

Белгородский государственный университет

**Лабораторный практикум
по Web-технологиям
Часть 1. Основы HTML технологий
Лабораторный практикум**

БЕЛГОРОД 2003

Белгородский государственный университет

Лабораторный практикум
по Web-технологиям
Часть 1. Основы HTML технологий

Лабораторный практикум

БЕЛГОРОД 2003

ББК 32. 81 я73

Л12

Авторы- составители

М. Ф. ТУБОЛЬЦЕВ

Н. П. ПУТИВЦЕВА

И. В. ГУРЬЯНОВА

О. В. НЕМЫКИНА

Рецензенты:

доцент кафедры математических методов и информационных технологий в экономике управления **Перлов Ю. М.**

доцент кафедры информационных систем и технологий Белгородского Университета Потребительской Кооперации **Куртов Н. И.**

Лабораторный практикум по Web-технологиям. Часть 1. Основы HTML технологий /

М. Ф. ТУБОЛЬЦЕВ, Н. П. ПУТИВЦЕВА, И. В. ГУРЬЯНОВА, О. В. НЕМЫКИНА. – Белгород: Изд-во БелГУ, 2003.

В лабораторном практикуме рассматриваются основы html-технологий. Лабораторный практикум содержит методические указания и задания лабораторных работ, а также подробно разобранные примеры и справочный материал. Данное пособие подходит как для самостоятельного изучения, так и для проведения занятий со студентами как экономических, так и иных специальностей в рамках курса «Информатика» и «Информационные технологии». Перечень и содержание лабораторных работ соответствует действующему стандарту HTML 4.0.

Содержание.

Предисловие 5

Введение в язык HTML 7

Структура HTML-документа 9

Функциональные блочные элементы 10

Лабораторная работа №1 11

Текстовое оформление страниц 11

Вставка изображений в html-страницы 14

Выравнивание изображений 14

Лабораторная работа №2 18

Создание гиперссылок 18

Ссылки внутри страницы 19

Ссылка на новое окно 19

Лабораторная работа №3 21

Создание списков 21

Нумерованные списки 21

Маркированные списки 22

Лабораторная работа № 4 24

Создание таблиц 24

Параметры таблицы 24

Лабораторная работа №5 30

Создание фреймов 30

Лабораторная работа №6 33

Каскадные таблицы стилей 33

CSS Текст 33

Свойства текста 34

Границы и рамки 35

Типы рамок 36

Приложения 40

Приложение 1 40

Названия цветов 40

Приложение 2 44

Краткий справочник по тегам 44

Предисловие

В последние два десятилетия Интернет прочно вошел в нашу жизнь. В наше время он охватывает почти все сферы жизни: культура, искусство, бизнес. Наряду с обычными музеями, библиотеками, магазинами все большую популярность во всем мире завоевывают аналогичные Интернет-ресурсы. Даже бизнес становится виртуальным: создаются Web-представительства фирм, электронные магазины и торговые площадки, электронные банки, которые дают возможность осуществлять типичные для этих сфер операции с большей скоростью. В связи с этим профессия Web-дизайнера и Web-программиста стала востребованной и у нас в России. Интернет-технологии создания указанных web-ресурсов требует владения такими языками для написания серверных скриптов, как Perl, PHP, ASP, владения языком XML, а также умения создавать динамические страницы и сайты, используя скрипты VBScript и JavaScript, которые позволяют осуществлять вычисления, работу с датой и временем, изменение элементов страницы по желанию пользователя.

Perl – язык для написания CGI-скриптов (Common GateWay Interface). CGI-скрипт – это программа, которая выполняется на Web-сервере по запросу посетителя сайта. CGI – это не язык программирования, а специальный интерфейс, с помощью которого и происходит запуск скрипта и взаимодействие с ним.

PHP-скрипт (Hypertext Preprocessor) – скрипт, написанный на языке PHP и внедренный в HTML-документ.

ASP-скрипт (Active Server Pages) – скрипт, написанный на языке Visual Basic Scripting Edition и внедренный в HTML-документ.

Но это потом. А на первом этапе необходимо научиться создавать простые HTML-страницы. Не стоит считать, что в этом нет необходимости. Простого HTML вполне хватит для двухстраничной коммерции, которая распространяется сейчас по всему миру. HTML позволяет создать «заготовки» страниц, а уже потом можно добавлять в них различные скрипты. XML же – это надстройка над HTML. Поэтому без изучения HTML не обойтись. Итак, HTML – Hypertext Markup Language – язык разметки гипертекста. Гипертекст – расширенный текст, который может содержать в себе текст, иллюстрации, различные внедренные объекты и ссылки на другие ресурсы или на другие Web-страницы.

При передаче информации через Интернет используются как раз HTML-страницы – файлы *.htm и *.html, поскольку они содержат в себе гипертекст, имеют размер меньше, чем текстовые или иные файлы и для их просмотра нужен только браузер.

HTML-документ представляет собой обычный текстовый документ с управляющими конструкциями языка HTML – тегами, которые и производят действия «форматирования» над текстовыми блоками и осуществляют вставку различных объектов в документ.

Создавать HTML-страницы можно вручную – путем самостоятельного написания тегов в текстовом редакторе или используя Web-редакторы с автоматизацией ввода нужных тегов – например, WebEdit, Arachnophilia или же используя программы, которые создают теги сами в ответ на действие пользователя на

странице – по принципу What You See Is What You Get – например, MS Word, MS Frontpage. Но они задают абсолютное форматирование и часто дописывают много лишнего кода к странице. К тому же для создания более уникальных страниц ими не обойдешься, для этого лучше писать теги HTML самому и использовать каскадные таблицы стилей.

Мы предлагаем использовать для создания Web-страниц блокнот, который входит в стандартный набор любой из операционных систем Windows и не требует предварительного изучения.

Наш практикум содержит в себе все основные разделы, необходимые для создания Web-страниц. В каждом разделе есть примеры использования тегов, показан результат действия этих тегов, а также представлены лабораторные работы. Практикум можно использовать для самостоятельного изучения технологии создания статических Web-страниц студентам любых специальностей или тем, кто хочет самостоятельно изучить основы языка HTML.

Введение в язык HTML

Автономные Web-документы используют язык HTML (HypertextMarkup Language - язык разметки гипертекста). Гипертекст, то есть расширенный текст, включает дополнительные элементы: иллюстрации, ссылки, вставные объекты. Под разметкой понимается использование специальных кодов, легко отделяемых от смыслового содержания документа и используемых для реализации гипертекста. Применение этих кодов подчиняется строгим правилам, определяемым спецификацией языка HTML.

HTML -документ - это файл, содержащий обыкновенный текст со специальными командами. Такой файл может быть подготовлен в произвольном текстовом редакторе (существуют, однако, специальные программы-конверторы и HTML-редакторы).

HTML -документ состоит из содержимого, то есть собственно полезной информации, и команд, задающих структуру.

Каждая команда (управляющая конструкция) HTML -документа (тег) должна заключаться в угловые скобки - вот так: <тег>. Чаще всего в документе встречаются парные теги (открывающий и соответствующий ему закрывающий), так как браузеру необходимо знать область действия тега. Существуют и одиночные теги, однако, используются они только там, где область действия очевидна и дополнительной информации не требуется (ясно, например, что если мы встретили тег "начало абзаца" (<P>), то предыдущий абзац уже закончился). В сомнительном же случае лучше перестраховаться и поставить закрывающий парный тег, иначе документ может оказаться нечитаемым. Открывающий и закрывающий теги называются одинаково и отличаются друг от друга только символом "наклонная черта" или "слэш" - "/", который ставится сразу после открывающей угловой скобки закрывающего тега. Закрытие парных тегов выполняется так, чтобы соблюдались правила вложения.

```
<B><I>На этот текст воздействуют два тега</I></B>
```

Кроме того, тег может включать атрибут, дающий дополнительную информацию браузеру. Например, при помощи атрибута можно попросить браузер изменить величину шрифта, ориентацию изображения по отношению к строке следующего за ним текста, поменять цвет фона документа и т. д. В парных тегах атрибуты добавляются только к открывающему тегу. Атрибуты представляют собой дополнительные ключевые слова, отделяемые от ключевого слова, определяющего тег, и от других атрибутов пробелами и размещаемые до завершающего тег символа ">". Способ применения некоторых атрибутов требует указания значения атрибута. Значение атрибута отделяется от ключевого слова атрибута символом "=" (знак равенства) и заключается в кавычки.

```
<H1 ALIGN="LEFT">
```

Язык HTML в большинстве случаев совершенно равнодушен к регистру, в котором набираются теги. Скажем, браузеру совершенно все равно, наберете вы тег, служащий для рисования горизонтальной линии, как **<HR>** или **<hr>** - эффект будет один и тот же.

HTML не признает никакого дополнительного форматирования текста, кроме как с помощью тегов. В результате текст, превосходно смотрящийся в текстовом редакторе, в окне браузера сольется в единую нечитаемую массу. Так, на месте нескольких пробелов будет лишь один пробел. Исчезнут все заголовки, пустые строки, деление текста на абзацы. Без HTML -тегов браузер просто игнорирует все элементы форматирования.

Определение HTML как языка разметки основывается на том, что при удалении из документа всех тегов получается текстовый документ, совершенно эквивалентный по содержанию исходному гипертекстовому документу. Таким образом, при отображении документа HTML сами теги не отображаются, но влияют на способ отображения остальной части документа.

