить:

- 1 месяц
- 2 месяца
- 3 месяца

Наиболее подходящие дни занятий:
(выбирайте с нажатой клавишей ctrl)

- понедельник
- вторник
- среда
- четверг
- пятница
- суббота
- воскресенье

Рис. 9. Пример раскрывающихся списков

Существуют еще теги `<optgroup>` `</optgroup>`, позволяющие группировать элементы списка по каким-либо признакам. Например, мы хотим задать каталог сайтов в виде списка, тогда удобнее разбить его на группы по интересам. Для этого нужны теги `<optgroup>` `</optgroup>` с одним параметром `label`, который задает название группе элементов.

```
<html>
<body>
<form name="forma1">
  Каталог сайтов:<br>
  <select name="catalog" size="9">
    <optgroup label="Компьютеры">
      <option value="1">интернет</option>
      <option value="2">мобильники</option>
      <option value="3">hardware</option>
    </optgroup>
    <optgroup label="Работа">
      <option value="4">вакансии</option>
      <option value="5">трудоустройство</option>
      <option value="6">резюме</option>
    </optgroup>
    <optgroup label="Дом">
      <option value="7">здоровье</option>
      <option value="8">красота</option>
      <option value="9">дети</option>
    </optgroup>
  </select>
</form>
</body>
</html>
```

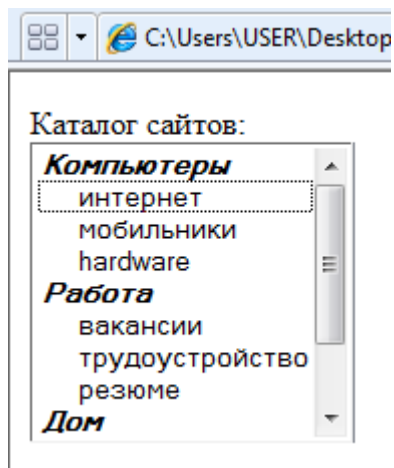



Рис. 10. Пример применения тега <optgroup>

Надписи

Все элементы формы можно связать с их надписями при помощи элемента <label> и его параметра for, значением которого является имя элемента, с которым связывается надпись.

```
<html>
  <head>
    <title>Заголовок документа</title>
  </head>
  <body>
    <form>
      <fieldset><legend> Личная информация </legend>
      <table border="0" cellspacing="5" cellpadding="5">
        <caption><H2>Форма регистрации</H2></caption>
        <tr>
          <td>Имя</td>
          <td><input type="text" name="name" size="25"></td>
        </tr>
        <tr>
          <td>e-mail</td>
          <td><input type="text" name="e-mail" size="25"></td>
        </tr>
        <tr>
          <td>Пароль</td>
          <td>
            <input type="password" name="password" size="25">
          </td>
        </tr>
        <tr>
          <td>Повтор пароля</td>
          <td>
            <input type="password" name="password2" size="25">
          </td>
        </tr>
        <tr>
          <td>Пол</td>
          <td>
```


</html>

Личная информация

Форма регистрации

Имя

e-mail

Пароль

Повтор пароля

Пол ☒ мужской ☐ женский

Увлечения

компьютеры
спорт
игры
животные

Ваше образование: Среднее ☐ Специальное ☐ Высшее ☒

Ваши пожелания

Отправка

Рис. 12. Обобщающий пример работы с формами с формализованным разделением содержимого

Элемент `<fieldset>...</fieldset>` предназначен для группировки элементов, связанных друг с другом, разделяя, таким образом, форму на логические фрагменты. Каждой группе элементов можно присвоить название с помощью элемента `<legend>`, который идет сразу за тегом `<fieldset>`. Название группы проявляется слева в верхней границе `<fieldset>`.

