МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

Государственное образовательное учреждение высшего профессионального образования "Оренбургский государственный университет"

Л.Ф. НАСЕЙКИНА

РАЗРАБОТКА WEB-ПРИЛОЖЕНИЙ СРЕДСТВАМИ ЯЗЫКА HTML

Рекомендовано Ученым советом государственного образовательного учреждения высшего профессионального образования "Оренбургский государственный университет" в качестве учебного пособия для студентов, обучающихся по программам высшего профессионального образования по специальности "Программное обеспечение вычислительной техники и автоматизированных систем"

Оренбург 2007

УДК 004.7(075) ББК 32.973.202я73 Н 31

Рецензент доктор технических наук, профессор Пищухин А.М.

Насейкина Л.Ф.

H – 31 Разработка Web-приложений средствами языка HTML: учебное пособие. / Л.Ф. Насейкина. - Оренбург: РИК ГОУ ОГУ, 2007. – 180 с.

ISBN....

Учебное пособие посвящено изучению языка гипертекстовой разметки HTML. В рамках пособия рассматриваются особенности создания простых Web-страниц, с использованием элементов форматирования текста, абзацев, таблиц, графики и мультимедиа.

Также изучается возможность включения в Web-страницы различных элементов Web-дизайна, таких как фреймы, формы, карты изображения. Также рассматриваются вопросы планирования Web-сайта и загрузки готовых страниц на Web-сервер.

Учебное пособие предназначено для студентов, обучающихся по программам высшего образования по специальности 230105.65 – "Программное обеспечение вычислительной техники и автоматизированных систем" при изучении дисциплины "Сетевые информационные технологии".

Н<u>1404000000</u> 6Л9-06

ББК 32.973.202я73

ISBN

© Насейкина Л.Ф., 2007 © РИК ГОУ ОГУ, 2007

Содержание

Введение	5
1 Планирование Web-сайта	6
1.1 Основные этапы планирования сайта	
1.2 Организация информации на Web-узле	6
2 Основы языка HTML	12
2.1 Введение в HTML	12
2.2 Структура HTML-файла	13
3 Оформление текста в HTML	15
3.1 Заголовки	15
3.2 Абзацы и переводы строки	
3.3 Разделы	22
3.4 Списки в HTML	23
3.4.1 Упорядоченные списки	23
3.4.2 Неупорядоченные списки	25
3.4.3 Списки определений	
3.4.4 Необычные списки	
3.5 Еscape-последовательности	
3.6 Логическое форматирование текста	
3.7 Физическое форматирование текста	
3.8 Управление цветами в HTML	
3.9 Форматирование шрифта	
3.10 Горизонтальные линейки	43
3.11 Бегушая строка	45
4 Таблицы	47
4.1 Использование заголовков таблицы	49
4.2 Атрибуты таблиц	
4.2.1 ATDUGYT COLSPAN.	
4.2.2 Атрибут ROWSPAN	
4.2.3 Атрибут WIDTH	
4.2.4 Атрибуты ALIGN и VALIGN	
4.2.5 ATDUGYT CELLPADDING.	
4.2.6 Atpuбvt CELLSPACING.	
4.2.7 Atpuбyt BORDER	
4.2.8 Atpuбyt BGCOLOR	
4.2.9 Atpuбyt BACKGROUND.	
4.3 Сложные таблицы	
4.3.1 Обтекание таблиц	
4.3.2 Вложенность таблиц	
4.3.3 Структурированные таблицы	
5 Гиперссылки	
5.1 Относительные гиперссылки	
5.2 Абсолютные гиперссылки.	

6	Графика и мультимедиа в HTML	. 81
6.1	Графика	81
6.2	Мультимедиа	. 87
6.2.1	Элемент ОВЈЕСТ	. 87
6.2.2	Элемент EMBED	89
6.2.3	Элемент BGSOUND	91
6.3	МІМЕ-типы	92
7	Карты изображений	93
8	Фреймы	97
8.1	Задание фреймсета	97
8.2	Ссылки во фреймах	102
8.3	Плавающие фреймы	106
9	НТМС-формы	. 108
9.1	Задание формы	108
9.2	Задание элементов форм	109
9.2.1	Ter INPUT.	109
9.2.2	Тег SELECT	121
9.2.3	Ter TEXTAREA	126
9.3	Атрибуты форм	. 127
9.3.1	Атрибут АСТІОЛ	128
9.3.2	Атрибут МЕТНОО	128
9.3.3	Атрибут ENCTYPE	129
9.4	Пример формы	130
10	Использование CGI для создания интерактивных интерфейсов	132
10.1	Спецификация CGI	134
10.1.1	Переменные окружения	134
10.1.2	Стандартный вывод	135
10.1.3	Стандартный входной поток	136
10.1.4	Аргументы командной строки	137
10.2	Последовательность действий для обработки входных данных сді-	
	модуля для разных методов запроса GET и POST	137
10.3	Примеры сді-модулей	138
11	Загрузка страницы на Web-сервер	141
Заключ	аение	145
Списон	к использованных источников	146
Прило	жение А Контрольные вопросы для закрепления материала	147
Прилоз	жение Б Тесты	150

Введение

Русская часть сети Internet растет день ото дня. Сегодня в России уже никого не удивишь словосочетанием "домашняя страничка". Если раньше создание Web-страниц было делом профессиональных Web-дизайнеров, то сегодня это под силу даже новичкам.