Структура HTML-документа

```
<HTML>
<HEAD>
<TITLE> Моя домашняя страница </TITLE>
</HEAD>
<BODY>
</BODY>
</HTML>
```

Первый тег, который вы здесь видите, **<HTML>**, сообщает браузеру о том, что он имеет дело именно с документом в формате HTML. Тег **<HTML>** и парный ему закрывающий тег **</HTML>** можно считать как бы "конвертом", в который помещается весь документ. Любой HTML -документ состоит из заголовка, который задается при помощи тега **<TITLE>**, и тела документа, который определяется тегом **<BODY>**. В заголовке документа размещается служебная информация, комментарии автора и заголовок страницы, заключаемый в теги **<TITLE>**. Заголовок, вписанный между тегами **<TITLE>**, в основное текстовое поле браузеру не попадает, а, как правило, размещается в заголовке окна браузера. HTML-документ - это просто текстовый файл с расширением ***.htm** (Unix-системы могут содержать файлы с расширением ***.html**). Большинство элементов языка HTML описывает части содержания документа и помещается между тегами **<BODY>** и **</BODY>**, то есть, внутри структурного элемента BODY. Такие элементы делят на блочные и текстовые. Блочные элементы относятся к частям текста уровня абзаца. Текстовые элементы описывают свойства отдельных фраз и еще более мелких частей текста. Теперь можно сформулировать правила вложения элементов.

- Элементы не должны пересекаться. Другими словами, если открывающий тег располагается внутри элемента, то и соответствующий закрывающий тег должен располагаться внутри этого же элемента.

- Блочные элементы могут содержать вложенные блочные и текстовые элементы.

- Текстовые элементы могут содержать вложенные текстовые элементы.

- Текстовые элементы не могут содержать вложенные блочные элементы.

Строго говоря, все правила языка HTML можно рассматривать исключительно как "пожелания". Средство, используемое для отображения Web-документа, сделает все возможное, чтобы истолковать разметку наиболее разумным образом. Тем не менее, гарантию правильного воспроизведения документа дает только неукоснительное следование требованиям спецификации языка.

Функциональные блочные элементы

В большинстве документов основными функциональными элементами являются заголовки и абзацы. Язык HTML поддерживает шесть уровней заголовков. Они задаются при помощи парных тегов от **<H1>** до **<H6>**. При отображении Web-документа на экране компьютера эти элементы показываются при помощи шрифтов разного размера.

Обычные абзацы задаются с помощью парного тега **<P>**. Язык HTML не содержит средств для создания абзацного отступа ("красной строки"), поэтому при отображении на экране компьютера абзацы разделяются пустой строкой. Закрывающий тег **</P>** рассматривается как необязательный. Подразумевается, что он стоит перед тегом, который задает начало очередного абзаца документа. Например:

```
<h1>Заголовок</h1>
<p>Первый абзац
<p>Второй абзац
<h2>Заголовок второго уровня</h2>
```

Заголовок

Первый абзац

Второй абзац

Заголовок второго уровня

Следствием наличия специального тега, определяющего абзац, является тот факт, что обычного символа конца строки, вводимого по нажатию клавиши ENTER, для создания абзацного отступа недостаточно. Язык HTML рассматривает символы конца строки и пробелы особым образом. Любая последовательность, состоящая только из пробелов и символов конца строки, при отображении документа рассматривается как одиночный пробел. Это, в частности, означает, что символ конца строки даже не осуществляет перехода на новую строку (для этой цели используется текстовый элемент, задаваемый непарным тегом **
**).

Лабораторная работа №1

Текстовое оформление страниц

С помощью тега `` можно изменить параметры шрифта. Для тега используются следующие параметры: **face**, **size** и **color**.

Параметр **Face** служит для задания гарнитуры шрифтов использующихся для текста. Названий шрифтов можно указать несколько, через запятую. В этом случае, если первый указанный шрифт не будет найден, будет использоваться следующий по списку.

Пример 1. Использование параметра face

```
<font face="Arial, Helvetica, sans-serif">Текст будет написан шрифтом Arial.</font>
```

Size задает размер шрифта в условных единицах от 1 до 7. Средний размер, используемый по умолчанию принят 3. Размер шрифта можно указывать как абсолютной величиной (например, `size=4`), так и относительной (например, `size=+1`, `size=-1`). В последнем случае размер изменяется относительно базового.

Пример 2. Задание размера шрифта

```
<font size=1>Шрифт размера 1</font><br>
<font size=2>Шрифт размера 2</font><br>
<font size=3>Шрифт размера 3</font><br>
<font size=4>Шрифт размера 4</font><br>
<font size=5>Шрифт размера 5</font><br>
<font size=6>Шрифт размера 6</font><br>
<font size=7>Шрифт размера 7</font><br>
```

Шрифт размера 1

Шрифт размера 2

Шрифт размера 3

Шрифт размера 4

Шрифт размера 5

Шрифт размера 6

Шрифт размера 7

Color определяет цвет текста, который можно задавать с помощью названий цветов или в шестнадцатеричном формате.

Пример 3. Изменение цвета текста

```
<font size=5 color=red face=Arial>П</font>ервая буква этого предложения будет написана шрифтом Arial, красным цветом и увеличенной.</font>
```

Первая буква этого предложения будет написана шрифтом Arial, красным цветом и увеличенной.

Видоизменение текста - средства его форматирования, такие как выбор начертания шрифта и использование эффектов, позволяющих менять вид текста. В таблице перечислены основные теги, которые применяются для изменения оформления текста.

Код HTML	Описание	Пример
<code>Текст</code>	Жирный текст	Текст
<code><i>Текст</i></code>	Курсивное начертание текста	<i>Текст</i>
<code><u>Текст</u></code>	Подчеркнутый текст	<u>Текст</u>
<code><sup>Текст</sup></code>	Верхний индекс	$e=mc^2$
<code><sub>Текст</sub></code>	Нижний индекс	H ₂ O
<code><strike>Текст</strike></code>	Зачеркнутый текст	Текст
<code><pre>Текст</pre></code>	Текст пишется как есть, включая все пробелы	Текст
<code>Текст</code>	Курсивный текст	<i>Текст</i>
<code>Текст</code>	Жирный текст	Текст

Обычно для создания верхнего или нижнего индекса используется тег **small**, делающий индекс меньше по размеру основного шрифта.

Пример 4. Создание нижнего индекса

`Формула серной кислоты:`

`<i>H_{<small>2</small>}SO_{<small>4</small>}</i>`

Формула серной кислоты:

H_2SO_4

Выравнивание текста

Выравнивание текста определяет его внешний вид и ориентацию краев абзаца и может выполняться по левому, правому краю, по центру или по ширине.

Код HTML	Описание	Пример
<code><p>Текст</p></code>	Добавляет новый параграф, по умолчанию выровненный по левому краю. Перед параграфом автоматически добавляется пустая строка.	Текст
<code><p align=left>Текст</p></code>	Выравнивание по левому краю.	Текст
<code><p align=right>Текст</p></code>	Выравнивание по правому краю.	Текст
<code><p align=center>Текст</p></code>	Выравнивание по центру.	Текст
<code><p align=justify>Текст</p></code>	Выравнивание по ширине.	Текст по ширине

Задание

Оформите текст, как показано ниже:

Октябрь уж наступил,

Уж роца отряхает

Последние листы

С нагих своих ветвей.

Дохнул осенний *хлад,*

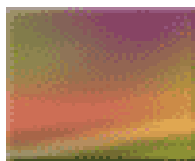
Дорога промерзает,

Журча, ^{еще} бежит

За мельницу ручей .

Вставка изображений в html-страницы

Для встраивания изображения в документ используется тег **IMG**, имеющий единственный обязательный параметр **src**, который определяет адрес файла с картинкой.



Файл с рисунком, изображенным ниже, называется *sample.gif* и размещается в папке *images* корня сайта.

Для указания адреса изображения можно задавать как абсолютный, так и относительный адрес.






Пример 1 Вставка изображения в документ

```
<html>
<body>
 - это абсолютный адрес размещения изображения
 - адрес размещения изображения относительно корня сайта
 - адрес размещения изображения относительно текущего HTML-документа
</body>
</html>
```

Выравнивание изображений

Для изображений можно указывать их положение относительно текста или других изображений на веб-странице. Способ выравнивания изображений задается параметром **align** тега **IMG**. В таблице перечислены возможные значения этого параметра и результат его использования.

Код HTML	Описание	Пример
<code></code>	Верхняя граница изображения выравнивается по самому высокому текстовому элементу текущей строки.	Lorem ipsum dolor sit amet,  consectetur adipiscing elit...
<code></code>	Верхняя граница изображения выравнивается по самому высокому элементу текущей строки.	ipsum  Lorem ipsum dolor sit amet, consectetur adipiscing elit...
<code></code>	Выравнивание середины изображения по базовой линии текущей строки.	Lorem ipsum dolor sit amet,  consec-

		tetuer adipiscing elit...
<code></code>	Выравнивание середины изображения по середине текущей строки.	 Lorem ipsum dolor sit amet, consectetuer adipiscing elit...
<code></code>	Выравнивание изображения по базовой линии текущей строки.	 Lorem ipsum dolor sit amet, consectetuer adipiscing elit...
<code></code>	Выравнивание нижней границы изображения по окружающему тексту..	 Lorem ipsum dolor sit amet, consectetuer adipiscing elit...
<code></code>	Выравнивает изображение по левому краю окна.	 Lorem ipsum dolor sit amet, consectetuer adipiscing elit...
<code></code>	Выравнивает изображение по правому краю окна.	 Lorem ipsum dolor sit amet, consectetuer adipiscing elit...

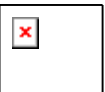
Наиболее популярные параметры – **left** и **right**, создающие обтекание текста вокруг изображения. Чтобы текст не прилегал плотно к рисунку, рекомендуется в теге **IMG** добавить параметр **hspace** и **vspace**, задающих расстояние до текста в пикселах.