[перейти к содержанию](#)

НОВЫЕ ТИПЫ ПОЛЕЙ ВВОДА ЭЛЕМЕНТОВ ФОРМЫ В HTML 5.0

Color

```
<input id="color" name="color" type="color">
```

Этот тип поля очень прост для понимания: он обеспечивает выбор цвета для пользователя, и генерирует шестнадцатеричное значение. Палитра цветов будет нативной для каждой операционной системы или браузера, как и большинство новых полей.

Date

```
<input id="date" name="date" type="date">
```

Поле Date позволяет выбирать дату из раскрывающегося списка.

Date-time

`<input id="departure-day-time" name="departure-day-time" type="date-time">`

Используется для обеспечения возможности одновременного выбора не только даты, но и времени, используя тип поля "date-time".

Datetime-local

`<input id="arrival-time" name="arrival-time" type="datetime-local">`

Этот тип поля точно такой же как "date-time", но он предоставляет время в локальной временной зоне.

Email

`<input type="email" name="email" id="email">`

Этот тип поля создан специально для ввода адреса электронной почты.

Month

`<input id="expiry" name="expiry" type="month">`

Это поле ввода позволяет выбрать месяц года, это очень удобно, например, для ввода информации о кредитной карте в форме заказа.

Number

`<input type="number" min="48" max="84" step="0.5" value="48" name="height-inches">`

Поле Number предназначено только для ввода чисел. Можно указать разрешенный диапазон для чисел, если будут указаны атрибуты min и max, как в приведенном выше примере, а также указать шаг в этом диапазоне.

Range

`<input type="range" min="1" max="100" value="0" id="happiness">`

В поле Range браузер выводит ползунок, который пользователь может перетаскивать влево-вправо. Можно указать минимальное и максимальное значение, а также значение по умолчанию.

Tel

`<input type="tel" name="tel" id="tel">`

Этот тип поля очень удобно использовать в смартфонах, так как оно вызывает цифровую клавиатуру телефона, чтобы можно было ввести номер телефона максимально просто. Оно, однако, не проверяет правильность ввода, в отличие от Email или URL.

Time

`<input name="start-time" id="start-time" type="time">`

Этот тип ввода позволяет пользователям выбирать только определенное время, например, часы и минуты.

Url

```
<input type="url" name="url" id="url">
```

Это поле проверяет на правильность ввода URL-адреса, а на смартфоне откроется URL-клавиатура.

Week

```
<input id="conference" name="conference" type="week">
```

Это поле предоставляет пользователю возможность выбора определенной недели.

НОВЫЕ АТТРИБУТЫ ФОРМ

Рассмотрим новые атрибуты форм, которые могут быть использованы в сочетании с новыми или ранее существовавшими полями. Существует в общей сложности четырнадцать новых атрибутов.

Autocomplete

```
<input type="text" name="fullname" id="fullname" autocomplete="on">
```

Автозаполнение - этот атрибут помогает пользователям заполнить поле на основе предыдущих записей, он может либо сэкономить время, либо вызвать проблемы, так что аккуратнее с этим.

Autofocus

```
<input type="text" name="username" id="username" autofocus>
```

Этот атрибут ставит курсор в нужное поле при загрузке страницы, так что пользователь может сразу начать печатать.

Form

```
<input type="button" name="filter" form="filter">
```

Form связывает поле с формой по идентификатору, поэтому элемент не обязательно должен быть в пределах соответствующей формы, чтобы быть её частью.

Formaction

```
<input type="submit" value="Send" formaction="email.php">
```

Указывает обработчик формы.

Formenctype

```
<input type="submit" value="Send" formenctype="application/x-www-form-urlencoded">
```

Устанавливает способ кодирования данных формы при их отправке на сервер.

Formnovalidate

```
<form action="login.php">
  <label for="name">Полное имя:</label>
  <input type="text" name="name" value="Иванов Иван">
  <input type="submit" formnovalidate value="Send">
</form>
```

Этот атрибут указывает, что форму не нужно проверять при отправке данных. Атрибут может быть установлен для любого типа поля.

