

А. Е. Пупцев

**Технология построения Web-страниц
с помощью языка разметки гипертекста HTML**

Министерство образования Республики Беларусь
Академия последиplomного образования

А. Е. Пупцев

Технология построения Web-страниц
с помощью языка разметки гипертекста HTML

Учебное пособие

Минск 2001

Предисловие

В учебном пособии рассматриваются основные возможности языка разметки гипертекстовых документов HTML (HyperText Markup Language): форматирование текста, построение таблиц и списков, организация гиперссылок, фреймов и форм, таблицы стилей, которые используются при построении Web-страниц.

Учебное пособие содержит лабораторные работы, структурное содержание которых одинаковое. В начале каждой из них дается краткий теоретический материал, затем рассматриваются типовые задания по соответствующей теме и их решение на языке HTML с подробным объяснением, в конце предлагаются варианты заданий практического характера.

Издание предназначено для курсов повышения квалификации учителей информатики средних и средних специальных учебных заведений и курсов переподготовки учителей с получением диплома о переподготовке и присвоением квалификации «учителя информатики». Оно может быть использовано преподавателями педагогических университетов, ведущих подготовку студентов по специальности П 01.01.02 «Информатика» в учебном курсе «Современные проблемы информатики».

Автор учебного пособия выражает благодарность своим ученикам О. Трамбовичу и Е. Крупчик за ценные предложения и замечания при разработке учебного пособия.

1. История создания языка HTML

В 1992 году Тим Бернерс Ли (Tim Berners-Lee) создал основы для построения Всемирной информационной сети (World Wide Web или просто Web), задача которой заключалась в распространении информации посредством сетевой компьютерной технологии в Internet. Основываясь на работах Теда Нельсона (Ted Nelson) и концепции гипертекста Ли включил в Web графику, типографические текстовые стили и, самое главное, - гипертекстовые ссылки.

Web построена на трех основных технологиях:

- языке разметки гипертекстовых документов HTML (HyperText Markup Language), который служит для написания Web-страниц;
- протоколе передачи гипертекста HTTP для пересылки этих страниц;
- программах клиентах (Web-браузерах) для получения данных, их интерпретации и отображении результатов.

Благодаря HTML практически каждый, у кого есть текстовый редактор и подключение к Internet, может создать привлекательные Web-страницы и представить информацию так, как это не возможно сделать с помощью других электронных средств.

В дальнейшем будут подробно рассмотрены структура гипертекстового документа и основные возможности языка разметки гипертекстовых документов HTML. Предлагаемые для изучения конструкции языка HTML достаточно стандартны и связаны в основном с версиями HTML 3.x и HTML 4.x.

2. Основные терминологические понятия

Язык HTML – язык разметки гипертекста, который используется при подготовке документов во Всемирной компьютерной паутине WWW.

Гипертекст – это текст, содержащий гиперссылки.

Гиперссылка – фрагмент текста, который указывает на объекты различной природы: файлы, картинки, тексты, графику и др.

Тег (дескриптор) – стартовый или конечный элемент разметки, который имеет вид: < тег > - стартовый элемент разметки, < /тег > - конечный элемент разметки.

Атрибут – параметр или свойство элементов разметки, связанное с конкретным тегом.

Код HTML – набор конструкций языка HTML, т. е. совокупность тегов и атрибутов.

HTML файл (страница) – документ, построенный в виде гипертекста на основе языка HTML. Файлы HTML обычно имеют расширение .htm или .html.

Web-страница – документ (файл), построенный в виде гипертекста и размещенный в World Wide Web.

Сайт – набор Web-страниц, принадлежащих, как правило, одному владельцу или группе владельцев, объединенных общими целями и задачами.

Браузер – программа для просмотра Web-страниц.

URL (Uniform Resource Locator) – адрес некоторого объекта в Internet.

3. Структура гипертекстового документа

При формировании гипертекстового документа необходимо определить базисные элементы его структуры. К базисным элементам относятся: признаки начала и конца документа, область заголовка документа, имя документа, тело документа, комментарии, элементы заголовка и горизонтальные линии.

Каждый из базисных элементов описывается с помощью специальных тегов и, если необходимо, атрибутов к этим тегам.

`< HTML >` `< /html >` - теги, являющиеся признаками начала и конца документа. Каждый документ должен начинаться с тега `< HTML >` и заканчиваться тегом `< /html >`.

`< Head >` `< /head >` - тег области заголовка Web-страницы, который служит для формирования общей структуры документа.

`< Title > текст < /title >` - тег имени документа, текст, размещенный за тегом `< Title >` отображается в заголовке окна браузера.

`< ! текст комментариев >` - тег, позволяющий формировать комментарии, которые браузером игнорируются и не отображаются.

`< Body >` ... `< /body >` - тег, который включает в себе непосредственно гипертекст (тело документа), определяющий Web-страницу.

Тег `< Body >` может содержать дополнительные атрибуты для цветового оформления Web-страницы: *bgcolor*, *text*, *link*, *vlink*.

Для оформления цвет указывается по имени (английский эквивалент), например, *red* (красный), *green* (зеленый), *blue* (синий) или тремя двухразрядными 16-ми числами, которые определяют цвет и его интенсивность. Интенсивность красного, зеленого и синего цветов определяются парами шестнадцатеричных чисел соответственно.

Названия цветов и их коды представлены в таблице 1.

Таблица 1.

Русское название цвета	Английское название цвета	Код цвета
аквамарин	aqua	#00FFFF
белый	white	#FFFFFF
желтый	yellow	#FFFF00
зеленый	green	#008000
каштановый	maroon	#800000
красный	red	#FF0000
оливковый	olive	#808000
пурпурный	purple	#800080

светло-зеленый	lime	#00FF00
серебристый	silver	#C0C0C0
серый	gray	#808080
сизый	teal	#008080
синий	blue	#0000FF
ультрамарин	navy	#000080
фуксиновый	fuchsia	#FF00FF
черный	black	#000000

Атрибуты тега *< Body >* имеют следующее назначение:

bcolor = *цвет* – цвет фона Web-страницы;

text = *цвет* – цвет текста на Web-странице;

link = *цвет* – цвет текста гиперссылок;

vlink = *цвет* – цвет использованных гиперссылок.

При формировании Web-страниц часто требуется определить уровни заголовков, которых существует шесть. Каждый уровень обозначается Н1, Н2,...Н6. Для описания уровней используются соответствующие теги, например, для третьего уровня *< H3 >* и *< /H3 >*. Заголовок уровня 1 самый крупный, а уровня 6 самый маленький. Для тегов вида *< H1 >* обычно указывается одно из трех значений атрибута этого тега *align*:

align = *center* – выравнивание заголовка по центру;

align = *left* – выравнивание заголовка по левому краю;

align = *right* – выравнивание заголовка по правому краю.

Разбитие текста Web-страницы на части выполняют горизонтальные линии, которые описывает тег *< HR >*.

< HR > - тег для задания горизонтальной линии может использовать дополнительно атрибуты:

align = *выравнивание линии*;

size = *толщина линии в пикселах*;

width = *длина линии в пикселах или процентах*;

color = *цвет линии*.

К структуре HTML документа часто относят тег *< Meta >*, который содержит специальную служебную информацию, не отображающуюся при просмотре Web-страницы.

Тег *< Meta >* обычно содержит атрибуты со следующим назначением:

name = *“Expires” content* = *“Дата”* – « срок годности » документа;

name = *“Author” content* = *“Имя автора”* – имя автора Web-страницы;

http - equiv = *“Content – Type”* – используется для указаний инструкций пользователю;

content = *“text/html; charset = windows – 1251”* – указание на то, что в документе содержится текст в форме HTML и определена кодовая страница.

4. Форматирование текста

Текст Web-страницы – произвольные символы, определяемые по умолчанию как текстовые данные.

Форматирование текста Web-страницы выполняет большая группа тегов. Рассмотрим основные из них:

< *P* > < /*p* > - тег начала и конца абзаца. Данный тег может использовать атрибут *align*, назначение которого рассматривалось ранее в параграфе 3.

< *Pre* > < /*pre* > - тег вывода текста определенного пользователем заранее;

< *Center* > < /*center* > - тег для центрирования текста любого содержания;

< *Div* > < /*div* > - тег центрирования текста в зависимости от значения атрибута *align*;

< *Big* > < /*big* > - тег увеличенного размера шрифта;

< *Small* > < /*small* > - тег уменьшенного размера шрифта;

< *TT* > < /*tt* > - тег для обозначения текста телетайпа (моноширинного шрифта);

< *Strike* > < /*strike* > - тег для создания перечеркнутого текста;

< *U* > < /*u* > - тег для формирования подчеркнутого текста;

< *I* > < /*i* > - тег вывода текста курсивом;

< *SUB* > < /*sub* > - тег для написания нижних индексов;

< *SUP* > < /*sup* > - тег для написания верхних индексов;

< *Font* > < /*font* > - тег для определения типа, размера и цвета шрифта. Все эти характеристики тега < *Font* > определяются с помощью специальных атрибутов:

size = абсолютный размер шрифта. Абсолютный размер принимает значения от 1 до 7.