Пример 2 Обтекание текста вокруг рисунка

```

<html>
<body>
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diem nonummy
nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisis enim ad
minim veniam, quis nostrud exerci tution ullamcorper suscipit lobortis nisl ut aliquip ex ea
commodo consequat.
</body>
</html>

```



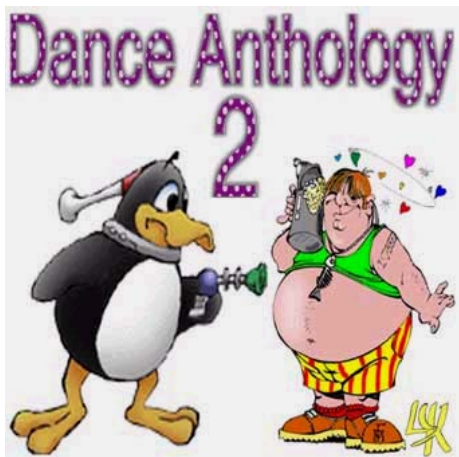
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diem nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisis enim ad minim veniam, quis nostrud exerci tution ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Задание

Выровняйте рисунки и текст так, как показано ниже:



Этот рисунок находится слева и текст выравнивается по левой границе. Эффект достигается без таблицы



Этот рисунок находится слева и текст выравнивается по центру. Эффект достигается без таблицы

Этот рисунок находится справа и текст выравнивается по правой границе. Эффект достигается без таблицы



Лабораторная работа №2

Создание гиперссылок

Как сделать ссылку

Для создания ссылки необходимо сообщить браузеру, что является ссылкой, а также указать адрес документа, на который следует сделать ссылку. Оба действия выполняются с помощью тега **A**, который имеет единственный параметр **href**. В качестве значения используется адрес документа (**URL**).

Адрес ссылки может быть абсолютным и относительным. Абсолютные адреса работают везде и всюду независимо от имени сайта или веб-страницы, где прописана ссылка.

Пример 1. Использование абсолютных ссылок

```
<html>
<body>
<a href=www.yandex.ru>Поисковая система Яндекс</a>
</body>
</html>
```

Относительные ссылки, как следует из их названия, построены относительно текущего документа или адреса. Примеры таких адресов:

1. /
2. /demo/
3. /images/pic.gif
4. ../help/me.html
5. manual/info.html

Первые две ссылки называются неполные и указывают веб-серверу загружать файл `index.html` (или `default.html`) находящемуся в корне сайта (пример 1) или папке `demo` (пример 2). Если файл `index.html` отсутствует, браузер, как правило, показывает список файлов, находящихся в данном каталоге. Слэш перед адресом говорит о том, что адресация начинается от корня сайта (пример 3), двоеточие - перейти на уровень выше в списке каталогов сайта (пример 4).

Пример 2. Использование относительных ссылок

```
<html>
<body>
<a href=images/xxx.jpg>Посмотрите на мою фотографию!</a><br>
<a href=tip.html>Как сделать такое же фото?</a>
</body>
</html>
```

Ссылки внутри страницы

Большие документы читаются лучше, если они имеют оглавление со ссылками на соответствующие разделы. Для создания ссылки следует вначале сделать закладку в соответствующем месте и дать ей имя при помощи параметра `name` тега **A**.

Пример 1. Создание внутренней ссылки

```
<html>
<body>
  <a name=top></a> Друг уронил утюг в унитаз. И разбил его. Причем так разбил, что по назначению унитаз и использовать никак нельзя. Мгновением назад только что вот все было хорошо и вот уже дыра, да такая, что можно забыть, что есть такой предмет в доме. Махнул рукой нечаянно, а потом мучайся...
  <a href=#top>Наверх</a>
</body>
</html>
```

Между тегам `` и `` отсутствует текст, так как требуется лишь указать местоположение перехода по ссылке, находящейся внизу страницы. 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 надлежит указать ее адрес и в конце добавить символ решетки `#` и имя закладки.

Пример 2. Ссылка на закладку из другой веб-страницы

```
<html>
<body>
  <a href=text.html#bottom>Перейти к нижней части текста</a>
</body>
</html>
```

Ссылка на новое окно

Если требуется сделать ссылку на документ, который открывается в новом окне браузера, используется параметр `target=_blank` тега **A**.

Создание нового окна обычно требуется в случаях, когда делается ссылка на другой сайт, в остальном лучше открывать документы в текущем окне, поскольку обилие окон может сбить читателя с толку.

Так как ссылки на текущее или новое окно ничем не отличаются друг от друга, на некоторых сайтах рядом со ссылкой ставят специальную иконку, показывающую, что документ открывается в новом окне.

Пример 2. Создание ссылки на новое окно

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

```
<a href=www.bsu.edu.ru>Обычная ссылка на сайт www.bsu.edu.ru</a><br>
<a href=www.bsu.edu.ru target=_blank>Ссылка открывает новое окно на сайт
www.bsu.edu.ru</a>
</body>
</html>
```

[Обычная ссылка на сайт www.bsu.edu.ru](http://www.bsu.edu.ru)
[Ссылка открывает новое окно на сайт www.bsu.edu.ru](http://www.bsu.edu.ru)

Задание

Используя внутренние ссылки, создать следующий словарь терминов:

Словарь терминов

[А](#) [Б](#) [В](#) [Г](#) [Д](#) [Е](#)

А

АВТЕНТИЧЕСКИЙ КАДАНС

кадансовый оборот, в котором заключительная тоническая гармония предваряется доминантовой

АЛИКВОТНЫЕ СТРУНЫ

резонирующие струны, к которым исполнитель не прикасается во время игры

АТАКТА

гармонический элемент на басу нижнего или верхнего вводного тона

[В начало](#)

Б

БАГАТЕЛЬ

небольшая нетрудная для исполнения пьеса

БАРТОКОВСКОЕ ПИЦЦИКАТО

сильный щипок струны с последующим ударом струны о гриф

БОНАНГ

набор из 10-12 гонгов разного размера

[В начало](#)

В

[В начало](#)

Г

[В начало](#)

Д

[В начало](#)

Лабораторная работа №3

Создание списков

Нумерованные списки

Нумерованные списки представляют собой набор элементов с их порядковыми номерами. Вид и тип нумерации зависит от параметров тега **OL**, который и используется для создания списка. В качестве маркеров могут быть следующие значения: арабские цифры заглавные латинские буквы прописные латинские буквы заглавные римские цифры прописные римские цифры

Ниже, в таблице приведены различные параметры тега **OL** и результат их применения.

Код HTML	Пример
<pre> текст текст текст </pre>	Нумерованный список с параметрами по умолчанию: 1. текст 2. текст 3. текст
<pre><ol start="5"></pre>	Нумерованный список начинающийся с пяти: 5. текст 6. текст 7. текст
<pre><ol type="A"></pre>	Нумерованный список с заглавными буквами латинского алфавита: A. текст B. текст C. текст
<pre><ol type="a"></pre>	Нумерованный список с прописными буквами латинского алфавита: a текст b текст c текст
<pre><ol type="I"></pre>	Нумерованный список с римскими буквами: I. текст II. текст III. текст
<pre><ol type="i"></pre>	Нумерованный список с прописными римскими буквами: i. текст ii. текст iii. текст
<pre><ol type="1"></pre>	Нумерованный список с арабскими цифрами: 1. текст 2. текст 3. текст

<pre><ol type="I" start="7"></pre>	<p>Список с римскими цифрами начинающийся с семи:</p> <p>IV. текст</p> <p>V. текст</p> <p>VI. текст</p>
--	---

Маркированные списки

Маркированные списки позволяют разбить большой текст на отдельные блоки. Тем самым привлекается внимание читателя к тексту и повышается его читабельность. С учетом худшего восприятия текста с экрана монитора, чем печатного варианта, это является весьма полезным приемом.

Для установки маркированного списка используется тег **UL** и **LI** (Пример 1)

Пример 1. Создание маркированного списка

```
<html>
<head>
<body>
Что следует учитывать при тестировании сайта:
<ul>
<li>работоспособность всех ссылок</li>
<li>поддержку разных браузеров</li>
<li>читабельность текста</li>
</ul>
</body>
</html>
```

Ниже показан результат примера 1.

<p>Что следует учитывать при тестировании сайта:</p> <ul style="list-style-type: none"> • работоспособность всех ссылок • поддержку разных браузеров • читабельность текста
--

Обратите внимание, что у маркированного текста появляются отступы сверху и снизу. Чтобы от них избавиться, список можно делать без тега **UL**. При этом исчезнут и отступы текста перед маркерами.

Пример 2. Создание маркированного списка без отступов

```
<html>
<head>
<body>
Что следует учитывать при тестировании сайта:
<li>работоспособность всех ссылок</li>
<li>поддержку разных браузеров</li>
<li>читабельность текста</li>
</body>
</html>
```

Ниже показан результат примера 2.

Что следует учитывать при тестировании сайта: <ul style="list-style-type: none">• работоспособность всех ссылок• поддержку разных браузеров• читабельность текста

Маркеры могут принимать один из трех видов: круг (по умолчанию), окружность и квадрат. Для выбора типа маркера используется параметр **type="..."** тега **UL**. Вместо многоточия подставляется одно из трех значений указанных в таблице.

Код HTML	Пример
<code><ul type="disc"></code>	Что следует учитывать при тестировании сайта: <ul style="list-style-type: none">• работоспособность всех ссылок• поддержку разных браузеров• читабельность текста
<code><ul type="circle"></code>	Что следует учитывать при тестировании сайта: <ul style="list-style-type: none"><input type="checkbox"/> работоспособность всех ссылок<input type="checkbox"/> поддержку разных браузеров<input type="checkbox"/> читабельность текста
<code><ul type="square"></code>	Что следует учитывать при тестировании сайта: <ul style="list-style-type: none">■ работоспособность всех ссылок■ поддержку разных браузеров■ читабельность текста

С помощью CSS этот список можно расширить и вместо встроенных символов использовать в качестве маркера рисунок.

Лабораторная работа № 4

Создание таблиц

Таблица состоит из строк и столбцов ячеек, которые могут содержать текст и рисунки. Обычно таблицы используются для упорядочения и представления данных, однако возможности таблиц этим не ограничиваются. С помощью таблиц удобно верстать макеты страниц, расположив нужным образом фрагменты текста и изображений.

Для добавления таблицы на веб-страницу используется тег-контейнер **TABLE**. Таблица должна содержать хотя бы одну строку и колонку.

Для добавления строк используются теги **<tr>** и **</tr>**. Чтобы разделить строки на колонки применяются теги **<td>** и **</td>**.

