

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, внутрь 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 используемых 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

`Текст будет написан шрифтом Arial.`

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

`Шрифт размера 1
`
`Шрифт размера 2
`
`Шрифт размера 3
`
`Шрифт размера 4
`
`Шрифт размера 5
`
`Шрифт размера 6
`
`Шрифт размера 7
`

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

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

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

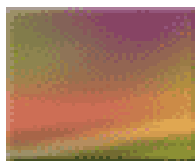
Выравнивание текста

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

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

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

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

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.

Что следует учитывать при тестировании сайта:

- 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:
• 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
<ul type="disc">	Что следует учитывать при тестировании сайта: • работоспособность всех ссылок • поддержку разных браузеров • читабельность текста
<ul type="circle">	Что следует учитывать при тестировании сайта: <input type="checkbox"/> работоспособность всех ссылок <input type="checkbox"/> поддержку разных браузеров <input type="checkbox"/> читабельность текста
<ul type="square">	Что следует учитывать при тестировании сайта: <input checked="" type="checkbox"/> работоспособность всех ссылок <input checked="" type="checkbox"/> поддержку разных браузеров <input checked="" type="checkbox"/> 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
bordercolordark=	#rrggbb	Тень рамки	bordercolordark=#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>
    </TABLE>

```

```

        <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" CELLSPACING="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 **—** ("—").

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:

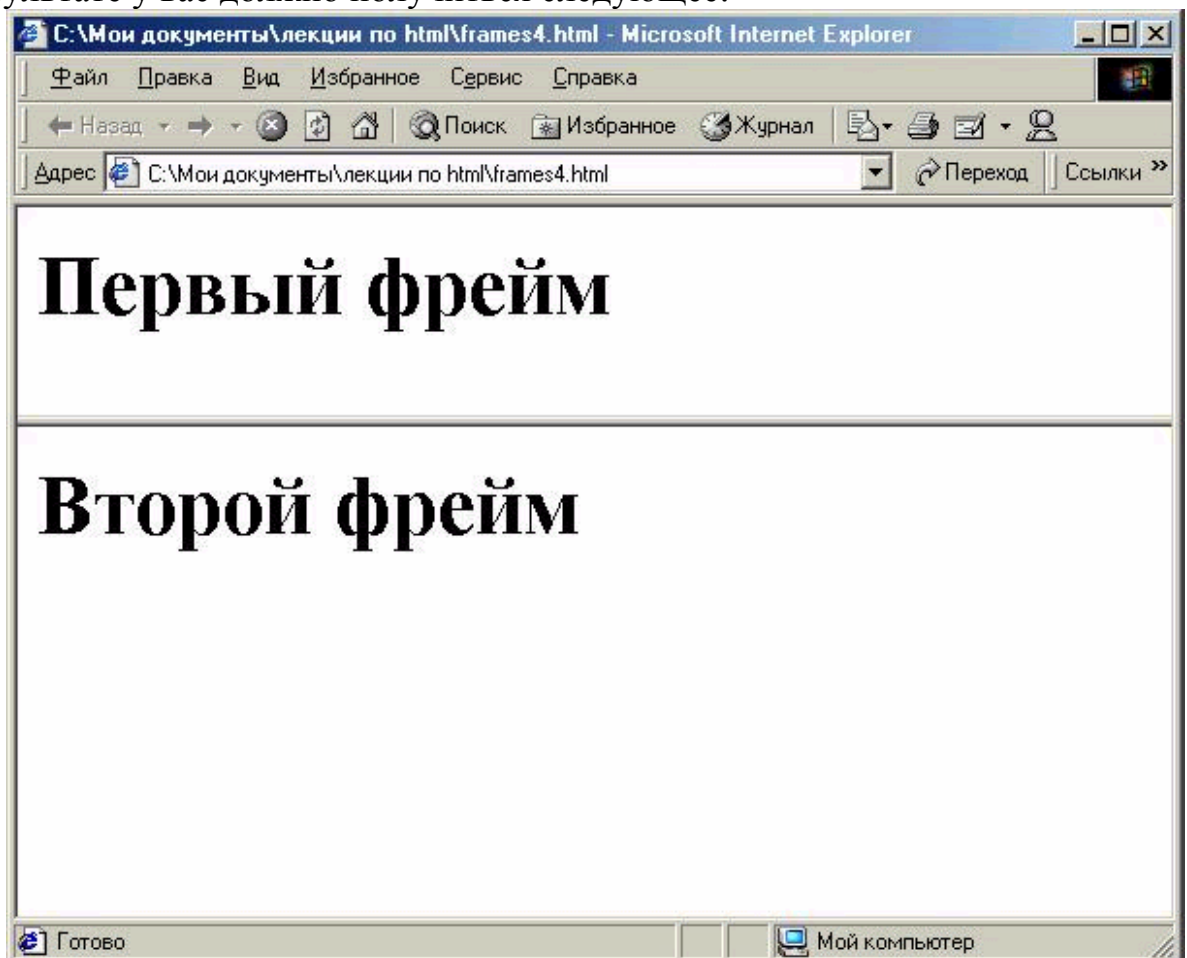
HTML>

```
<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

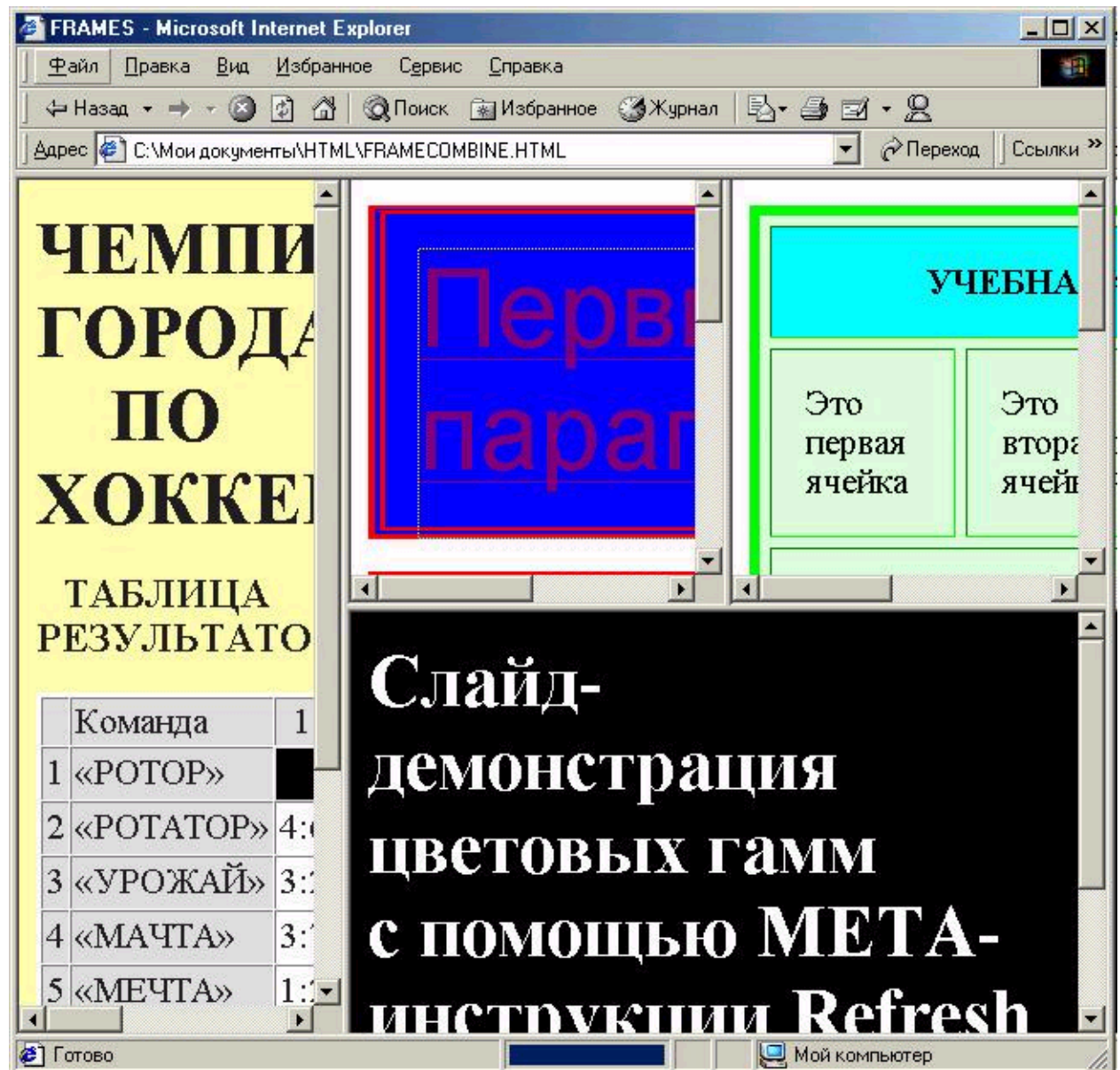
Распространенное явление - комбинация вертикальных и горизонтальных фреймов.

```
<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.