Размер шрифта может задаваться не в абсолютном, а в относительном виде по отношению к базовому размеру, например, *size* = + число или *size* = - число. Так для базового размера равного двум, относительный размер может находиться в пределах от –1 до +5.

color = цвет – атрибут цвета шрифта.

Изменение направления текста на обратный выполняет тег < *BDO* > < /*bdo* > с атрибутом *DIR*. При этом шаблон записи выглядит так:

< *BDO dir* = “*RTL*” > Текст, направление которого меняется < /*bdo* >

При формировании текста на странице он может быть представлен в виде бегущей строки с помощью тега < *Marquee* > < /*marquee* >. Для определения движения текста слева на право или справа на лево используется атрибут *direction*. Например, для организации движения текста слева на право шаблон записи будет выглядеть следующим образом:

< *Marquee direction* = *left? right* > текст < /*marquee* >

Типовое задание. Отформатируйте текст, как представлено ниже:

Форматирование текста

ВНЕШНИЙ ВИД

полужирный шрифт

курсивный шрифт

подчеркнутое начертание

~~перечеркнутое начертание~~

Печатная машинка

Обычный текст и элемент SUB

Обычный текст и элемент SUP

ЗАДАНИЕ ЦВЕТА, РАЗМЕРА И ТИПА ШРИФТА

Задан зеленый цвет шрифта размера 4 типа Courier

ЗАДАНИЕ АБСОЛЮТНЫХ РАЗМЕРОВ ШРИФТОВ

Шрифт размера 7

Шрифт размера 6

Шрифт размера 5

Шрифт размера 4

Шрифт размера 3

Шрифт размера 2

Шрифт размера 1

ЗАДАНИЕ ОТНОСИТЕЛЬНЫХ РАЗМЕРОВ ШРИФТОВ

Шрифт размера +4

Шрифт размера +3

Шрифт размера +2

Шрифт размера +1

Шрифт размера +0

Шрифт размера -1

Шрифт размера -2

Большой шрифт (используется элемент BIG)

Маленький шрифт (используется элемент SMALL)

ОПРЕДЕЛЕНИЕ СТРУКТУРЫ ТЕКСТА

Заголовок 1
Заголовок 2

Заголовок 3

Заголовок 4

Заголовок 5

Заголовок 6

Использование элемента CENTER

Выравнивание текста (DIV)

Элемент BR обеспечивает разрыв
строки

Эта строка, хоть и длинная, не должна разрываться
броузером, потому что текст этой строки находится внутри
элемента NOBR

Использование элемента PRE.

Он запрещает форматирование
текста браузером и выводит его так,
как определил пользователь.

Направление текста можно изменить

Форматирование текста

Рассмотрите и проанализируйте решение практического задания:

```
<HTML>
<HEAD>
<TITLE> Форматирование текста </title></head>
<META http-equiv="Content-Type" content="text/html; charset=windows-
1251">
<BODY>

<!-- цвет фона серебристый, цвет текста синий -->
<body bgcolor=silver text=blue>

<CENTER>
<FONT size=6 color="navy"> <B>Ф</b></font><FONT
size=5><I>ормат<FONT color="lime"><SUB>ирование </sub></i>
</font><FONT size=2 color="purple">m</font><FONT size=3
color="blue">e</font><FONT size=4 color="green">к</font><FONT size=5
color="maroon">c</font><FONT size=6 color="fuchsia">m</font><FONT
size=7 color="orange">a</font>
</center>

<!-- горизонтальная линия -->
<HR align=center size=6 color=maroon>

<CENTER>
<FONT color=red><H3>ВНЕШНИЙ ВИД</h3>
</font>
</center>
<HR>
<P><b>полужирный шрифт </b>
<P><i>курсивный шрифт</i>
<P><u>подчеркнутое начертание</u>
<P><del>перечеркнутое начертание</del>
<P><TT>Печатная машинка</TT>
```

<P> Обычный текст и _{элемент SUB}

<P> Обычный текст и ^{элемент SUP}

<HR size=4 color=fuchsia>

<CENTER>

<H3>ЗАДАНИЕ ЦВЕТА, РАЗМЕРА И ТИПА
ШРИФТА</h3>

</center>

<HR>

<P> Задан зеленый цвет
шрифта размера 4 типа Courier

<HR size=4 color=fuchsia>

<CENTER>

<H3>ЗАДАНИЕ АБСОЛЮТНЫХ РАЗМЕРОВ
ШРИФТОВ</h3>

</center>

<HR>

<P> Шрифт размера 7

<P> Шрифт размера 6

<P> Шрифт размера 5

<P> Шрифт размера 4

<P> Шрифт размера 3

<P> Шрифт размера 2

<P> Шрифт размера 1

<HR size=4 color=fuchsia>

<CENTER>

<H3>ЗАДАНИЕ ОТНОСИТЕЛЬНЫХ РАЗМЕРОВ
ШРИФТОВ</h3>

</center>

<HR>

<P> Шрифт размера +4

<P> Шрифт размера +3

<P> Шрифт размера +2

<P> Шрифт размера +1

<P> Шрифт размера +0

<P> Шрифт размера -1

<P> Шрифт размера -2

<HR size=4 color=fuchsia>

<P> <BIG> Большой шрифт (используется элемент BIG) </big>

<P> <SMALL> Маленький шрифт (используется элемент SMALL) </small>
 <HR size=4 color=fuchsia>

<CENTER>
 <H3>ОПРЕДЕЛЕНИЕ СТРУКТУРЫ ТЕКСТА</h3>

 </center>
 <HR>
 <H1>Заголовок 1</h1>
 <H2>Заголовок 2</h2>
 <H3>Заголовок 3</h3>
 <H4>Заголовок 4</h4>
 <H5>Заголовок 5</h5>
 <H6>Заголовок 6</h6>
 <HR>

<P> <CENTER> Использование элемента CENTER </center>
 <HR>

<P> <DIV align="right">Выравнивание текста (DIV) </div>

<HR>
 <P> Элемент BR обеспечивает разрыв
 строки

<HR>
 <P><NOBR> Эта строка, хоть и длинная, не должна разрываться
 браузером, потому что текст этой строки находится внутри элемента
 NOBR </nobr>

<HR>
 <PRE> Использование элемента PRE.
 Он запрещает форматирование
 текста браузером и выводит его так,
 как определил пользователь.</pre>

<HR>
 <P>Направление текста можно изменить
 <P>
 <BDO dir="RTL">Направление текста можно изменить</BDO>
 <HR size=4 color=fuchsia>

</BODY></HTML>

5. Гиперссылки и добавление изображения.

Гиперссылки

При рассмотрении понятия ссылка следует выделить две ее составляющие части: место размещения ссылки в документе (элемент привязки или якорь) и направление перехода по данной ссылке (адрес ссылки).

В качестве элементов привязки могут выступать: слова, группы слов (текстовые элементы) или изображения (графические элементы).

текстовые элементы привязки обычно выделяются цветом и подчеркиваются, графические элементы (изображения) не подчеркиваются и не выделяются цветом, но часто обводятся рамкой. К графическим элементам привязки относятся маркеры, пиктограммы, реклама и др.

Ссылки на URL (адрес Web-страницы) подразделяются на относительные и абсолютные.

Относительной ссылкой называют ссылку на URL файла, находящегося на том же компьютере.

Абсолютная ссылка – это ссылка, в которой конкретно указаны компьютер, каталог и файл.

За организацию ссылок в языке HTML отвечает тег $\langle A \rangle \langle /a \rangle$, который чаще всего использует следующий шаблон:

Произвольный текст $\langle A \ href = \text{“Адрес ссылки”} \rangle$ текст для щелчка по нему мышью $\langle /a \rangle$

Атрибут *href* тега $\langle A \rangle$ может указывать на метку внутри страницы (внутренние ссылки), файл на локальном диске или ресурс Internet.

Организация внутренних ссылок, позволяющая осуществлять переход внутри текущей страницы, выполняется в два этапа:

- сначала в нужных местах Web-страницы устанавливается метка

$\langle A \ name = \text{“Метка”} \rangle \langle /a \rangle$;

- на втором этапе определяется ссылка на метку по следующему шаблону:

Текст подсказки $\langle A \ href = \text{“\#Метка”} \rangle$ Текст для щелчка $\langle /a \rangle$

Если в качестве адреса ссылки должно быть размещено указание на файл на локальном диске, то атрибут *href* принимает значение вида:

href = $\text{“file: // Диск: \ Путь к файлу”}$.

Общий шаблон атрибута *href*: *href* = “URL” или *href* = $\text{“Протокол: // Адрес ссылки”}$.

Для обеспечения доступа к различным протоколам используются схемы доступа, представленные в таблице 2.

Обозначение ресурса	Название ресурса
<i>ftp</i>	Доступ к архивам файлов при помощи протокола FTP (file transfer protocol)
<i>gothor</i>	Доступ к системе Gopher
<i>http</i>	Доступ к WWW
<i>mailto</i>	Отправка сообщения по электронной

	почте
<i>news</i>	Доступ к новостям USENET
<i>telnet</i>	Подключение к протоколу telnet
<i>wais</i>	Подключение к системе поиска WAIS

Приведем несколько примеров записи тега `< A > < /a >`.