Formmethod

```
<input type="submit" value="Send" formmethod="POST">
```

Определяет, какой метод HTTP использовать, чтобы отправить форму, например, POST или GET.

Formtarget

```
<input type="submit" value="Send" formtarget="_self">
```

Определяет окно или фрейм в которое будет загружаться результат, возвращаемый обработчиком формы.

List & Multiple

```
<label>Мой любимый цвет:
<datalist id="colours">
  <select name="colours">
    <option value="Красный">Красный</option>
    <option value="Желтый">Желтый</option>
    <option value="Зеленый">Зеленый</option>
    <!-- ... -->
  </select>
```

Если другой, укажите:

```
</datalist>
  <input type="text" name="colour" list="colours" multiple>
</label>
```

Тег <datalist> определяет список предопределенных вариантов элемента <input>, которые можно выбрать при наборе в текстовом поле.

Novalidate

```
<form action="login.php" novalidate>
  <label for="name">Full name:</label>
  <input type="text" name="name" value="Пример">
  <input type="submit" value="Send">
</form>
```

Указывает, что форма не должна быть провалидирована при её отправке на сервер, этот атрибут устанавливается только на элемент формы.

Placeholder

```
<input type="password" name="password" id="password" placeholder="Введите ваш пароль">
```

Placeholder обычно отображается в виде светло-серого текста в поле, до тех пор пока поле не активно. Обычно этот атрибут содержит подсказку, что именно необходимо ввести в это поле.

Required

`<input type="text" name="username" id="username" required>`

Атрибут 'required' делает поле обязательным к заполнению и форма не может быть отправлена, пока требуемые данные не будут введены.

7.2. ЗАДАНИЕ НА ПРАКТИЧЕСКУЮ РАБОТУ №5

Написать Web-страницу с использованием элементов форм. Необходимо:

1. Проработать различные элементы форм на странице;
2. Применить теги `<fieldset>` и `<legend>` для логического оформления (логической компоновки) элементов форм на Web-странице;
3. Использовать на веб-странице элементы форм, присущие HTML 5.0;
4. Применить в разрабатываемых Web-страницах новые атрибуты элементов форм, описанные в пункте 1.3.3;
5. Для пунктов 3 и 4 задания на практическую работу разработать форму, максимально использующую элементы тега `<form>`. для создания страницы портфолио при оформлении на почасовую работу. При этом необходимо с помощью элементов выбора даты и времени определить желаемые дни и часы работы, временной промежуток работы с перерывами. Использовать поля автозаполнения и т.п.

7.3. ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ И ПОДГОТОВКИ К ЗАЩИТЕ ПРАКТИЧЕСКОЙ РАБОТЫ №5

1. Опишите элементы форм, укажите атрибуты каждого из элементов;
2. Приведите примеры интегрирования указанных элементов на страницу;
3. Перечислите новые возможности работы с формами в HTML 5.0;
4. Расскажите о логической компоновке элементов форм.

[перейти к содержанию](#)

8. ПРАКТИЧЕСКАЯ РАБОТА № 6. ОСНОВЫ РАБОТЫ В HTML. ФРЕЙМЫ.

8.1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ К ПРАКТИЧЕСКОЙ РАБОТЕ №6

Теги работы с фреймами. <Frameset>

При использовании фреймов необходимо как минимум три HTML-файла: первый определяет [фреймовую структуру](#) и делит окно браузера на две части, а оставшиеся два документа загружаются в заданные окна. Количество фреймов не обязательно равно двум, может быть и больше, но никак не меньше двух.

Разбиение документа на отдельные окна (фреймы) определяется с помощью тегов <frameset>... </frameset>. Например,

```
<frameset rows=" 30%,*">  
  
<frame name="a1" src="f1.html">  
  
<frame name="a2" src="f2.html">  
  
</frameset>
```

В данном случае окно документа будет разбито на два горизонтальных фрейма, параметры которых будут описаны ниже.

При работе с фреймами отсутствует тег <body>... </body>.