Параметры таблицы

Для изменения вида и свойств таблицы используется множество параметров, которые добавляются в теге **TABLE**.

<table параметр1=... параметр2=...>

Описание параметров таблицы и их свойств описано ниже.

Свойство	Значение	Описание	Пример
align=	Left Right Center	Выравнивание таблицы	align=center
background=	URL	Фоновый рисунок	background=pic.gif
bgcolor=	#rrggbb	Цвет фона таблицы	bgcolor=#FF9900
border=	n	Толщина рамки в пикселах	border=2
bordercolor=	#rrggbb	Цвет рамки	border-color=#333333
bordercolor-dark=	#rrggbb	Тень рамки	bordercolor-dark=#f0f0f0
cellpadding=	n	Расстояние между ячейкой и ее содержимым	cellpadding=7
cellspacing=	n	Дистанция между ячейками	cellspacing=3
nowrap		Запрещает переносы строк в тексте	<table nowrap>
frame=	Void Above Below Lhs Rhs Hsides Vsides Box	Задание типа рамки таблицы	frame=hsides

valign=	Top Bottom	Выравнивание по высоте	valign=top
width=	n n%	Минимальная ширина таблицы, можно задавать в пикселах или процентах	width=90%
height	n n%	Минимальная высота таблицы, можно задавать в пикселах или процентах	height=18

Примечание

1. Таблица, если не указано особо, всегда выравнивается по левому краю;
2. Параметр **background**, отвечающий за рисунок фона, своеобразно понимается в разных браузерах. IE вставляет картинку во всю таблицу, если таблица по размеру больше фонового рисунка, он повторяется по горизонтали или вертикали. **Netscape** добавляет фоновое изображение в каждую ячейку таблицы;
3. По умолчанию, таблица выводится без рамки. Однако **Netscape** добавляет тонкую линию между ячеек. Чтобы ее не было, всегда указывайте параметр **border=0**;
4. Если ширина таблицы не указана, она подгоняется под содержание ячеек.

Задание 1

Создать 3 таблицы, используя приведенный ниже листинг

1.1 Создание таблицы

```

<HTML>
  <HEAD>
<TITLE>ТАБЛИЦА</TITLE>
  </HEAD>
<BODY>
  <TABLE border="2" align=center>
    <TR>
      <TD colspan=2 align=center>
        <B>Заголовок Таблицы.</B>
      </TD>
    </TR>
    <TR>
      <TD align="center">
        Первая ячейка первой строки
      </TD>
      <TD align="center">
        Вторая ячейка первой строки
      </TD>
    </TR>
  </TABLE>

```

```

<TD align="center">
    Первая ячейка второй строки
</TD>
<TD align="center">
    Вторая ячейка второй строки
</TD>
</TR>
</TABLE> <
</BODY>
</HTML>

```

Тег **<TABLE>** задает таблицу

Атрибуты **border="2"** и **align=center** задают, соответственно, размер границ таблицы и выравнивание ее по центру страницы

Тег **<TR>** задает строку таблицы

Тег **<TD>** задает ячейку таблицы

Атрибут **colspan=n** объединяет n ячеек по горизонтали (по столбцам)

В результате получится следующая таблица, состоящая из двух столбцов и двух строк:

Заголовок Таблицы.	
1 ячейка первой строки	2 ячейка первой строки
1 ячейка второй строки	2 ячейка второй строки

1.2. Сложная таблица

```

<HTML>
  <HEAD>
    <TITLE>СЛОЖНАЯ ТАБЛИЦА</TITLE>
  </HEAD>
  <BODY>
    <TABLE border="1" width="75%" align=center>
      <TR>
        <TD width="66%" colspan="2">
          <P align="center">
            Две ячейки, объединенные по горизонтали
          </TD >
        </TR >
        <TR >
          <TD width="33%" rowspan="2" valign="middle" >
            Две ячейки, объединенные по вертикали
          </TD >
          <TD width="33%">
            по левому краю
          </TD>
        </TR >
      <TR <

```

```

    <TD width="33%" <
    <P align="right" >по правому краю
    </TD >
</TR >
</TABLE >
</BODY >
</HTML >

```

Атрибут **colspan=n** объединяет n ячеек по горизонтали (по столбцам)

Атрибут **rowspan=n** объединяет n ячеек по вертикали (по строкам)

Атрибут **valign="middle"** выравнивает текст в ячейке по центру ячейки по вертикали

В результате получается следующая таблица:

Две ячейки, объединенные по горизонтали	
Две ячейки, объединенные по вертикали	по левому краю
	по правому краю

1.3. Более сложная таблица

```

<HTML>
<HEAD>
<TITLE>УЧЕБНАЯ ТАБЛИЦА</TITLE>
</HEAD>
<BODY>
<TABLE WIDTH="50%" BORDER="6" CELSPACING="6"
  CELLPADDING="20" ALIGN="center" BORDERCOLORLIGHT="Lime"
  BORDERCOLORDARK="Green" BGCOLOR="#DFFFFD">
<THEAD BGCOLOR="Aqua">
<TR><TH COLSPAN="3">УЧЕБНАЯ ТАБЛИЦА</TH></TR>
</THEAD>
<TBODY>
<TR>
  <TD WIDTH="33%">Это первая ячейка</TD>
  <TD WIDTH="33%">Это вторая ячейка</TD>
  <TD ROWSPAN="3">А это три ячейки третьего столбца
  объединились в одну большую</TD>
</TR>
<TR>
  <TD COLSPAN="2">Это единственная ячейка второй строки,
  объединяющая оба столбца</TD>
</TR>
<TR>
  <TD>Это первая ячейка третьей строки</TD>
  <TD WIDTH="33%">А это вторая ячейка третьей строки</TD>
</TR>
</TBODY>

```

```

<TFOOT BGCOLOR="Yellow">
<TR><TD COLSPAN="3" ALIGN="center">
<SMALL>конец</SMALL></TD></TR>
</TFOOT>
</TABLE>
</BODY>
</HTML>

```

Атрибут **CELLSPACING="6"** задает свободное пространство между ячейками таблицы

Атрибут **CELLPADDING="20"** задает свободное пространство между данными в ячейке и ее границами

Атрибут **BORDERCOLORLIGHT="Lime"** задает цвет левого и верхнего углов таблицы

Атрибут **BORDERCOLORDARK="Green"** задает цвет правого и нижнего углов таблицы

Атрибут **BGCOLOR="#DFFFFD"** задает цвет фона таблицы

В результате получается следующая таблица:

УЧЕБНАЯ ТАБЛИЦА		
Это первая ячейка	Это вторая ячейка	А это три ячейки третьего столбца объединились в одну большую
Это единственная ячейка второй строки, объединяющая оба столбца		
Это первая ячейка третьей строки	А это вторая ячейка третьей строки	
конец		

Задание 2

Создать турнирную таблицу, представленную ниже на рисунке.

Для задания кавычек используйте специальные символы **«**; ("«") и **»**; ("»"), для создания длинного тире используйте специальный символ **—** ("—").

ТУРНИРНАЯ ТАБЛИЦА - Microsoft Internet Explorer

Файл Правка Вид Избранное Сервис Справка

Назад Поиск Избранное Журнал

Адрес C:\Мои документы\HTML\ТУРНИРНАЯ ТАБЛИЦА.html

ЧЕМПИОНАТ ГОРОДА ПО ХОККЕЮ

ТАБЛИЦА РЕЗУЛЬТАТОВ

	Команда	1	2	3	4	5	6	Шайбы	Очки
1	«РОТОР»		6:4	2:3	7:3	2:1	0:0	17—11	7
2	«РОТАТОР»	4:6		5:4	5:3	5:5	2:2	21—20	6
3	«УРОЖАЙ»	3:2	4:5		0:1	11:2	14:4	32—14	6
4	«МАЧТА»	3:7	3:5	1:0		1:1	10:0	18—13	5
5	«МЕЧТА»	1:2	5:5	2:11	1:1		8:3	17—22	4
6	«ОКАЗИЯ»	0:0	2:2	4:14	0:10	3:8		9—34	2

Готово

Мой компьютер

Лабораторная работа №5

Создание фреймов

Фреймы - это окна независимого просмотра HTML-документов. Иногда бывает очень удобно использовать навигацию по странице в виде меню, оформленного в отдельном фрейме, и основного окна, где будет представлена вся основная информация, определяемая действиями пользователя в области меню.

Для создания фрейма используется тег **FRAMESET**, который заменяет тег **BODY** в документе и используется для разделения экрана на области. Внутри данного тега находятся теги **FRAME**, которые указывают на HTML-документ, предназначенный для загрузки в область.