Пример 1. Организация гиперссылки на Web-страницу БГПУ им. М. Танка
`< A href = "http: // www.bspu.unibel.by" > БГПУ < /a >`

Пример 2. Формирование ссылки для отправки сообщения по электронной почте:

`< A href = "mailto: me@mycomp.com" > Пишите < a/ >`

Пример 3. Ссылка на раздел новостей USENET

`< A href = "news: news.newusers.questions" > Новости < /a >`

Добавление изображений

Для хранения изображений используются десятки различных форматов: GIF, JPEG, psx, png, wmf, bmp, и др. Однако для работы с изображениями в документах HTML обычно выбирают форматы, которые распознаются браузерами: GIF, JPEG. Для вставки изображения используется тег `< IMG >` с атрибутом *SRC*, который указывает URL графического файла
`< IMG SRC = "URL" >`

Тег `< IMG >` использует атрибуты, значения которых имеют большую важность при размещении изображения на Web-странице. Основные атрибуты тега `< IMG >` представлены в таблиц 3:

Название атрибута	Назначение / значение атрибута
<i>ALIGN</i>	Выравнивание текста по: <i>TOP</i> – верхнему краю изображения; <i>MIDDLE</i> – средней части изображения; <i>BOTTOM</i> – нижнему краю изображения; <i>LEFT</i> – показывает изображение слева; <i>RIGHT</i> – показывает изображение в правой части окна
<i>HEIGHT</i>	Указывает высоту изображения в пикселах или процентах

<i>WIDTH</i>	Указывает высоту изображения в пикселах или процентах
<i>Border</i>	Присваивает значение ширины рамки в пикселах или процентах
<i>VSPACE</i>	Расширяет свободное пространство в пикселах вокруг изображения по вертикали
<i>HSPACE</i>	Расширяет свободное пространство в пикселах вокруг изображения по горизонтали

Пример 4. Выведите на Web-странице стандартную вставку для Windowsиз файла Облака.bmp по середине страницы высотой 250 пикселей, шириной 200 пикселей, шириной рамки 2 пиксела и отступами вокруг изображения по горизонтали – 100, по вертикали – 50

```
< IMG src = "file: // c:\Windows\Облака.bmp"
align = "middle" height = 250 width = 200 border = 2 hspace = 100
vspace = 50
```

При организации ссылки, в качестве которой используется изображение применяется шаблон из комбинации двух тегов < A > < /a > и < IMG >:

```
< A href = "Адрес ссылки" > < IMG src = "ссылка на рисунок" > < a />
```

Типовое задание: Постройте на Web-странице гиперссылки следующего вида:

ГИПЕРССЫЛКИ

ВНУТРЕННИЕ ССЫЛКИ

Переход в [конец](#) документа

Переход к [ссылке 1](#)

Здесь расположена **ссылка 1**

ССЫЛКИ НА АДРЕС ЭЛЕКТРОННОЙ ПОЧТЫ

Текстовый элемент привязки в ссылке

Свои предложения высылайте [автору](#) этого пособия.

Графический элемент привязки в ссылке

Нажмите на картинку, чтобы отправить сообщение автору.

ССЫЛКИ НА РЕСУРСЫ INTERNET

[БГПУ](#) страница

Переход в [начало](#) документа.

Проанализируйте код HTML для организации ссылок и определите виды этих ссылок.

```
<HTML>
<HEAD>
<TITLE>Гиперссылки</title>
<META http-equiv="Content-Type" content="text/html; charset=windows-
1251">
<BODY bgcolor=yellow>

<A name="top"></a>

<CENTER>
<FONT size=6 color=lime><I> ГИПЕРССЫЛКИ </i></font>
</center>
<! горизонтальная линия>
<HR align=center size=6 color=blue>

<CENTER>
<FONT color=red><H3>ВНУТРЕННИЕ ССЫЛКИ </h3>
</font>
</center>
<HR>
Переход в <A href="#bottom">конец</a> документа<P>
<HR>
Переход к <A href="#S001"><B>ссылке 1</b></a><P>
<HR>
Здесь расположена <B>ссылка 1</b><A name="S001"></a>
<HR align=center size=4 color=blue>
```

```

<CENTER>
<FONT color=red><H3>ССЫЛКИ НА АДРЕС ЭЛЕКТРОННОЙ
ПОЧТЫ</h3>
</font>
</center>
<HR>
<CENTER>
<FONT color=green><H4>Текстовый элемент привязки в ссылке</h4>
</font>
</center>
Свои предложения высылайте
<A href="mailto:Pup@ape.edu.by">автору</a>
этого пособия.
<HR>
<CENTER>
<FONT color=green><H4>Графический элемент привязки в ссылке</h4>
</font>
</center>
<P>Нажмите на картинку, чтобы отправить сообщение автору.
<P>
<A HREF="mailto:Pup@ape.edu.by">
<IMG SRC="postbox.gif"
WIDTH="65" HEIGHT="135"></a>
<HR align=center size=4 color=blue>

<CENTER>
<FONT color=red><H3>ССЫЛКИ НА РЕСУРСЫ INTERNET</h3>
</font>
</center>
<A HREF="http://www.bspu.unibel.by">БГПУ</a> страница
<HR align=center size=4 color=blue>

<A name="bottom"></a><P>
Переход в <A href="#top">начало</a> документа.
<HR align=center size=4 color=blue>

</body>
</html>

```

6. Построение списков

Все теги для создания списков могут быть условно разделены на две группы: одни определяют общий вид списка, а другие задают внутреннюю структуру списка.

Существуют в основном четыре вида списков, которые могут быть размещены на Web-странице: нумерованный (маркированный), нумерованный, вложенный (иерархический), и список с определениями.

Ненумерованный список (unordered list) строится по следующему шаблону:

шаблон:	Вид списка:
< UL >	
< Li > Первая строка списка	Первая строка списка
< Li > Вторая строка списка	Вторая строка списка
...	...
< Li > n-ая строка списка	n-ая строка списка
< /UL >	

Как видно из представленного шаблона для построения списка необходимо использовать два вида тегов: < UL > < /ul > - для обрамления списка и < Li > - для обозначения каждого пункта списка.

При построении нумерованного списка вместо тега < UL > < /ul > используется тег < Ol > < /ol > , к которому добавляется атрибут *Type* для определения способа нумерации. некоторые возможные варианты нумерации представлены в таблице 4.

Таблица 4

Значение атрибута type	Способ нумерации
<i>type</i> = 'a'	a, b, c, d ...
<i>type</i> = '4'	4, 5, 6, 7, 8 ...
<i>type</i> = 'I'	I, II, III, IV ...
<i>type</i> = 'B'	B, C, D ...

Атрибут *Type* является частью тега < OL > и записывается, например, < OL type = 'I' >.

Список с определениями (definition list) строится с помощью тегов трех наименований:

< DL > < /dl > - тег для определения начала и конца списка с определениями;
 < DT > - тег для описания пункта списка с определениями, в котором могут быть вложенности;

< DD > - тег для описания вложенных пунктов. шаблон списка с определениями в кодах HTML и его внешний вид на странице представлен ниже

Шаблон	Вид списка
< DL >	
< DT > Первая строка	Первая строка
< DD > Определение 1	Определение 1

< DD > Определение 2
< DT > Вторая строка
< DD > Определение 3
< /dt >

Определение 2
Вторая строка
Определение 3

При формировании вложенных списков используются различные комбинации трех видов тегов < UL > < /ul >, < OL > < /ol > и < Li >.

Ненумерованный (маркированный) список может иметь своеобразный стиль маркеров, которые могут задаваться двумя способами с помощью атрибута *class* в теге < OL > с указанием кода маркера, например, < OL class = "COI" > или атрибута *type* в теге < UL >, например, < UL type = circle > (см. типовое задание).