Тег <frameset> заменяет [тег <body>](#) в документе и применяется для разделения экрана на области. Внутри данного тега находятся теги <frame>, которые указывают на HTML-документ, предназначенный для загрузки в область. Рассмотрим данную тему на конкретных примерах.

Пример 1.

Необходимо разбить область документа так, как показано на Рисунке 1.

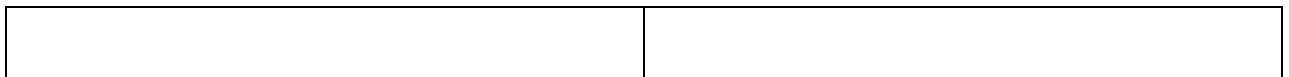


Рис.1. Разбиение окна документа на два вертикальных фрейма.

В данном случае тег <frameset> будет выглядеть следующим образом:

```
<frameset cols="50%,*">
```

Весь файл разбиения на подобную структуру будет содержать следующий код:


```

<frameset cols="50%,*">
<frame src="f1.html">
<frame src="f2.html">
</frameset>

```

В данном случае в файлах f1.html и f2.html находятся html-файлы произвольного содержания. В качестве значения атрибута src может выступать путь к файлу любого типа.

Таким образом, тег <frameset> определяет структуру документа, т.е. разбиение Web-страницы на отдельные окна, в которых могут отображаться документы различного типа.

Например, необходимо определить следующую структуру документа.

Пример 2. Необходимо разбить область документа так, как показано на Рисунке 2.

Рис.2. Разбиение окна документа на сложную фреймовую структуру

Программный код будет выглядеть следующим образом:

```

<html>
<frameset rows="33%,33%,34%">
<frameset cols="15%,20%,65%">
<frame src="f2.html">
<frame src="f2.html">
<frame src="f2.html">
</frameset>
<frame src="f2.html">
<frameset cols="33%,33%,34%">
<frame src="f2.html">
<frame src="f2.html">
<frame src="f2.html">
</frameset>
</frameset>
</html>

```

Результат выполнения приведен на рис. 3.

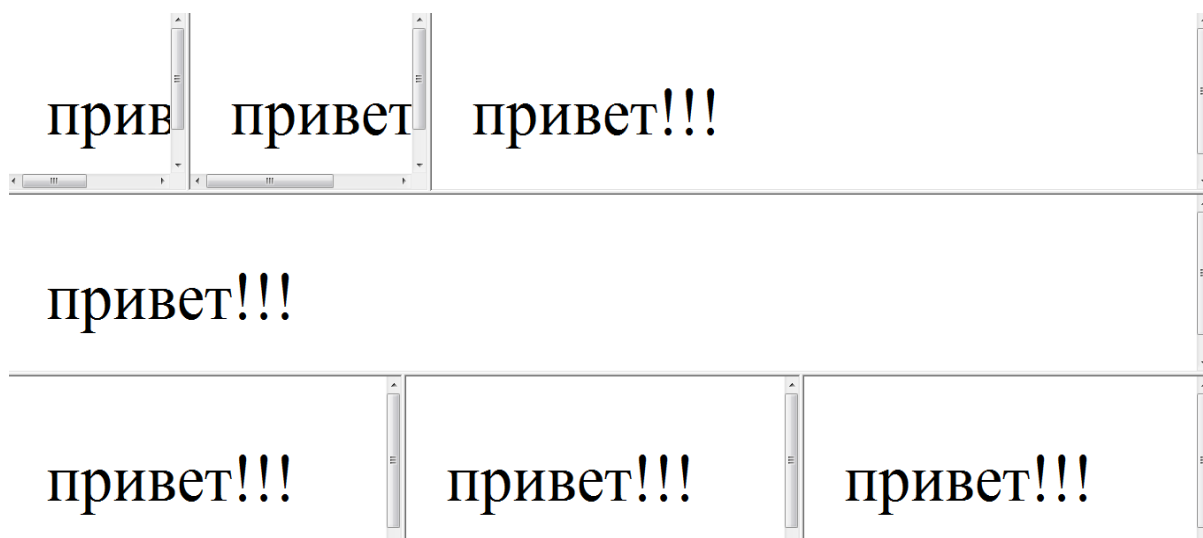


Рис. 3. Пример разбиения окна web-страницы на фреймы

Для данного примера следует учитывать, что:

В соответствии с программным кодом все окно разбивается на три строки;

Первая строка – это `<frameset>`, состоящий из трех столбцов. То есть первая строка разбивается на три ячейки. В каждом подокне отображается файл, прописанный в атрибуте `src`.

Вторая строка – это окно, в котором отображается содержимое файла `f2.html`;

Третья строка аналогична по содержанию первой строке.

Атрибуты тега `<frameset>`

- ✓ `Rows` – количество строк, на которое разбивается рассматриваемая область документа;
- ✓ `Cols` – количество столбцов, на которое разбивается рассматриваемая область документа;

Значения данных атрибутов могут выражаться в пикселах, % от ширины или высоты экрана пользователя или * (оставшееся пространство).

- ✓ `Name` – имя набора фреймов
- ✓ `Scrolling` – определяет наличие полос прокруток (`auto`, `yes`, `no`)
- ✓ `Frameborder` – задаёт показ границ между фреймами (`yes`, `no`), если `Frameborder= yes`, то добавляются параметры:
 - `border` – толщина рамки
 - `bordercolor` – цвет рамки

```
<html><head><title>фреймы</title></head>
<frameset rows="20%, 80%" frameborder="yes" border="1">
<frame ...>
<frameset cols="300px, *">
<frame ...>
<frame ...>
</frameset>
</frameset>
```

</html>

Далее будет показано, как задать отдельные фреймы и определить их содержимое.

Описание отдельных фреймов

Атрибуты тега <frame>:

- ✓ Name – имя фрейма;
- ✓ Src – URL открываемого в этом фрейме файла;
- ✓ Scrolling – определяет наличие полос прокруток (auto, yes, no);
- ✓ Marginwidth – отступ по горизонтали;
- ✓ Marginheight – отступ по вертикали;
- ✓ Framborder – задаёт показ границ между фреймами (yes, no), если Framborder= yes, то добавляются параметры:
 - Border – толщина рамки
 - Bordercolor – цвет рамки
- ✓ Noresize – запрещает интерактивно перемещать границы фрейма.

Для организации межфреймового взаимодействия существует параметр *target*:

Target=_blank - открывает документ в новом окне.

Target=_self - открывает документ в том же фрейме.

Target=_top – документ загружается поверх всех фреймов.

Target=_parent – документ загружается во фрейм, являющийся родительским для данного фрейма.

Target="имя фрейма" – загружается во фрейм с данным именем.

Пример фреймовой организации страницы показан на рисунке 4, а код примера приведен ниже.