Пример 1. Создание простого фрейма

```
<HTML>
<FRAMESET cols="30%, 70%" border=0>
<FRAME src="menu.html" name="Menu">
<FRAME src="main.html" name="Main">
</FRAMESET>
</HTML>
```

В приведенном примере присутствует объявление двух фреймов, которые будут располагаться вертикальными полосами и занимать соответственно 30 и 70 процентов рабочей области. Вертикальное расположение устанавливается параметром **cols="..."**, а для горизонтальных полос используют параметр **rows="..."**. Параметр **border="..."** определяет границу между фреймами. Определение документа, изначально загружаемого при открытии этого файла (этот документ является HTML-страницей.), задается атрибутом тега **<FRAME> src="..."**, при этом необходимо указать параметр **name="..."**, позволяющий задать "имя" созданной области в виде последовательности латинских букв и цифр, использованной как значение этого атрибута. Это имя можно использовать, чтобы загружать новые документы в ранее созданную область. Для этого в тег **<A>**, определяющий гиперссылку, необходимо добавить атрибут **target="..."**, значение которого совпадает с ранее определенным именем области. При переходе по данной гиперссылке новый документ загрузится в указанный фрейм. Например, предположим, что начальная страница Web-узла состоит из двух фреймов: слева располагается навигационная панель, а справа - текущая страница. Если правой области присвоено имя, используемое во всех ссылках, имеющих в левой области, то щелчок на любой ссылке навигационной панели приведет к обновлению информации в соседней области, оставляя навигационную панель без изменений. В данном документе находятся только ссылки на другие (существующие) HTML-документы, которые будут загружены браузером при открытии страницы с фреймами. Файл **menu.html** имеет следующую структуру:

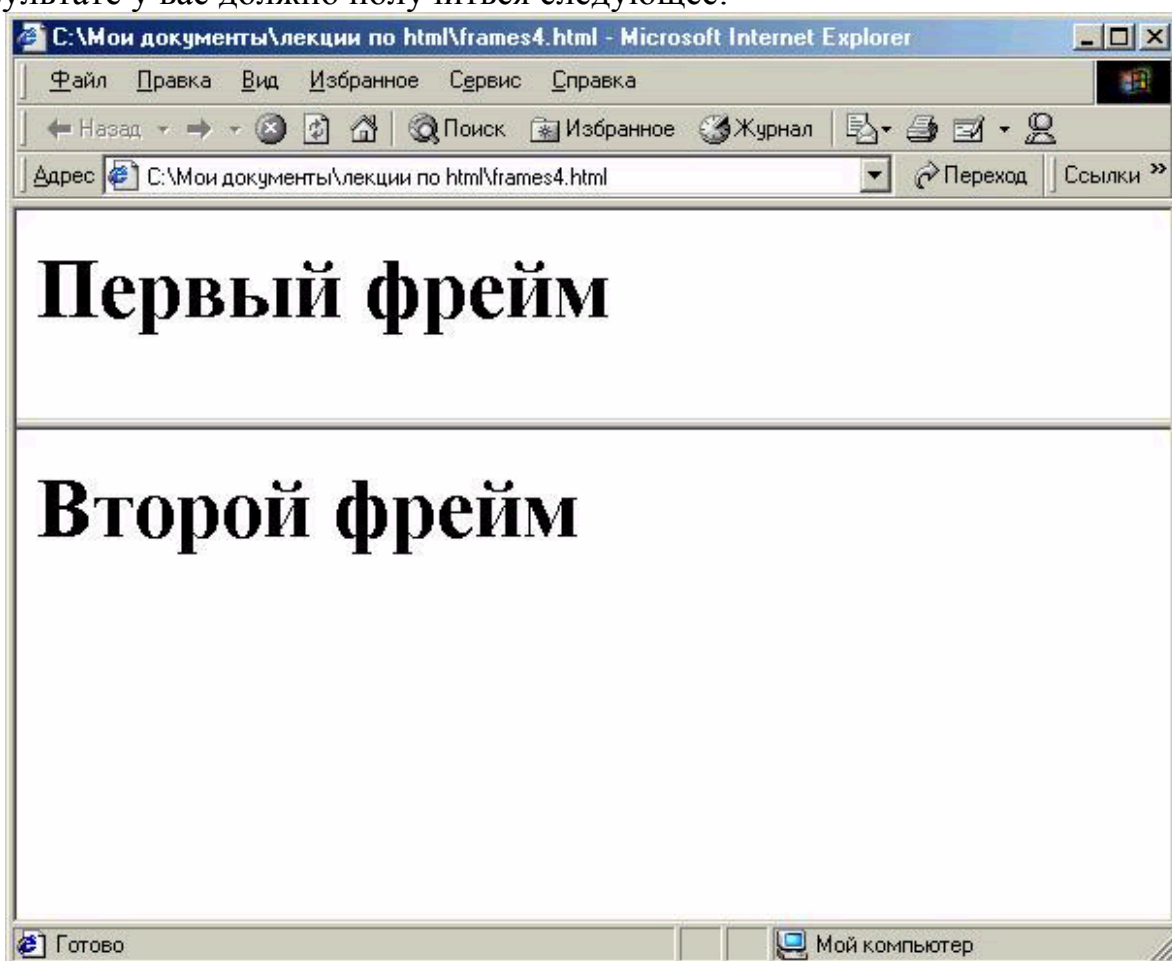
```
HTML>
```

```
<HEAD>
<TITLE></TITLE>
</HEAD>
<BODY>
<H1>Первый фрейм</H1>
</BODY>
</HTML>
```

Файл main.html также имеет указанную выше структуру. Следует сказать, что оба этих файла должны располагаться в том же каталоге, где расположен файл с фреймами.

Задание 1

Расположить эти фреймы по вертикали, используя атрибут **rows="..."**. В результате у вас должно получиться следующее:



Задание 2

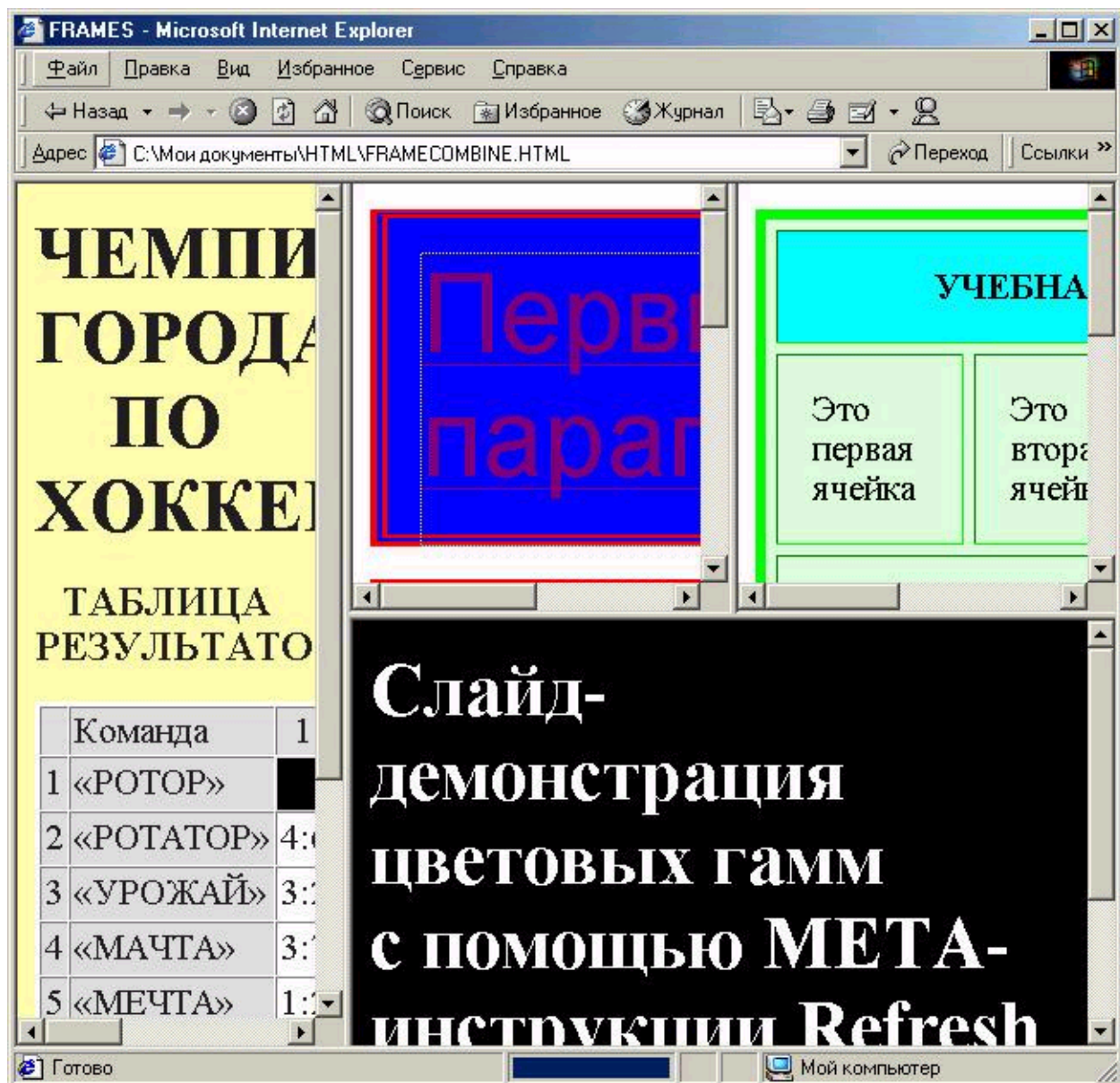
Распространенное явление - комбинация вертикальных и горизонтальных фреймов.

```
<FRAMESET cols="*, 55%">  символ * означает все оставшееся место
<FRAME src="homepage.htm" name="Frame1">
<FRAMESET rows="15%, 15%, 70%">
<FRAME src="menu.htm" name=" Frame2">
<FRAME src="menu2.htm" name=" Frame3">
<FRAME src="menu3.htm" name=" Frame4">
```

</FRAMESET>

</FRAMESET>

Используя вложенные теги **FRAMESET**, разбейте страницу следующим образом:



Задание 3

Используя атрибут **target="..."**, сделать в фреймах гиперссылки, по нажатию на которые в эти же или в другие фреймы произойдет загрузка других страниц.

Лабораторная работа №6 Каскадные таблицы стилей CSS Текст

С помощью CSS можно определять стиль и вид текста. Аналогично тому, что используется тег **FONT**, задающий свойства шрифта, но стили обладают большими возможностями и позволяют сократить код HTML.

Свойства шрифта

Свойство	Значение	NC	IE	Описание	Пример
font-family	имя шрифта	4+	4+	Задает список шрифтов	p {font-family: Arial, serif}
font-style	Normal Italic Oblique	4+ 4+	4+ 4+	Нормальный шрифт Курсив Наклонный шрифт	p {font-style: italic}
font-variant	Normal Small-caps		4+ 4+	Капитель (особые прописные буквы)	p {font-variant: small-caps}
font-weight	Normal Lighter Bold Bolder 100-900	4+ 4W 4+ 4W 4+	4+ 4+ 4+ 4+	Нормальная жирность Светлое начертание Полужирный Жирный 100-светлый шрифт, 900-самый жирный	p {font-weight: bold}
font-size	Размер шрифта Normal Pt Px %	4+ 4+ 4+ 4+	4+ 4+ 4+ 4+	нормальный размер пункты пикселы проценты	font-size: normal font-size: 12pt font-size: 12px font-size: 120%

NC - Netscape Communicator, **IE** - Microsoft Internet Explorer; **4+** означает, что свойство поддерживается в браузере 4 версии и выше. **4W** - работает только под Windows.