Типовое задание: Сформируйте Web-страницу со списками следующего вида:

Списки

МАРКИРОВАННЫЙ СПИСОК

- Пункт 1 списка
- Пункт 2 списка
- Пункт 3 списка

НУМЕРОВАННЫЙ СПИСОК

(использован атрибут value =4, нумерация осуществляется римскими числами)

- IV. Пункт 1
- V. Пункт 2
- VI. Пункт 3
- VII. Пункт 4

ВЛОЖЕННЫЙ СПИСОК

- Пункт 1
 - 1. Пункт 1.1
 - 2. Пункт 1.2
 - Пункт 2
 - 1. Пункт 2.1
 - 2. Пункт 2.2
 - Пункт 3
 - 1. Пункт 3.1
 - 2. Пункт 3.2
-

СПИСОК С ОПРЕДЕЛЕНИЯМИ

- Пункт 1
 Определение пункта 1
 Другое определение пункта 1
- Пункт 2
 Определение пункта 2
- Пункт 3
 Определение пункта 3
-

ДВА СПОСОБА ПЕРЕОПРЕДЕЛЕНИЯ СТИЛЯ В МАРКИРОВАННОМ СПИСКЕ

- Пункт 1
- Пункт 2

-
- Пункт 1 списка
 - Пункт 2 списка
 - Пункт 3 списка

Разберите и проанализируйте алгоритм в кодах HTML:

```
<HTML>
<HEAD>
<TITLE>Построение списков </title>
<META http-equiv="Content-Type" content="text/html; charset=windows-
1251">
<STYLE type="text/css">
OL.c01 {list-style-type: square}
</style>
</head>
<BODY>

<! цвет фона светло-зеленый>
<body bgcolor=lime>

<CENTER>
<FONT size=7 color="purple">c</font><FONT size=6
color="blue">n</font><FONT size=5 color="green">u</font><FONT size=4
color="maroon">c</font><FONT size=3 color="fuchsia">к</font><FONT
size=2 color="orange">u</font>
</center>

<! горизонтальная линия>
<HR align=center size=6 color=orange>

<CENTER>
```

```
<FONT color=red><H3>МАРКИРОВАННЫЙ СПИСОК</h3>
</font>
</center>
<UL>
```

```
<LI>Пункт 1 списка
```

```
<LI>Пункт 2 списка
```

```
<LI>Пункт 3 списка
```

```
</ul>
```

```
<HR>
```

```
<CENTER>
```

```
<FONT color=red><H3>НУМЕРОВАННЫЙ СПИСОК <BR>
```

```
(использован атрибут value =4, нумерация осуществляется римскими
числами)</h3>
```

```
</font>
```

```
</center>
```

```
<OL type="I">
```

```
<LI value = 4>Пункт 1
```

```
<LI>Пункт 2
```

```
<LI>Пункт 3
```

```
<LI>Пункт 4
```

```
</ol>
```

```
<HR>
```

```
<CENTER>
```

```
<FONT color=red><H3>ВЛОЖЕННЫЙ СПИСОК</h3>
```

```
</font>
```

```
</center>
```

```
<UL>
```

```
<LI>Пункт 1
```

```
<OL>
```

```
<LI>Пункт 1.1
```

```
<LI>Пункт 1.2
```

```
</ol>
```

```
<LI>Пункт 2
```

```
<OL>
```

```
<LI>Пункт 2.1
```

```
<LI>Пункт 2.2
```

```
</ol>
```

```
<LI>Пункт 3
```

```
<OL>
```

```
<LI>Пункт 3.1
```

```
<LI>Пункт 3.2
```

```
</ol>
```

```
</ul>
```

<HR>

<CENTER>

<H3>СПИСОК С ОПРЕДЕЛЕНИЯМИ</h3>

</center>

<DL>

<DT>Пункт 1

<DD>Определение пункта 1

<DD>Другое определение пункта 1

<DT>Пункт 2

<DD>Определение пункта 2

<DT>Пункт 3

<DD>Определение пункта 3

</dl>

<HR>

<CENTER>

<H3>ДВА СПОСОБА ПЕРЕОПРЕДЕЛЕНИЯ СТИЛЯ В
МАРКИРОВАННОМ СПИСКЕ</h3>

</center>

<OL class="c01">

 Пункт 1

 Пункт 2

<HR>

<UL type=circle>

Пункт 1 списка

Пункт 2 списка

Пункт 3 списка

<HR>

</BODY></HTML>

Лабораторная работа №3

Формирование списков

Сформируйте списки следующих видов:

Вариант 1.

Пункт 1

А. пункт 1.1

- ♦ пункт 1.1.1
- ♦ пункт 1.1.2

- ♦ пункт 1.1.3
- В. пункт 1.2
 - пункт 1.2.1
 - пункт 1.2.2

Пункт 2

- 2. пункт 1
 - 1. пункт 2.1.1
 - 2. пункт 2.1.2
 - 3. пункт 2.1.3
- 3. пункт 2
 - пункт 2.2.1
 - пункт 2.2.2
 - пункт 2.2.3

Пункт 3

Определение пункта 3

Другое определение пункта 3

Вариант 2.

□ Пункт 1

- пункт 1.1
 - 3 пункт 1.1.1
 - 4 пункт 1.1.2
 - 5 пункт 1.1.3
- пункт 1.2
 - пункт 1.2.1
 - определение пункта 1.2.1
 - другое определение пункта 1.2.1
 - пункт 1.2.2
 - определение пункта 1.2.2
 - другое определение пункта 1.2.2
 - пункт 1.2.3
 - определение пункта 1.2.3

□ Пункт 2

- С пункт 2.1
- D пункт 2.2

□ Пункт 3

- пункт 3.1
- пункт 3.2

Вариант 3

III Пункт 1

- пункт 1.1
- пункт 1.2

- пункт 1.3
- IV Пункт 2
 - 1 пункт 2.1
 - пункт 2.1.1
 - определение пункта 2.1.1
 - пункт 2.1.2
 - определение пункта 2.1.2
 - другое определение пункта 2.1.2
 - пункт 2.1.3
 - определение пункта 2.1.3
 - 2 пункт 2.2
 - пункт 2.2.1
 - В пункт 2.2.1.1
 - С пункт 2.2.1.2
 - пункт 2.2.2
- V Пункт 3
 - пункт 3.1
 - пункт 3.2

Вариант 4

- 5 Пункт 1
 - пункт 1.1
 - определение пункта 1.1
 - пункт 1.2
 - определение пункта 1.2
 - пункт 1.3
 - определение пункта 1.3
 - другое определение пункта 1.3
- 6 Пункт 2
 - пункт 2.1
 - 1 пункт 2.1.1
 - пункт 2.1.1.1
 - пункт 2.1.1.2
 - пункт 2.1.1.3
 - 2 пункт 2.
 - III пункт 2.1.2.1
 - пункт 2.2
 - пункт 2.2.1
 - пункт 2.2.2
- 7 Пункт 3

Вариант 5

- Пункт 1
 - 3 пункт 1.1
 - пункт 1.1.1
 - определение пункта 1.1.1
 - другое определение пункта 1.1.1
 - пункт 1.1.2
 - определение пункта 1.1.2
 - пункт 1.1.3
 - А пункт 1.1.3.1
 - В пункт 1.1.3.2
 - 4 пункт 1.2
 - пункт 1.2.1
 - пункт 1.2.2
 - пункт 1.2.2.1
 - пункт 1.2.2.2
- Пункт 2
 - А пункт 2.1
 - В пункт 2.2
 - пункт 2.2.1
 - пункт 2.2.2
 - пункт 2.2.3
- Пункт 3

Вариант 6

- Пункт 1
 - пункт 1.1
 - А пункт 1.1.1
 - пункт 1.1.1.1
 - пункт 1.1.1.2
 - пункт 1.1.1.3
 - В пункт 1.1.2
 - пункт 1.2
 - пункт 1.2.1
 - определение пункта 1.2.1
 - другое определение пункта 1.2.1
 - пункт 1.2.2
 - определение пункта 1.2.2
 - пункт 1.2.3
- Пункт 2
 - III пункт 2.1
 - пункт 2.1.1
 - пункт 2.1.2

- IV пункт 2.2
- A пункт 2.2.1
- B пункт 2.2.2
- C пункт 2.2.3

7. Таблицы

Формирование таблицы на Web-странице сопровождается прорисовкой браузером ее рамок. Размер рамок автоматически согласовывается с размером окна просмотра браузера, размером текстов и изображений в ячейках таблицы.

При построении таблицы используются теги трех групп:

- основной элемент описания таблицы – тег `< Table > < /table >`;
- определение строк в таблице – тег `< TR >`;
- описание ячеек в таблице – теги `< TD >` и `< TH >`.

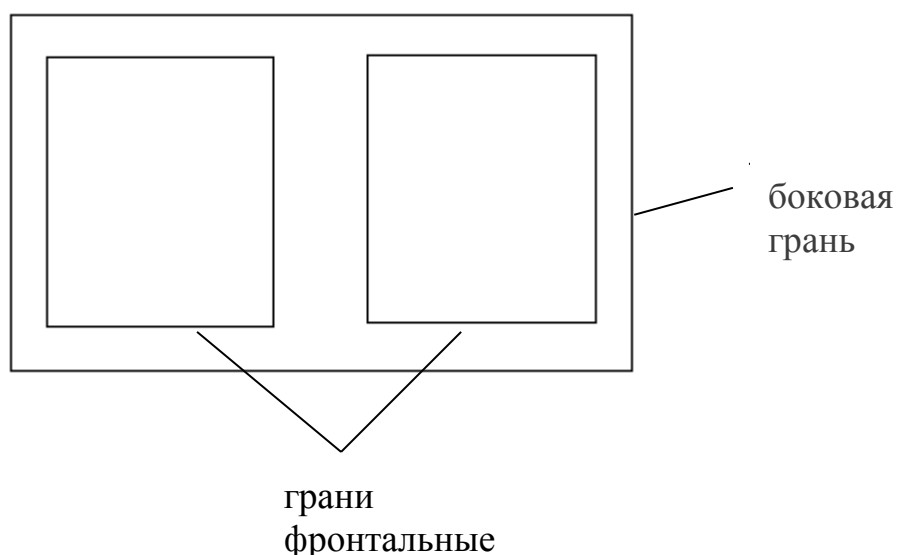
Тег `< Table > < /table >` использует целый ряд дополнительных атрибутов: *align*, *width*, *border*, *cellspacing*, *cellpadding*. В дальнейшем рассмотрим назначение каждого атрибута в отдельности.