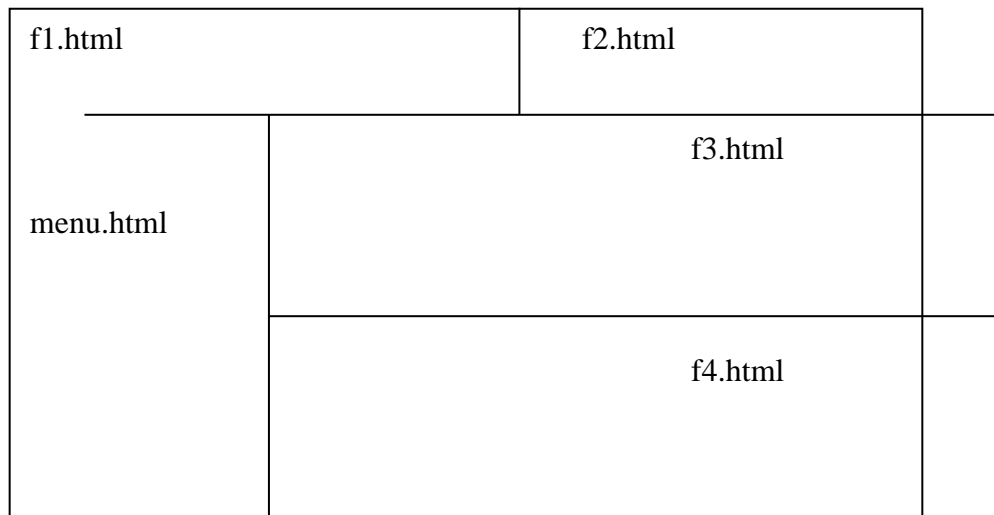


Рис. 4. Пример формы

<html>

<head>

<title>фреймы</title>

</head>

<frameset rows="30%, *">

<frameset cols="50%, *">

```

<frame src="f1.html" marginwidth="15px" marginheight="15px" >
<frame src="f2.html" >
</frameset>
<frameset cols="20%, *">
<frame src="menu.html" >
<frameset rows="50%, *">
<frame src="f3.html">
<frame src="f4.html">
</frameset>
</frameset>
</frameset>
</html>

```

Если требуется по нажатию на гиперссылку, расположенную в каком-либо фрейме, открыть указанный в атрибуте <href> документ в другом фрейме, то в данном [теге <a>](#) прописывается target="значение, которое является именем для требуемого фрейма".

```

<html>
<head>
<title>menu.html</title>
</head>
<body>
<a href="ff1.html" target="img1"> CC1 </a> <br>
<a href="ff2.html" target="img2"> CC2 </a> <br>
<a href="ff3.html" target="img3"> CC3 </a>
</body>
</html>

```

Плавающие фреймы

С помощью парного тега <iframe> можно вставлять фреймы в обычные html документы. Плавающие фреймы используются для минимизации места на странице.

Атрибуты:

- *Src* – URL открываемого в этом фрейме файла;
- *Scrolling* – определяет наличие полос прокруток (auto, yes, no);
- *Marginwidth* – отступ по горизонтали;
- *Marginheight* – отступ по вертикали;
- *Framborder* – задаёт показ границ между фреймами (yes, no), если Framborder= yes, то добавляются параметры:
 - Border* – толщина рамки
 - Bordercolor* – цвет рамки
- *Vspace* – расстояние между фреймами и текстом на странице по вертикали;
- *Hspace* – расстояние между фреймами и текстом на странице по горизонтали;
- *Width, height* – размеры фрейма;
- *Align* – выравнивание в плавающем фрейме (left, right, center)

Пример:

```

<html>
<head>
<title>menu</title>
</head>

```

```

<body>
<br><br><br>
<table border="1" width="80%">
<tr><td bgcolor="red">&nbsp;</td></tr>
<tr><td bgcolor="yellow">&nbsp;</td></tr>
</table><br><br>
<iframe src="ss.pdf" hspace="2px" vspace="20px" marginwidth="15px" marginheight="15px"
width="600px" height="600px">
</iframe>
</body>
</html>

```

Пример выполнения данного кода приведен на рисунке 5.



Рис. 5. Интеграция плавающего фрейма на Web-страницу.

Ниже приведен этот же код с немного измененными атрибутами элементов.

```

<html>
<head>
<title>плавающие фреймы</title>
</head>
<body>

```

```


<div style='position:absolute; left:560; top:10'>
<iframe src='ss.pdf' hspace="2px" vspace="20px" marginwidth="15px"
marginheight="15px">
</iframe></div>
<table border="1" bordercolorlight='green' bordercolordark='silver' bordercolor='navy'
width="80%">
<tr><td bgcolor='red'>&nbsp;</td></tr>
<tr><td bgcolor='yellow'>&nbsp;</td></tr>
</table>
</body>
</html>

```

Результат выполнения данного кода приведен на рисунке 6.