Замечание:

Когда размер шрифта задается абсолютными значениями, т.е. указывается конкретное значение шрифта в пунктах или пикселах, то изменить эту величину с помощью опции браузера *Вид | Размер шрифта* невозможно. Если шрифт установлен слишком мелким, то исправить этот недостаток читателю простыми

средствами не представляется возможным. Поэтому лучше задавать размер шрифта в процентах (Пример 1).

Пример 1. Задание свойств шрифта с помощью CSS

```
<html>
<style>
H1 { font-family: Arial, Helvetica, Verdana, sans-serif; font-size: 150%; font-weight:
light }
</style>
<body>
<H1>Заголовок</H1>
Обычный текст
</body>
</html>
```

Ниже приведен результат примера 1.

Заголовок
Обычный текст

В таблице Примера 2 даны некоторые параметры и результат их применения.

Пример 2. Результат использования различных параметров шрифтов

Пример	Пример	Пример	Пример	Пример
font-family: Verdana, sans- serif; font-size: 120%; font- weight: light	font-size: large; font-weight: bold	font-family: Arial, sans- serif; font-size: x-small; font- weight: bold	font-variant: small-caps	font-style: italic; font- weight: bold

Свойства текста

Кроме изменения параметров шрифтов, можно управлять и свойствами всего текста. Значения свойств приведены в таблице.

Свойство	Значение	NC	IE	Описание	Пример
line-height	Normal	4W	4+	Интерлиньяж (межстрочный интервал)	line-height: normal
	Множитель	4+	4+		line-height: 1.5
	Точно	4+	4+		line-height: 12px
	%	4+	4+		line-height: 120%
text-decoration	None	4+	4+	Убрать все оформление	text-decoration: none
	Underline	4+	4+	Подчеркивание	
	Overline	4+	4W	Линия над тек-	
	line-through	4+	4+	стом	
	Blink	4+		Перечеркивание	

				Мигание текста	
text-transform	None	4+	4W	Убрать все эффекты	text-transform: capitalize
	Capitalize	4+	4W	Начинать с Прописных	
	Uppercase	4+	4W	ВСЕ ПРОПИСНЫЕ	
	Lowercase	4+	4W	все строчные	
text-align	Left	4+	4+	Выравнивание текста	text-align: justify выравнивание по ширине
	Right	4+	4+		
	Center	4+	4+		
	Justify	4+	4W		
text-indent	Точно	4+	4+	Отступ первой строки	text-indent: 15px; text-indent: 10%
	%	4+	4+		

NC - Netscape Communicator, IE - Microsoft Internet Explorer; 4+ означает, что свойство поддерживается в браузере 4 версии и выше. 4W - работает только под Windows.

Ниже, в Примере 3 приведены некоторые параметры текста и результат их применения.

Пример 3. Результат использования различных параметров текста

Пример: и это все о нем	Пример: текст по центру	Пример: Это не ссылка, а просто текст	Пример: отступ первой строки	Пример: полуторный межстрочный интервал
text-transform: capitalize	text-align: center	text-decoration: underline	text-indent: 20px	line-height: 1.5

Границы и рамки

Спецификация CSS описывает несколько свойств, с помощью которых можно создавать границу вокруг различных элементов и управлять ее видом. Границы - одна из наиболее слабых сторон CSS, т.к. браузеры содержат большое количество ошибок и по разному интерпретируют параметры. Старшие версии браузеров отображают рамки вокруг элементов более корректно.

Свойство	Значение	NC	IE	Op	Описание	Пример
padding-top padding-right padding-bottom padding-left padding	Значение %	4+	4+	3+	Отступ от границы элемента до его содержимого	table {padding: 15px 15px}
border-top-width border-right-	Thin Medium Thick	4+	4+	3+	Ширина границы	P {border-top-width: 4px}

width border-bottom-width border-left-width border-width	Значение					
border-color	цвет	4+	4+	3+	Цвет границы	P {border-color: red}
border-style	None Dotted Dashed Solid Double Groove Ridge Inset Outset	4+ 6 6 4+ 4+ 4+ 4+ 4+ 4+	4+ 5.5+ 5.5+ 4+ 4+ 4+ 4+ 4+	3+	Стиль рамки	table {border-style: double}
border-top border-right border-bottom border-left	border-top-width border-style цвет	6	4+	3+	Определяет толщину, стиль и цвет каждой границы	table {border-top: solid 4px red; border-left: solid 4px blue}
border	см. выше	4+	4+	3+	Задаёт толщину, стиль и цвет рамки	table {border: solid 4px red}

NC - Netscape Communicator, IE - Microsoft Internet Explorer, Op - Opera; 4+ означает, что свойство поддерживается в браузере 4 версии и выше.

Типы рамок

Для управления видом рамки предоставляется восемь значений параметра **border-style**. Результат их действия представлен на рисунке 1.

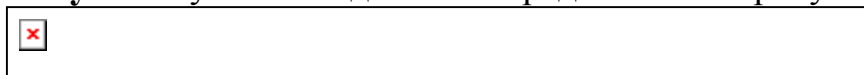
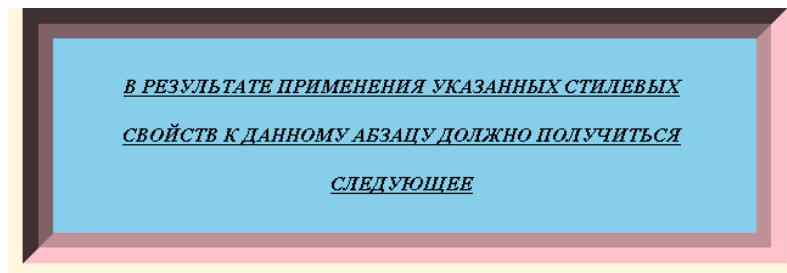


Рис.1. Стили рамок

Первые две - **dotted** и **dashed** стали поддерживаться браузерами Netscape и Internet Explorer только со старших версий.

```
<p style="color: yellow;
background-color: deepskyblue;
text-decoration: underline;
text-transform: uppercase;
border: pink inset 25;
padding: 20;
font-size: larger;
line-height: 40px;
text-align: center;"> ... </p>
```



Атрибут **style="..."** задает стилевое оформление абзаца

Атрибут **color: yellow;** задает цвет текста

Атрибут **background-color: deepskyblue;** задает цвет фона для абзаца

Атрибут **text-decoration: underline;** задает подчеркивание для текста

Атрибут **text-transform: uppercase;** задает режим заглавных букв для текста

Атрибут **border: pink inset 25;** задает рамку вокруг абзаца, соответственно, розовую выпуклую толщиной 25 пикселей

Атрибут **PADDING: 20;** задает

Атрибут **font-size: larger;** задает размер шрифта

АТРИБУТ **line-height: 40px;** задает межстрочный интервал

АТРИБУТ **text-align: center;** задает выравнивание текста внутри абзаца по центру

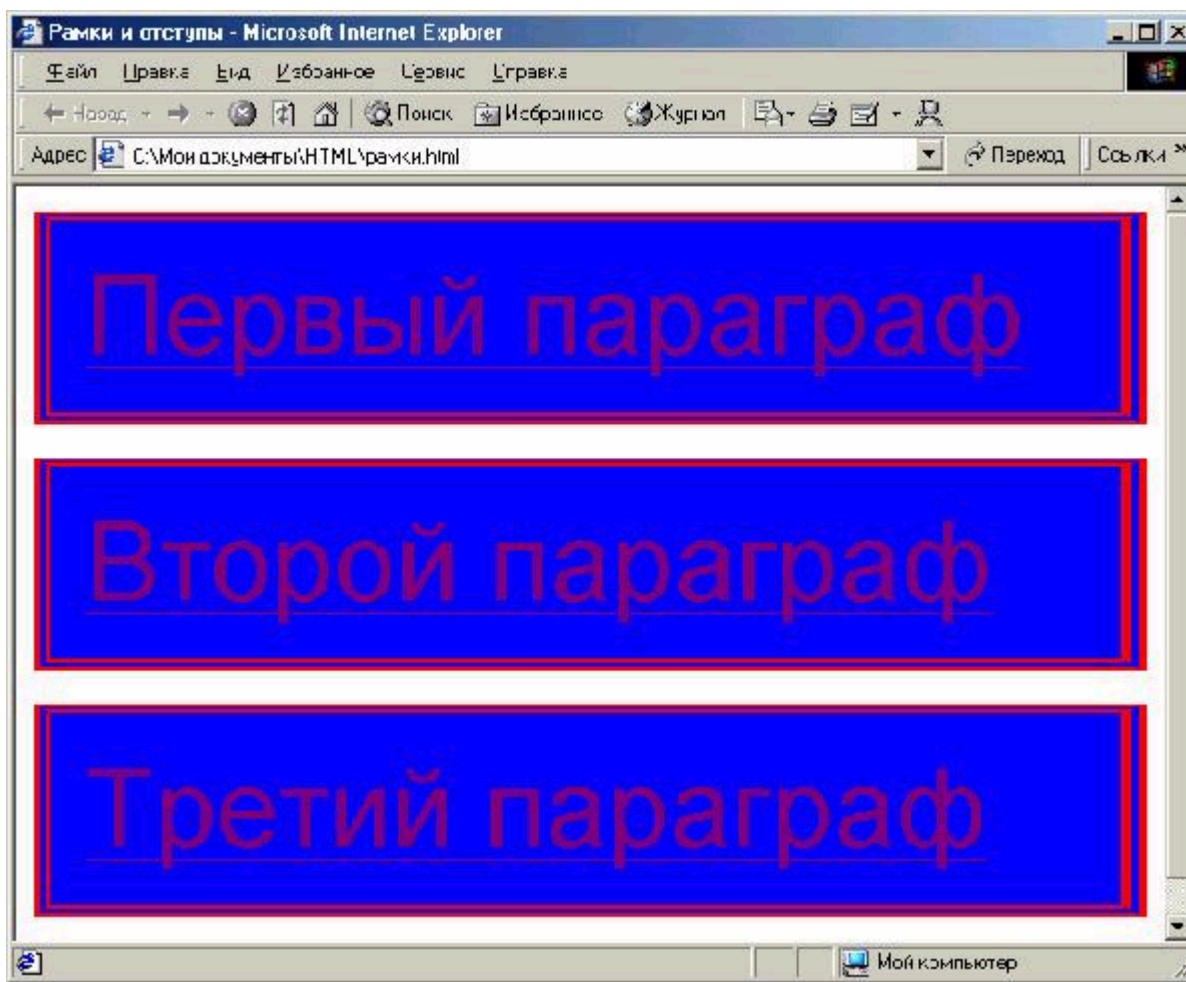
Задание 1

Создать в графическом редакторе Paint небольшой рисунок, который будет использоваться в качестве маркера списка. Создать маркированный список со своим маркером, используя стилевое свойство **LIST-STYLE-IMAGE: url(имя_рисунка)**.