Для таблицы на странице применяется атрибут *align*:

- *align* = “left” – выравнивание таблицы по левому краю;
- *align* = “center” – выравнивание таблицы по центру;
- *align* = “right” – выравнивание таблицы по правому краю.

Ширина таблицы может быть задана в пикселах или в процентном отношении к ширине страницы, например, *width* = 450 или *width* = 40%.

При построении таблицы создается “впечатление” их выпуклости с помощью двух граней фронтальной и боковой:



Ширину боковой грани в пикселах определяет атрибут *border* = ширина боковой грани, а ширину фронтальных граней определяет атрибут *cellspacing* = ширина фронтальных граней.

За расстояние размера пустого пространства вокруг текста в ячейках отвечает атрибут *cellpadding* = число пикселей.

Сразу за тегом `< Table >` может быть размещен специальный тег для определения заголовка страницы `< Caption > < /caption >`. С помощью атрибута `align` в этом теге заголовок может располагаться над таблицей `align = "top"` или под таблицей `align = "bottom"`.

Каждую строку в таблице описывает тег `< TR >`. Для выравнивания всех ячеек в строке может использоваться атрибут `align` со своими стандартными значениями `left`, `center`, `right`.

Для описания в ячейках заголовков используется внутри тега `< TR >` тег `< TH >`. При этом текст заголовка в ячейке обычно выделяется полужирным шрифтом.

Цвет шрифта текста в таблице, как правило задается с помощью тега `< Font >` с атрибутом `color`, например, `< Font color = red >`, а цвет фона внутри ячейки задается атрибутом `bgcolor` внутри тега `< TH >`, например, `< TH bgcolor = "lime" >`.

Тег `< TD >` определяет обычную ячейку в таблице.

В дальнейшем рассмотрим несколько примеров построения таблиц.

Пример 1. Постройте таблицу следующего вида:

СТАНДАРТНАЯ ТАБЛИЦА

Заголовок таблицы	
Заголовок 1	Заголовок 2
Ячейка 1	Ячейка 2

Запись на языке HTML (фрагмент) представлен ниже:

```
<CENTER>
<FONT color=red><H3>СТАНДАРТНАЯ ТАБЛИЦА</h3>
</font>
</center>
<TABLE border=4 cellspacing=3>
<CAPTION> Заголовок таблицы </caption>
<TR><TH bgcolor="lime">Заголовок 1
    <TH bgcolor="yellow">Заголовок 2
<TR><TD>Ячейка 1
    <TD>Ячейка 2
</table>
```

Пример 2. Сформулируйте таблицу из шести ячеек с тремя заголовками и одной пустой ячейкой в правом верхнем углу:

СПЕЦИАЛЬНАЯ ТАБЛИЦА

Заголовок таблицы		
	Заголовок 1	Заголовок 2
Заголовок 3	Ячейка 1	Ячейка 2

Фрагмент построения таблицы на языке HTML:

```
<CENTER>
<FONT color=red><H3>СПЕЦИАЛЬНАЯ ТАБЛИЦА</h3>
</font>
</center>
<TABLE border=4 cellpadding=3>
<CAPTION> Заголовок таблицы </caption>
<TR><TH bgcolor="lime">&#160;
    <TH bgcolor="lime">Заголовок 1
    <TH bgcolor="lime">Заголовок 2
<TR><TH bgcolor="lime">Заголовок 3
    <TD>Ячейка 1
    <TD>Ячейка 2
</table>
```

Ячейки в таблице могут быть объединены с помощью атрибутов $rowspan = n$, где n ($n > 1$) - количество строк, выделяемых под ячейку или $colspan = k$, где k ($k > 1$) – количество столбцов, выделяемых под ячейку.

Пример 3. Постройте таблицу с объединенными ячейками:

ОБЪЕДИНЕНИЕ ЯЧЕЕК

ВАРИАНТ 1

Заголовок 1	Заголовок 2
Ячейка 1	Ячейка 2
	Ячейка 3
	Ячейка 4

Фрагмент алгоритма на языке HTML представлен ниже:

```
<CENTER>
<FONT color=red><H3>ОБЪЕДИНЕНИЕ ЯЧЕЕК<BR>
ВАРИАНТ 1</h3>
</font>
</center>
<TABLE border=4 cellpadding=0 width=70%>
<TR><TD bgcolor="yellow"><B>Заголовок 1</b>
    <TD bgcolor="yellow"><B>Заголовок 2</b>
```

```

<TR><TD rowspan=3 bgcolor="lime">Ячейка 1
    <TD>Ячейка 2
<TR><TD>Ячейка 3
<TR><TD>Ячейка 4
< /table >

```

При построении таблицы на языке HTML допускается вложение в ячейки других таблиц.

Лабораторная работа №4.

Организация и форматирование таблиц.
Постройте таблицы следующих видов:

Вариант 1

Заголовок таблицы

	Заголовок 1	Заголовок 2		
Заголовок 3	Ячейка 1	Ячейка 1 Таблицы 2	Ячейка 2 Таблицы 2	Ячейка 3
Заголовок 4	Ячейка 2	Ячейка 3 Таблицы 2	Ячейка 4 Таблицы 2	Ячейка 4
Заголовок 5	Ячейка 5			

Вариант 2

Заголовок таблицы

Заголовок 1		Заголовок 2			
Ячейка				Заголовок 3	
Ячейка 1 Таблицы 2	Ячейка 2 Таблицы 2	Ячейка 3 Таблицы 2	Заголовок 4		
Ячейка 4 Таблицы 2	Ячейка 6 Таблицы 2				
Ячейка 5 Таблицы 2					

Вариант 3

Заголовок таблицы

Заголовок 1			Заголовок 2	
Ячейка 1	Ячейка 2			Ячейка 3
Ячейка 4		Заголовок 1 Таблицы 2	Ячейка 6	
	Заголовок 2 Таблицы 2	Ячейка 1 Таблицы 2		
Ячейка 5	Заголовок 3 Таблицы 2	Ячейка 2 Таблицы 2	Ячейка 7	

Вариант 4

Заголовок таблицы

Заголовок 1	Заголовок 2	Заголовок 3		
-------------	-------------	-------------	--	--

Ячейка 1	Ячейка 2	Ячейка 3	Ячейка 4	Ячейка 5
Ячейка 6		Заголовок 1 Таблицы 2	Заголовок 2 Таблицы 2	
Ячейка 7		Ячейка 1 Таблицы 2	Ячейка 2 Таблицы 2	Заголовок 3 Таблицы 2
		Ячейка 3 Таблицы 2		Заголовок 4 Таблицы 2

Вариант 5

Заголовок таблицы

	Заголовок 1			Заголовок 2
Заголовок 3	Ячейка 1	Ячейка 2	Ячейка 3	Ячейка 4
Заголовок 4				Ячейка 5
Заголовок 5	Заголовок 1 Таблицы 2		Заголовок 2 Таблицы 2	Ячейка 6
	Ячейка 1 Таблицы 2	Ячейка 2 Таблицы 2	Ячейка 3 Таблицы 2	Ячейка 7
	Ячейка 4 Таблицы 2	Ячейка 5 Таблицы 2		Ячейка 8

Вариант 6

Заголовок таблицы

Заголовок 1	Заголовок 2	Заголовок 3	Заголовок 4
-------------	-------------	-------------	-------------

Ячейка 1	Ячейка 2		Ячейка 3
Ячейка 4		Заголовок 1 Таблицы 2	Заголовок 2 Таблицы 2
	Заголовок 3 Таблицы 2	Ячейка 1 Таблицы 2	Ячейка 2 Таблицы 2
Ячейка 5	Заголовок 4 Таблицы 2	Ячейка 3 Таблицы 2	

8. Создание форм

Формы (формуляры) превращают WWW в интерактивную среду. Они реализуют простейший способ связи пользователя и сервера.

Форма – это набор полей для ввода текста, командные кнопки и флажки, содержимое которых возвращается серверу по указанию пользователя. Сервер в дальнейшем обрабатывает информацию и, если необходимо, возвращает пользователю ответы.

В дальнейшем рассмотрим использование некоторых тегов для создания форм и на фрагментах одного алгоритма виды таких форм. Тег *< Input >* позволяет создавать различные элементы управления в форме в зависимости от своих атрибутов *type*, *value*, *checked*, *size* и *name*. Значения атрибута *type* представлены в таблице 5.