Рис. 6. Результат варьирования расположения тегов в HTML-разметке

8.2. ЗАДАНИЕ НА ПРАКТИЧЕСКУЮ РАБОТУ №6

1. Представить Web-страницу, используя ранее изученные теги работы с текстом, цветом, таблицами, гиперссылками и другими элементами Web-страниц для фреймовой html-разметки.
2. Представить несколько вариантов фреймовой организации Web-страниц;
3. Представить примеры использования «плавающих» фреймов с размещением в них документов различного типа.
4. Выполнить выданное практическое задание.

8.3. ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ И ПОДГОТОВКИ К ЗАЩИТЕ ПРАКТИЧЕСКОЙ РАБОТЫ №6

1. Опишите атрибуты тега frameset;
2. Опишите атрибуты тега frame;
3. Продемонстрируйте различные способы фреймовой организации Web-страниц;
4. Выполните сравнительный анализ табличной и фреймовой организации html-документа;
5. Отметьте достоинства и недостатки фреймовых документов;
6. Опишите предназначение «плавающих» фреймов и приведите программные коды их интеграции на Web-страницу.

[перейти к содержанию](#)

9. ЛИТЕРАТУРА

Основная литература

1. Филиппов, С.А. Основы современного веб-программирования: учебное пособие / С.А. Филиппов. - М.: МИФИ, 2011. - 160 с. - URL: <http://biblioclub.ru/index.php?page=book&id=232424>
2. Сычев, А.В. Перспективные технологии и языки веб-разработки / А.В. Сычев. - М.: Национальный Открытый Университет «ИНТУИТ», 2016. - 494 с. - URL: <http://biblioclub.ru/index.php?page=book&id=429078>
3. Диков, А.В. Веб-технологии HTML и CSS : учебное пособие / А.В. Диков. - 2-е изд. - М. : Директ-Медиа, 2012. - 78 с. - URL: <http://biblioclub.ru/index.php?page=book&id=96968>
4. Лыткина, Е.А. Основы языка HTML: учебное пособие / Е.А. Лыткина, А.Г. Глотова; Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северный (Арктический) федеральный университет имени М. В. Ломоносова», Министерство образования и науки Российской Федерации. - Архангельск: САФУ, 2014. - 104 с. - URL: <http://biblioclub.ru/index.php?page=book&id=436328>
5. Брокшмидт, К. Пользовательский интерфейс приложений для Windows 8, созданных с использованием HTML, CSS и JavaScript : учебный курс / К. Брокшмидт. - 2-е изд., исправ. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 396 с. - URL: <http://biblioclub.ru/index.php?page=book&id=429247>

Дополнительная литература

1. Применение каскадных таблиц стилей (CSS): курс / П. Храмцов, С. Брик, А. Русак, А. Сурин. - 2-е изд., исправ. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 82 с. - URL: <http://biblioclub.ru/index.php?page=book&id=429258>
2. Пархимович, М.Н. Основы интернет-технологий : учебное пособие / М.Н. Пархимович, А.А. Липницкий, В.А. Некрасова ; Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северный (Арктический) федеральный университет имени М. В. Ломоносова», Министерство образования и науки Российской Федерации. - Архангельск : ИПЦ САФУ, 2013. - 366 с. - URL: <http://biblioclub.ru/index.php?page=book&id=436379>
3. Богданов, М.Р. Перспективные языки веб-разработки / М.Р. Богданов. - 2-е изд., испр. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 265 с. - URL: <http://biblioclub.ru/index.php?page=book&id=428953>

Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Научная электронная библиотека <http://eLIBRARY.RU>.
2. Электронно-библиотечная система <http://e.lanbook.com>.
3. Электронно-библиотечная система «Университетская библиотека онлайн» <http://biblioclub.ru>.
4. Открытый архив номеров журнала «Информационные технологии» <http://novtex.ru/IT/arhiv.htm>
5. Открытый архив номеров журнала «Мир ПК» <http://www.osp.ru/pcworld/archive>

[перейти к содержанию](#)