Например, **<UL style="LIST-STYLE-IMAGE: url(star2.jpg)">**

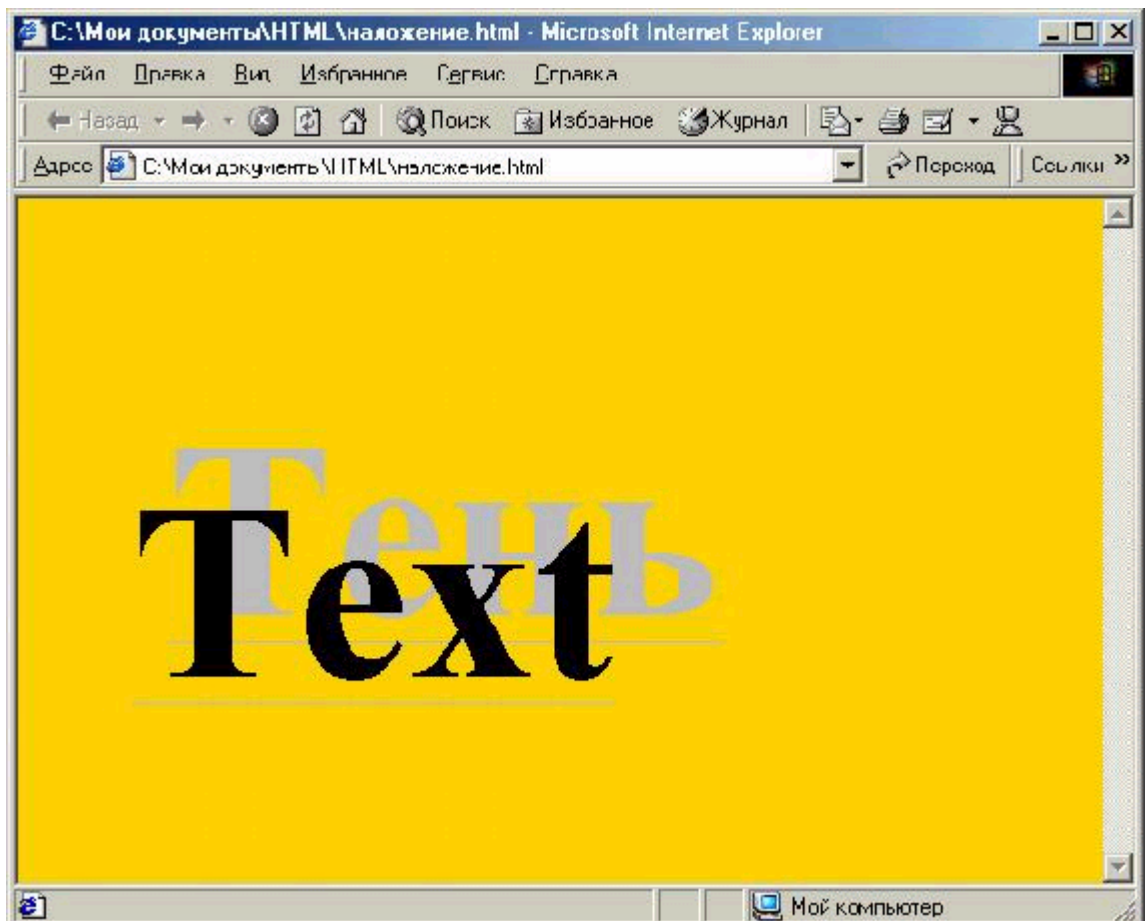
Задание 2

Используя свойства блоков текста, создать следующую страницу:



Задание 3

Значения отступов вокруг объектов можно указывать как положительные, так и отрицательные. Таким образом, можно использовать данную возможность для наложения одного слоя текста на другой. Например, можно создать текст с тенью, без использования трехмерного изображения. Создайте два стиля, которые отличаются цветом и размером отступов вокруг них. Используя отрицательные значения отступов и подбирая нужное значение, можно добиться того, что верхний слой как бы наползает на предыдущий. В результате у вас должно получиться следующее:



используйте контейнер `<DIV>` и свойства `margin-top` и `margin-left`.

Приложения
Приложение 1

Названия цветов

Название	Hex	Red	Green	Blue
aliceblue	#F0F8FF	240	248	255
antiquewhite	#FAEBD7	250	235	215
aqua	#00FFFF	00	255	255
aquamarine	#7FFFD4	127	255	212
azure	#F0FFFF	240	255	255
beige	#F5F5DC	245	245	220
bisque	#FFE4C4	255	228	196
black	#000000	00	00	00
blanchedalmond	#FFEBCD	255	235	205
blue	#0000FF	00	00	255
blueviolet	#8A2BE2	138	43	226
brown	#A52A2A	165	42	42
burlywood	#DE8887	222	136	135
cadetblue	#5F9EA0	95	158	160
chocolate	#D2691E	210	105	30
coral	#FF7F50	255	127	80
cornflowerblue	#6495ED	100	149	237
cornsilk	#FFF8DC	255	248	220
crimson	#DC143C	220	20	60
cyan	#00FFFF	00	255	255
darkblue	#00008B	00	00	139
darkcyan	#008B8B	00	139	139
darkgoldenrod	#B8860B	184	134	11
darkgray	#A9A9A9	169	169	169
darkgreen	#006400	00	100	00
darkkhaki	#BDB76D	189	183	109
darkmagenta	#8B008B	139	00	139
darkolivegreen	#556B2F	85	107	47
darkorange	#FF8C00	255	140	00
darkorchid	#9932CC	153	50	204
darkred	#8B0000	139	00	00
darksalmon	#E9967A	233	150	122
darkseagreen	#8FBC8F	143	188	143
darkslateblue	#483D8B	72	61	139

darkslategray	#2F4F4F	47	79	79
darkturquoise	#00CED1	00	206	209
darkviolet	#9400D3	148	00	211
deeppink	#FF1493	255	20	147
deepskyblue	#00BFFF	00	191	255
dimgray	#696969	105	105	105
dodgerblue	#1E90FF	30	144	255
firebrick	#B22222	178	34	34
floralwhite	#FFFAF0	255	250	240
forestgreen	#228B22	34	139	34
fuchsia	#FF00FF	255	00	255
gainsboro	#DCDCDC	220	220	220
ghostwhite	#F8F8FF	248	248	255
gold	#FFD700	255	215	00
goldenrod	#DAA520	218	165	32
gray	#808080	128	128	128
green	#008000	00	128	00
greenyellow	#ADFF2F	173	255	47
honeydew	#F0FFF0	240	255	240
hotpink	#FF69B4	255	105	180
indianred	#CD5C5C	205	92	92
indigo	#4B0082	75	00	130
ivory	#FFFFFF	255	255	240
khaki	#F0E68C	240	230	140
lavender	#E6E6FA	230	230	250
lavenderblush	FFF0F5	255	240	245
lemonchiffon	#FFFACD	255	250	205
lightblue	#ADD8E6	173	216	230
lightcoral	#F08080	240	128	128
lightcyan	#E0FFFF	224	255	255
lightgoldenrodyellow	#FAFAD2	250	250	210
lightgreen	#90EE90	144	238	144
lightpink	#FFB6C1	255	182	193
lightsalmon	#FFA07A	255	160	122
lightseagreen	#20B2AA	32	178	170
lightskyblue	#87CEFA	135	206	250
lightslategray	#778899	119	136	153
lightsteelblue	#B0C4DE	176	196	222

lightyellow	#FFFFE0	255	255	224
lime	#00FF00	00	255	00
limegreen	#32CD32	50	205	50
linen	#FAF0F6	250	240	246
magenta	#FF00FF	255	00	255
maroon	#800000	128	00	00
mediumaquamarine	#66CDAA	102	205	170
mediumblue	#0000CD	00	00	205
mediumorchid	#BA55D3	186	85	211
mediumpurple	#9370DB	147	112	219
mediumseagreen	#3CB371	60	179	113
mediumslateblue	#7B68EE	123	104	238
mediumspringgreen	#00FA9A	00	250	154
mediumturquoise	#48D1CC	72	209	204
mediumvioletred	#C71585	199	21	133
midnightblue	#191970	25	25	112
mintcream	#F5FFFA	245	255	250
mistyrose	#FFE4E1	255	228	225
moccasin	#FFE4B5	255	228	181
navajowhite	#FFDEAD	255	222	173
navy	#000080	00	00	128
oldlace	#FDF5E6	253	245	230
olive	#808000	128	128	00
olivedrab	#6B8E23	107	142	35
orange	#FFA500	255	165	00
orangered	#FF4500	255	69	00
orchid	#DA70D6	218	112	214
palegoldenrod	#EEE8AA	238	232	170
palegreen	#98FB98	152	251	152
paleturquoise	#AFEEEE	175	238	238
palevioletred	#DB7093	219	112	147
papayawhip	#FFefd5	255	239	213
peachpuff	#FFDAB9	255	218	185
peru	#CD853F	205	133	63
pink	#FFC0CB	255	192	203
plum	#DDA0DD	221	160	221
powderblue	#B0E0E6	176	224	230
purple	#800080	128	00	128

red	#FF0000	255	00	00
rosybrown	#BC8F8F	188	143	143
royalblue	#4169E1	65	105	225
salmon	#FA8072	250	128	114
sandybrown	#F4A460	244	164	96
seagreen	#2E2B57	43	46	87
seashell	#FFE5EE	255	229	238
sienna	#A0522D	160	82	45
silver	#C0C0C0	192	192	192
skyblue	#87CEEB	135	206	235
slateblue	#6A5ACD	106	90	205
slategray	#708090	112	128	144
snow	#FFFAFA	255	250	250
springgreen	#00FF7F	00	255	127
steelblue	#4682B4	70	130	180
tan	#D2B48C	210	180	140
teal	#008080	00	128	128
thistle	#D8BFD8	216	191	216
tomato	#FF6347	255	99	71
turquoise	#40E0D0	64	224	208
violet	#EE82EE	238	130	238
wheat	#F5DEB3	245	222	179
white	#FFFFFF	255	255	255
whitesmoke	#F5F5F5	245	245	245
yellow	#FFFF00	255	255	00
yellowgreen	#9ACD32	154	205	50