Таблица 5

Значение атрибута	Функциональное назначение
<i>type = "text"</i>	Создание поля ввода для текстовых данных
<i>type = "password"</i>	Создание строки для ввода пароля
<i>type = "checkbox"</i>	Создание флажков
<i>type = "radio"</i>	Формирование переключателей
<i>type = "submit"</i>	Описание кнопки, нажатие на которую подтверждает ввод информации в форму
<i>type = "reset"</i>	Описание кнопки, для отмены ввода данных в форму
<i>type = "image"</i>	Построение кнопки с рисунком
<i>type = "file"</i>	Средство выбора файла для присоединения к форме

Атрибут *value* используется для определения произвольного текста в поле ввода (*type = "text"*) или для надписи на кнопке (*type = "submit"*).

Атрибут *checked* определяет переключатель, требуемых по умолчанию.

Атрибут *size* описывает длину поля ввода, а атрибут *name* – имя поля (элемента управления).

Пример 1. Построение элементов управления различного вида с использованием тега *< Input >*

СОЗДАНИЕ ФОРМ

ПОЛЯ ВВОДА ЭЛЕМЕНТ INPUT

Ввод текстовой строки

Ввод пароля

Флажки

☒ ☐

Переключатели

☐ ☐ ☒

Кнопка подтверждения ввода

Кнопка с изображением

Кнопка очистки формы

Файл

Код в HTML для построения элементов форм представлен ниже.
<HTML>

```

<HEAD>
<TITLE>Создание форм </TITLE>
<META http-equiv="Content-Type" content="text/html; charset=windows-1251">
</HEAD>
<BODY bgcolor=navy text=lime>
<CENTER><FONT size=6 color=yellow><I>СОЗДАНИЕ
ФОРМ</font></i></center>
<P>
<HR color="red">

```

```

<CENTER>
<FONT color=red><H2>ПОЛЯ ВВОДА <BR> ЭЛЕМЕНТ INPUT</h2>
</font>
</center>
<H3> Ввод текстовой строки </h3>
<INPUT type="text" size=50>
<H3> Ввод пароля </h3>
<INPUT type="password">
<H3> Флажки </h3>
<INPUT type="checkbox" name="F001" checked>
<INPUT type="checkbox" name="F001">
<H3> Переключатели </h3>
<INPUT type="radio" name="S001" value="Первый">
<INPUT type="radio" name="S001" value="Второй">
<INPUT type="radio" name="S001" value="Третий" checked>
<H3> Кнопка подтверждения ввода </h3>
<INPUT type="submit" value="submit">
<H3> Кнопка с изображением </h3>
<INPUT type="image" src="fon01.gif">
<H3> Кнопка очистки формы </h3>
<INPUT type="reset" value="reset">
<H3> Файл </h3>
<INPUT type="file" name="photo" accept="image/*">
<HR color="red">

```

Для создания списка или меню на гипертекстовой странице применяется тег < Select >

< /select > с дополнительным внутренним тегом < Option >.

Тег < Option > предназначен для создания пунктов списка и может иметь два атрибута *selected* – определяет пункт меню, выбираемый по умолчанию, а атрибут *value* требуется для обработки данных на стороне сервера.

Пример 2. Построение элементов управления в формах с помощью тега < Select >.

ЭЛЕМЕНТ SELECT

Первый
Второй
Третий

```
<CENTER>
<FONT color=red><H2>ЭЛЕМЕНТ SELECT </h2>
</font>
</center>
<SELECT multiple>
<OPTION value=a>Первый
<OPTION value=b>Второй
<OPTION value=c>Третий
</select>
```

Выпадающее меню

```
<H3> Выпадающее меню </h3>
<select><option>первый </option>
<option>второй </option>
<option>третий </option>
</select>
<HR color="red">
```

Тег `< Textarea >` `< /textarea >` описывает специальные области для ввода и просмотра текста. Для указания количества строк и столбцов используются атрибуты `rows` и `cols` соответственно.

Пример 3. Формирование области для ввода текста из пяти строк и 30 столбцов.

ЭЛЕМЕНТ TEXTAREA

Фрагмент построения области в коде HTML:

```
<CENTER>
<FONT color=red><H2>ЭЛЕМЕНТ TEXTAREA </h2>
</font>
</center>
<TEXTAREA rows=5 cols=30>
Область для ввода текста
</textarea>
```

Пример 4. За построение кнопки с надписью и рисунком отвечают теги
< Button >
< /button > и тег < Img >.

КНОПКА С НАДПИСЬЮ И РИСУНКОМ

Pencil 

Фрагмент в коде HTML:

```
<CENTER>  
<FONT color=red><H2>КНОПКА С НАДПИСЬЮ И РИСУНКОМ </h2>  
</font>  
</center>  
<BUTTON name="submit" value="submit" type="submit">  
Pencil<IMG src="fon01.gif" alt="Рисунок"></button>
```

Пример 5. Построение группы полей.

ГРУППА ПОЛЕЙ

Заголовок группыИмя: Фамилия:
Телефон:

Фрагмент в коде HTML:

```
<HR color="red">  
  
<CENTER>  
<FONT color=red><H2>ГРУППА ПОЛЕЙ </h2>  
</font>  
</center>  
<FIELDSET>  
<LEGEND>Заголовок группы</legend>  
Имя: <INPUT name="inya2" type="text">  
Фамилия: <INPUT name="familiya2" type="text"><BR>  
Телефон: <INPUT name="telefon2" type="text"><BR>  
</fieldset>  
  
</BODY></HTML>
```

Типовое задание: Постройте форму следующего вида:

ПРИМЕР ФОРМЫ

Имя:	
Фамилия:	
e-mail:	

Страна проживания:	
Научный работник:	☐ Да ☐ Нет
Пол:	☐ М ☐ Ж
Дополнительная информация:	
send reset	

Проанализируйте представленный ниже код HTML и определите технологию построения управляющих элементов сложной формы:

<HTML>

<HEAD>

<TITLE>Пример формы </TITLE>

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

</HEAD>

<BODY bgcolor=blue>

<CENTER><I>ПРИМЕР
ФОРМЫ</i></center>

<P>

<HR color="red">

<TABLE border=3 bgcolor="lime">

<FORM action="mailto:Имя@сервер.домен" method="post">

<TR>

<TD><LABEL for="imya1">Имя: </label>

<TD><INPUT type="text" id="imya1">

<TR>

<TD><LABEL for="familiya1">Фамилия: </label>

<TD><INPUT type="text" id="familiya1">

<TR>

<TD><LABEL for="e-mail1">e-mail: </label>

<TD><INPUT type="text" id="e-mail1">

<TR>

<TD>Страна проживания:

<TD><SELECT><OPTION>Англия </option>

<OPTION>Беларусь </option>

<OPTION>Германия </option>

<OPTION>Франция </option><select>

<TR>

```
<TD>Научный работник:  
<TD>  
<INPUT type="checkbox" name="F001" value="Да">Да  
<INPUT type="checkbox" name="F001" value="Нет">Нет  
<TR>  
<TD>Пол:  
<TD>&nbsp; &nbsp;  
<INPUT type="radio" name="poll" value="Male"> М&nbsp;  
<INPUT type="radio" name="poll" value="Female"> Ж  
<TR>  
<TD>Дополнительная информация:  
<TD><TEXTAREA rows=5 cols=30>  
</textarea>  
<TR>  
<TD align="bottom">  
<INPUT align="bottom" type="submit" value="send">  
<INPUT type="reset" value="reset" >  
</form>  
</table>  
</body>  
</html>
```

Лабораторная работа №5.

Вариант 1

Вопрос 1			
Вопрос 2	Ответ 2.1 ☐		
Вопрос 3	☐ Ответ 2.1	☐ Ответ 2.2	☐ Ответ 2.3
Вопрос 4			
Вопрос 5	☐ Ответ 5.1	☐ Ответ 5.2	☐ Ответ 5.3
Вопрос 6			

Вопрос 7	Ответ 7.1 Ответ 7.2 Ответ 7.3 Ответ 7.4
Дополнительная информация	
send	reset кнопка с рисунком

Вариант 2.

Вопрос 1			send кнопка с надписью и рисунком reset
Вопрос 2	☐ Ответ 2.1	☐ Ответ 2.2	
Вопрос 3	Ответ 3.1 Ответ 3.2 Ответ 3.3		
Вопрос 4	☐ Ответ 4.1	☐ Ответ 4.2	
Вопрос 5			

Вопрос 6	Ответ 6.1	▼	
Дополнительная информация		▲	
		▼	

Вариант 3

кнопка с рисунком send reset	Вопрос 1	Вопрос 2	Вопрос 3
	Вопрос 4	Вопрос 5	Вопрос 6
	Ответ 4.1 ▼	☐ Ответ 5.1 ☐ Ответ 5.2	☐ Ответ 6.1 ☐ Ответ 6.2
	Вопрос 7		Вопрос 8
	Обзор		Ответ 8.1 Ответ 8.2 Ответ 8.3
Дополнительная информация			

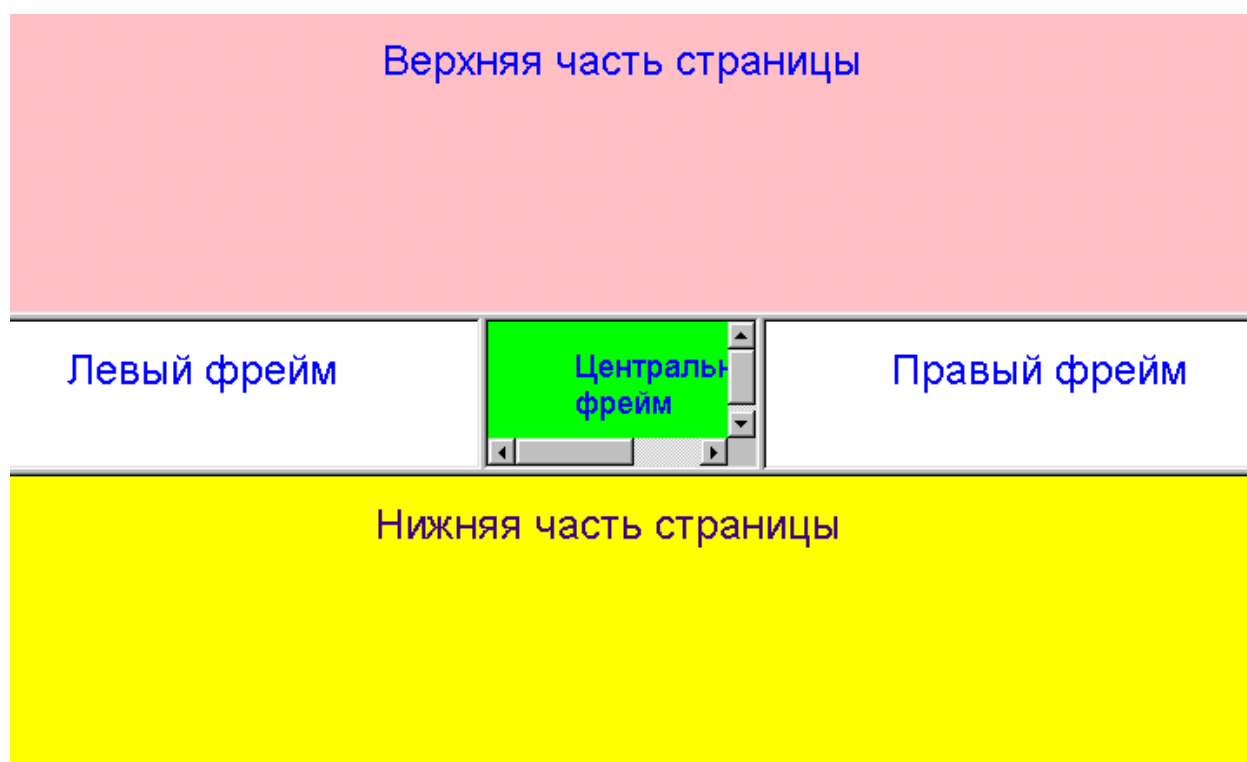
Вариант 4

кнопка с надписью и рисунком send reset		
Вопрос 1		
Вопрос 2	☐ Ответ 2.1	☐ Ответ 2.2

необходимости.

Тег `< Noframes >` `< /noframes >` применяется для предусмотрения ситуации, когда браузер не поддерживает фреймы.

Типовое задание. Разместите на Web-странице пять областей (фреймов), сопроводив центральный фрейм горизонтальной и вертикальной полосами прокрутки, как показано ниже:



Для выполнения данного задания сначала необходимо описать пять отдельных документов содержания для каждой из областей (фреймов). Документ содержания верхнего фрейма:

```

<HTML>
<HEAD>
<TITLE>Верхний фрейм</TITLE>
<META http-equiv="Content-Type" content="text/html; charset=windows-1251">
</HEAD>
<BODY bgcolor="pink">
<CENTER><FONT size=5 color="blue">Верхняя часть
страницы</font></center>
</BODY>
</HTML>

```

Документ содержания нижнего фрейма:

```

<HTML>
<HEAD>
<TITLE>Нижний фрейм</TITLE>
<META http-equiv="Content-Type" content="text/html; charset=windows-1251">
</HEAD>
<BODY bgcolor="yellow">
<CENTER><FONT size=5 color="indigo">Нижняя часть
страницы</font></center>
</BODY></HTML>

```

Документ содержания левого фрейма:

```

<HTML>
<HEAD>
<TITLE>Левый фрейм</TITLE>
<META http-equiv="Content-Type" content="text/html; charset=windows-1251">
</HEAD>
<BODY bgcolor="white">
<CENTER><FONT size=5 color="blue">Левый фрейм</font></center>
</BODY></HTML>

```

Документ содержания центрального фрейма:

```

<HTML>
<HEAD>
<TITLE>Фрейм</TITLE>
<META http-equiv="Content-Type" content="text/html; charset=windows-1251">
</HEAD>
<BODY bgcolor="lime">
<CENTER>
<FONT size=4 color="blue">Центральный фрейм</font>

```

```
</center>
</BODY></HTML>
```

Документ содержания правого фрейма:

```
<HTML>
<HEAD>
<TITLE>Правый фрейм</TITLE>
<META http-equiv="Content-Type" content="text/html; charset=windows-
1251">
</HEAD>
<BODY bgcolor="white">
<CENTER><FONT size=5 color="blue">Правый фрейм</font></center>
</BODY></HTML>
```

После построения документов содержания строится документ раскладка:

```
<HTML>
<HEAD>
<TITLE>Организация фреймов </TITLE>
<META http-equiv="Content-Type" content="text/html; charset=windows-
1251">
</HEAD>
<!-- горизонтальное деление экрана -->
<FRAMESET rows="40%,20%,*">
<!-- вставка документа "topfr.htm" в верхний фрейм -->
<FRAME src="topfr.htm" noresize scrolling="no">
<!-- вертикальное деление центральной области экрана -->
<FRAMESET cols="40%,20%,40%">
<!-- вставка документа "leftfr.htm" в левый фрейм -->
<FRAME src="leftfr.htm" scrolling="no">
<!-- вставка документа "centerfr.htm" в центральный фрейм -->
<FRAME src="centerfr.htm" scrolling="yes" marginwidth="50"
marginheight="15" >
<!-- вставка документа "rightfr.htm" в правый фрейм -->
<FRAME src="rightfr.htm" scrolling="no" >
</frameset>
<!-- вставка документа "bottomfr.htm" в нижний фрейм -->
<FRAME src="bottomfr.htm">
</frameset>
<NOFRAMES>
<CENTER><FONT size=6>Фреймы</font></center>
<HR color="blue">
Этот браузер не может воспроизводить фреймы
</noframes>
</frameset>
```

</HTML>

Таблицы стилей.

Таблицы стилей являются стандартным инструментом форматирования Web-страниц, использующим подходы, характерные для издательских систем. Большинство современных текстовых редакторов позволяют пользователю определять стиль, который будет использоваться для форматирования документа. В частности, можно выбрать стиль абзаца, например, с одинарным межстрочным интервалом и шрифтом Courier. Этот стиль форматирования может быть в дальнейшем применен к любому числу абзацев этого и других документов. Таблицы стилей HTML действуют точно таким же образом.

Преимуществом таблиц стилей HTML является возможность отделить операцию форматирования от содержания документа. Если в готовом документе, например, необходимо заменить цвет шрифта заголовков, для этого достаточно поменять только стиль этих заголовков. Делать изменения в тексте нет необходимости.

В HTML-документе могут использоваться таблицы стилей, созданные на любом языке. Однако, наибольшей популярностью пользуется язык CSS – “Cascading Style Sheets”. Для того, чтобы указать браузеру, какой именно язык используется, необходимо указывать в заголовке HTML-документа: *<META HTTP-EQUIV = “Content = “text /css”>*.

Существует три основных метода применения таблиц стилей к HTML-документу:

- встраивание (Embedding) – можно встроить таблицу стилей в документ с помощью тега *< Style >*;
- связывание (Linking) – можно связать HTML-документ с таблицей стилей, хранящейся в отдельном файле;
- непосредственное задание стиля (Inline) – можно определять стиль используя атрибут *style*.

Встраивание таблицы стилей в HTML-документ

Для включения таблицы стилей в HTML-документ используется тег *< Style >* в заголовке *< Head >* документа:

```
< Head >  
< Style type = “text /css” >  
Определение стиля  
< /style >  
< /head >
```

Тег *< Style >* имеет единственный атрибут *type*, определенный как “*text /css*” для того, чтобы браузеры, не поддерживающие таблицы стилей, могли игнорировать тег *< Style >*.

Задание стиля для одноименных элементов.

Определение стиля для одноименных элементов осуществляется по следующему шаблону:

элемент {свойство 1: значение 1; свойство 2: значение 2;...;свойство n: значение n} (Ш 1)

Пример.

Текст, не отформатированный никаким стилем.

Это стиль для абзацев P во всем документе, назначенный в элементе STYLE.

Это стиль для заголовка 1 во всем документе, назначенный в элементе STYLE.