Краткий справочник по тегам

Краткий справочник содержит описание тегов HTML. **IE** - Microsoft Internet Explorer, **NC** - Netscape Communicator. Число означает, с какой версии браузер поддерживает данный тег.

Тег	Версия HTML	NC	IE	Описание
A	1.0	3+	3+	Ссылка
ADDRESS	2.0	3+	3+	Информация об авторе
APPLET	3.2	3+	3+	Java апплет
AREA	3.2	3+	3+	Задание активной области для карты изображения
B	2.0	3+	3+	Жирный текст
BASE	2.0	3+	3+	URL, относительно которого ведется адресация ссылок
BASEFONT	3.2	3+	3+	Задание шрифта, размера и цвета текста по умолчанию
BGSOUND	-	-	3+	Фоновая музыка
BIG	3.2	3+	3+	Увеличивает размер шрифта
BLINK	-	3+	-	Мерцающий текст
BLOCKQUOTE	2.0	3+	3+	Создание цитаты
BODY	2.0	3+	3+	Тело документа
BR	2.0	3+	3+	Перевод строки
BUTTON	4.0	6+	4+	Кнопка
CAPTION	3.2	3+	3+	Заголовок таблицы
CENTER	3.2	3+	4+	Выравнивание элемента по центру окна браузера
CITE	2.0	3+	3+	Используется для выделения цитат
CODE	2.0	3+	3+	Обозначает фрагмент программного кода
COL	4.0	-	3+	Колонка таблицы
COLGROUP	4.0	-	3+	Служит для определения группой столбца таблицы
COMMENT	-	-	3+	Комментарий игнорируемый браузером
DD	2.0	3+	3+	Описание списка определений
DEL	4.0	-	4+	Отмечает текст как удаленный
DFN	3.2	-	3+	Отмечает текстовый фрагмент как определение
DIR	2.0	3+	3+	Создание списка
DIV	3.2	3+	3+	Контейнер для задания стиля блока, использу-

				ется также для создания слоев
DL	2.0	3+	3+	Создание списка определений
DT	2.0	3+	3+	Задание списка определений
EM	2.0	3+	3+	Выделение фрагмента текста
EMBED	-	3+	3+	Вставка внешнего объекта в HTML
FIELDSET	4.0	6+	4+	Группировка элементов форм
FONT	3.2	3+	3+	Изменение параметров шрифта
FORM	2.0	3+	3+	Добавление формы
FRAME	4.0	3+	3+	Определяет одиночный фрейм
FRAMESET	4.0	3+	3+	Разбивает окно браузера на фреймы
H1	2.0	3+	3+	Заголовок первого уровня
H2	2.0	3+	3+	Заголовок второго уровня
H3	2.0	3+	3+	Заголовок третьего уровня
H4	2.0	3+	3+	Заголовок четвертого уровня
H5	2.0	3+	3+	Заголовок пятого уровня
H6	2.0	3+	3+	Заголовок шестого уровня
HEAD	2.0	3+	3+	Блок для хранения служебной информации
HR	2.0	3+	3+	Горизонтальная линия
HTML	2.0	3+	3+	Указывает браузеру, что документ в формате HTML
I	2.0	3+	3+	Курсивное начертание текста
IFRAME	4.0	-	3+	Плавающий фрейм
IMG	2.0	3+	3+	Вставка изображения
INPUT	2.0	3+	3+	Поле формы
INS	4.0	-	4+	Отмечает текст как вставку
ISINDEX	2.0	3+	4+	Поиск по ключевым словам
KBD	2.0	3+	3+	Отмечает текст как вводимый пользователем с клавиатуры
LABEL	4.0	6+	4+	Название группы в форме
LAYER	-	4+	-	Создание слоя
LEGEND	4.0	6+	4+	Создает рамку вокруг формы
LI	2.0	3+	3+	Элемент списка
LINK	2.0	4+	3+	Связь с другими документами
MAP	3.2	3+	3+	Карта-изображение
MARQUEE	-	-	3+	Поле для перемещения текста
MENU	2.0	3+	3+	Создание списка
META	2.0	3+	3+	Метаинформация
MULTICOL	-	3+	-	Многоколоночный текст
NOBR	-	3+	3+	Запрет перевода строки

NOEMBED	-	3+	-	Альтернативный текст для встроенных объектов
NOFRAMES	4.0	3+	3+	Альтернативное содержание для браузеров не поддерживающих фреймы
NOLAYER	-	3+	-	Альтернативное содержание для браузеров не поддерживающих слои
NOSCRIPT	4.0	3+	3+	Альтернативное содержание для браузеров не поддерживающих скрипты
OBJECT	4.0	4+	3+	Вставка внешнего объекта
OL	2.0	3+	3+	Нумерованный список
OPTION	2.0	3+	3+	Создание выпадающего списка
P	2.0	3+	3+	Параграф
PARAM	3.2	3+	3+	Задание значения параметру.
PLAINTEXT	-	3+	3+	Обычный текст
PRE	2.0	3+	3+	Текст пишется как есть, включая все пробелы
Q	4.0	-	4+	Отмечает короткие цитаты в строке текста.
S	3.2	2+	3+	Зачеркнутый текст
SAMP	2.0	-	3+	Отмечает текст как пример
SCRIPT	3.2	3+	3+	Область скрипта
SELECT	2.0	3+	3+	Элемент выпадающего списка
SMALL	3.2	3+	3+	Уменьшает размер шрифта
SPAN	4.0	4+	3+	Встроенный элемент для задания стиля фрагмента
STRIKE	3.2	3+	3+	Зачеркнутый текст
STRONG	2.0	3+	3+	Выделение фрагментов текста
STYLE	3.2	4+	3+	Указание стиля
SUB	3.2	3+	3+	Нижний индекс
SUP	3.2	3+	3+	Верхний индекс
TABLE	3.2	3+	3+	Таблица
TBODY	4.0	-	4+	Тело таблицы
TD	3.2	3+	3+	Ячейка таблицы
TEXTAREA	2.0	3+	3+	Поле многострочного текста в форме
TFOOT	4.0	-	3+	Нижний колонтитул таблицы
TH	3.2	3+	3+	Ячейка таблицы, которая является заголовком
THEAD	4.0	-	4+	Верхний колонтитул таблицы
TITLE	2.0	3+	3+	Заголовок страницы
TR	3.2	3+	3+	Строка таблицы
TT	2.0	3+	3+	Моноширинный текст
U	3.2	3+	3+	Подчеркнутый текст

UL	2.0	3+	3+	Маркированный список
VAR	2.0	3+	3+	Отмечает имена переменных программ

Литература

1. Матросов А.В., Сергеев А.О., Чаунин Н.П. HTML 4.0 – СПб.:БХВ – Петербург, 2002. – 672с.:ил.
2. Пауэлл Томас А. WEB – дизайн: Пер. с англ. СПб.:БХВ – Петербург, 2002. – 1024с.:ил.
3. Имери, Винс. Как сделать бизнес в Internet: Пер. с англ. – 3-е изд. – К.; М.; СПб.: Диалектика, 1998ю – 464с.:ил.
4. Успенский И.В. Интернет как инструмент маркетинга. - СПб.:БХВ – Петербург, 2000. – 256с.:ил.
5. А.Д. Александровский. Создание Web-страниц с использованием FrontPage98 и JavaScript. – М.: ДМК, 1998 – 368 с.:ил.
6. Крупник А. Поиск в Интернете: самоучитель (знакомство, работа, развлечение). – СПб.: Питер, 2002. – 272 с.:ил.
7. Хоффман П. Internet. К.: Диалектика, 1995. – 160 с.:ил.
8. Палмер С. VBScript и Active: библиотека программиста. – СПб.: ЗАО «Издательство «Питер», 1999, -368 с.:ил.
9. Монкир, Майкл. Освой самостоятельно JavaScript за 24 часа.: Пер. с англ.: Уч.пос. – М.: «Издательский дом «Вильямс», 2000, -304 с.:ил.
- 10.И. Мавджой. ASP. Web – профессионалам: Пер. с англ. – К.: Издательская группа BHV, 2001. – 288с.

Учебное издание

МИХАИЛ ФЕДОРОВИЧ ТУБОЛЬЦЕВ НАТАЛЬЯ ПАВЛОВНА ПУТИВЦЕВА
ИРИНА ВЛАДИМИРОВНА ГУРЬЯНОВА ОЛЬГА ВАЛЕРЬЕВНА НЕМЫКИНА

**Лабораторный практикум
по Web-технологиям**

Часть 1. Основы HTML технологий