Это стиль для заголовка 3 во всем документе, назначенный в элементе STYLE.

```
<HTML>
<HEAD>
<TITLE>Таблицы стилей</title>
<META http-equiv="Content-Type" content="text/html; charset=windows-1251">
<META http-equiv="Content-Style-Type" content="text/css">

<STYLE type="text/css">
H1 { border-width: 5;border: outset; text-align: right; color: blue}
/* свойство border устанавливает тип рамки, значение outset говорит о том, что рамка объемная "выпуклая" ;
свойство border-width устанавливает ширину рамки;
свойство text-align задает выравнивание текста */
H3 {color: red; font-style: italic}
/* свойство font-style устанавливает начертание шрифта - обычное, курсив, наклонное (normal, italic, oblique)*/
P {text-indent: 10; color: green; font-size: 12pt}
/*свойство text-indent - определяет отступ для первой строки параграфа;
свойство font-size определяет размер шрифта*/
</style>
</head>

<BODY bgcolor=yellow>
<HR>
Текст, не отформатированный никаким стилем.
<P>Это стиль для абзацев P во всем документе, назначенный в элементе STYLE.</p>
```

<H1>Это стиль для заголовка 1 во всем документе, назначенный в элементе STYLE. </h1>

<H3>Это стиль для заголовка 3 во всем документе, назначенный в элементе STYLE.</h3>

<HR>

</body>

</html>

Использование классов в таблицах стилей.

Гораздо удобнее определить несколько стилей одного элемента, каждый из которых может использоваться в нужном месте страницы. Эти стили называются классами.

Шаблон для задания класса имеет следующий вид:

*элемент.имя_стиля{свойство 1: значение 1; свойство 2: значение 2; ... ;
свойство n: значение n}. (Ш 2)*

Теперь, включая в документ тег элемента, можно указать в нем конкретный стиль при помощи атрибута *class*:

< элемент class = "имя_стиля" >.

Пример.

Использование классов в заголовке H4

Текст абзаца без форматирования.

***Использование класса "new1" для форматирования
текста абзаца.***

*Использование класса "new2" для форматирования
текста абзаца.*

<HTML>

<HEAD>

<TITLE>Таблицы стилей</title>

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

<META http-equiv="Content-Style-Type" content="text/css">

<STYLE type="text/css">

H2.lime {color: lime; text-align: center}

P.new1 {text-indent: 20; color: green; font-size: 15pt;font-style: italic}

P.new2 {color :blue;font-style: italic;letter-spacing: 2pt}

*/*свойство letter-spacing устанавливает дополнительный отступ между
буквами*/*

</style>

</head>

<BODY bgcolor=yellow>

<HR>

<H2 class="lime"> Использование классов в заголовке H4 </h2>

<P> Текст абзаца без форматирования.

`<P class="new1">`Использование класса "new1" для форматирования текста абзаца.

`<P class="new2">`Использование класса "new2" для форматирования текста абзаца.

`<HR>`

`</body>`

`</html>`

Универсальные классы

Классы, которые не связаны с конкретными элементами называются универсальными классами. Они определяются следующим образом:

`#имя_стиля {свойство 1: значение 1; свойство 2: значение 2; ... ; свойство n: значение n}.` (Ш 3)

Этот стиль можно применять ко всем элементам, используя атрибут `id`:

`< элемент id = "имя_стиля" >.`

Пример.

Заголовок формата "fuchsia"

Текст абзаца без форматирования.

Абзац, отформатированный
универсальным стилем "fuchsia"

`<HTML>`

`<HEAD>`

`<TITLE>Таблицы стилей</title>`

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

`<META http-equiv="Content-Style-Type" content="text/css">`

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

`#fuchsia{color:fuchsia; font-size:28pt; font-weight: bold}`

`/*свойство font-weight устанавливает, насколько шрифт должен быть жирным */`

`</style>`

`</head>`

```
<BODY bgcolor=yellow>
<HR>
<H2 id="fuchsia"> Заголовок формата "fuchsia" </h2>
<P>Текст абзаца без форматирования.
<P id="fuchsia"> Абзац, отформатированный универсальным стилем
"fuchsia"</p>
<HR>
</body>
</html>
```

Отметим, что таблица стилей, встроенная с помощью тега `< Style >` в документ, действует только внутри этого документа. Для распространения действия таблицы на другие документы ее необходимо скопировать в каждый из них.

Связывание документа с таблицей стилей

Таблица стилей может создаваться в виде отдельного файла и использоваться для различных Web-страниц. Такая таблица называется каскадной (CSS).

Хранить каскадную таблицу стилей следует в текстовом файле с расширением .css. Его можно создать при помощи любого текстового редактора, используя шаблоны (Ш1, Ш2, Ш3) для определения стиля.

Чтобы подключить документу файл с таблицей стилей используется тег `< Link >` следующим образом:

```
< Link rel = stylesheet href = http://www.mypage.com/mysheet.css type = "text /css" >.
```

В атрибуте *href* указывается URL-адрес таблицы стилей, а атрибуту *type* дается значение `"text /css"`, что позволит браузерам, не поддерживающим таблицы стилей, проигнорировать указанный адрес.

Пример.

Текст абзаца без форматирования.

ИСПОЛЬЗОВАНИЕ КАСКАДНЫХ ТАБЛИЦ ДЛЯ ФОРМАТИРОВАНИЯ ТЕКСТА АБЗАЦА

Использование каскадных таблиц для форматирования заголовка Н3

Заголовок Н2 формата "fuchsia"

```
<HTML>
<HEAD>
<TITLE>Таблицы стилей</title>
```

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

```
<META http-equiv="Content-Style-Type" content="text/css">
```

```
<Link href="MyStyle.css" rel="stylesheet" type="text/css">
```

```
</head>
```

```
<BODY bgcolor=yellow>
```

```
<HR>
```

```
<P>Текст абзаца без форматирования.
```

```
<P class="kaskad">Использование каскадных таблиц для форматирования текста абзаца
```

```
<H3>Использование каскадных таблиц для форматирования заголовка H3</h3>
```

```
<H2 id="fuchsia">Заголовок H2 формата "fuchsia" </h2>
```

```
<HR>
```

```
</body>
```

```
</html>
```

Mystyle

```
P.kaskad {color : green;font-variant: small-caps}
/*свойство font-variant устанавливает режим "малых прописных" */
```

```
H3 {color: red; font-style: italic}
```

```
#fuchsia{color:fuchsia; font-size:28pt; font-weight: bold}
```

Непосредственное задание стиля

Определять стиль можно также и непосредственно, добавляя к нужному тегу атрибут *style* и присваивая ему строковое значение, указывающее новый стиль:

< элемент style="свойство 1: значение 1; свойство 2: значение 2 ; ... ; свойство n: значение n">.

Пример.

Текст, не отформатированный никаким стилем.

Текст, отформатирован с помощью атрибута style.

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Таблицы стилей</title>
```

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

```
<BODY bgcolor=yellow>
```

```
<HR>
```

Текст, не отформатированный никаким стилем.

```
<P style="text-indent: 10;font-size: 10pt; font-style: italic; color: blue">
```

Текст, отформатирован с помощью атрибута style. </p>

```
<HR>
```

```
</body>
```

```
</html>
```

Если непосредственное указание стиля вступает в конфликт с другими методами определения, оно всегда “берет верх”. Например, если при помощи связанного с документом файла стиля определен цвет заголовка H₁ как зеленый, а в самом теге заголовка изменен цвет на красный, то браузер выведет этот заголовок шрифтом красного цвета.

Оглавление

Предисловие

1. История создания языка HTML

2. Основные терминологические понятия

3. Структура гипертекстового документа

4. Формирование текста

Лабораторная работа №1. Формирование текста

5. Гиперссылки и добавление изображений

Лабораторная работа №2 Создание гиперссылок и вставка изображений

6. Построение списков

Лабораторная работа №3 Формирование списков

7. Таблицы

Лабораторная работа №4 Организация и форматирование таблиц

8. Создание форм

Лабораторная работа №5 Построение форм

9. Фреймы

Лабораторная работа №6 Организация фреймов

10. Таблицы стилей

Лабораторная работа №7 Использование таблиц стилей

Литература

Оглавление

Литература

1. Айзенменгер Р. HTML 3.2/4.0. Справочник. М.: БИНОМ, 1998.
2. Гончаров А. HTML в примерах. Санкт-Петербург: ПИТЕР, 1997
3. Браун М., Хоникатт Дж. Использование HTML 4. М., Санкт-Петербург: Вильяме, 1999.
4. Коржинский С. Настольная книга Web-мастера. М.: ЗАО Кнорус, 2000.
5. Воген Т. Мультимедиа. Практическое руководство. Мн.: 1997.
6. Кирмайер М. Мультимедиа. Санкт-Петербург: ВHV, 1994.
7. Werbach К. Краткое руководство по HTML. /<http://werbach.com/barebones>
8. Korpela J. Изучение HTML 3.2 на примерах. /[http://www.hut.fi/~jkorpela/HTML 3.2/all.html](http://www.hut.fi/~jkorpela/HTML%203.2/all.html)