"Всемирная паутина" World Wide Web является ведущим сервисом сети Internet. Документы, составляющие содержание WWW, называются Web-страницами, а формат, с помощью которого подготавливаются Web-страницы, называется HTML (HyperText Markup Language) или "язык разметки гипертекста".

НТМL является стандартным языком, предназначенным для создания гипертекстовых документов в среде WWW. НТМL представляет собой достаточно простой набор команд, которые описывают структуру документа. НТМL позволяет выделить в тексте отдельные логические части (заголовки, абзацы, списки и т.д.), поместить на Web-страницу отдельно подготовленную фотографию или картинку, организовать на странице ссылки для связи с другими документами.

Целью данного учебного пособия является подготовка студентов к умелому использованию возможностей языка HTML при создании своих собственных Web-сайтов. Изложение материала начинается с вопросов, касающихся правильной организации информации на Web-сайте. Затем рассматриваются простейшие эффекты форматирования HTML-страниц. Такие, как работа с текстом, абзацами, заголовками, списками, таблицами. Представлены особенности с работы с гипертекстовыми ссылками в HTML, особенности абсолютных и относительных гиперссылок.

Далее рассматриваются возможности добавления на HTML-страницу графических и видеофайлов, возможность работы с цветом в языке HTML. Также рассматривается добавление на Web-страницу таких элементов дизайна, как карты изображений, фреймы и формы.

Представлено рассмотрение работы CGI – Common Gateway Interface при обработке данных пользователя из форм. И, наконец, рассматриваются вопросы, касающиеся загрузки готовой Web-страницы на сервер.

После окончания изучения представленного учебного пособия студенты будут уметь разрабатывать общую структуру сайтов, добавлять интерактивные средства и эффекты мультимедиа и, наконец, размещать готовые Web-сайты в Internet путем загрузки их на Web-сервер.

Учебное пособие состоит и введения, одиннадцати глав, заключения, приложений и списка рекомендуемой литературы. Изложение материала во всех разделах опирается на множество примеров. В приложениях представлены контрольные вопросы для закрепления изученного материала, а также тесты для самопроверки.

В заключение, автор выражает признательность рецензенту и научному редактору за внимательное прочтение работы и замечания, способствовавшие улучшению качества предлагаемого учебного пособия.

1 Планирование Web-сайта

1.1 Основные этапы планирования сайта

Вообще, этап планирования - важнейший в разработке любого Webсайта. Это справедливо и для простейшей домашней странички, и для гигантского Web-сайта суперкорпорации. На сегодняшний день выделяются следующие этапы при планировании Web-сайтов.

- 1 Прежде всего, четко определить задачи Web-сайта. Что он должен делать: рассказывать о чем-либо, привлекать клиентов, просвещать или развлекать.
- 2 Определить, какая конкретно информация должна на Web-сайте присутствовать. Что требуется рассказать о себе или о компании.
- 3 Решить, в каком ключе будет выполнен дизайн Web-сайта. Будет ли он консервативным. Соответственно, домашняя страничка должна отражать эстетические наклонности автора, рекламный сайт лучше сделать повеселее, а новостной – поскромнее /1/.
- 4 Придумать логическую и физическую структуру сайта.

1.2 Организация информации на Web-узле

Правильная организация информации на Web-узле часто не менее важна, чем сама информация. Если посетители на Web-узле не смогут найти то, что им нужно, они просто уйдут с него. Под организацией информации понимается размещение сведений на одну и ту же тему в одном и том же месте. Разработчики часто применяют на узле древовидную структуру, схема которой представлена на рисунке 1.



Рисунок 1 - Типичная древовидная схема Web-узла

Подобная древовидная структура может оказаться наиболее приемлемой для Web-узлов. Тем не менее, с ее использованием возникает несколько вопросов: например, сколько можно сделать ответвлений на первом уровне дерева, насколько широким и насколько глубоким должно быть это дерево. Если на верхнем уровне будет слишком много вариантов, дерево будет плоским и в нем можно будет запутаться. Если же дерево будет слишком глубоким, посетителю придется перебираться с уровня на уровень, чтобы добраться до нужной информации. Сравнение глубины с шириной проиллюстрировано на рисунке 2.



Рисунок 2 - Соотношение ширины с глубиной

Узлы c плоской структурой могут ввести пользователя В замешательство количеством возможных вариантов перемещения. Желательно, чтобы ответвлений было от шести до восьми, в таком случае пользователь довольно легко может удержать их в памяти. Слишком большой выбор вариантов будет уже не оптимальным. В этом случае пользователь не просматривает все варианты, а скользит по краям, например, первый пункт, последний пункт или где-нибудь в середине. Слишком малый выбор вариантов может свидетельствовать о том, что создан лишний уровень, длинная последовательность страниц, на которых присутствует только по две ссылки, может замедлить всю работу. Может быть, в данном случае лучше было бы скомпоновать все ссылки на меньшем количестве страниц. Правильный выбор структуры облегчить узла может взаимодействие пользователя с информацией. К сожалению, чаще всего для Web-узлов выбирают древовидную структуру /1/.

Информацию на Web-сервере вовсе не обязательно организовывать в виде дерева. Иногда имеет смысл связать одну страницу с другой в простую цепочку. В форме линейной последовательности можно демонстрировать слайды, вести показ или презентацию чего-либо. Благодаря кнопке броузера "Назад", любое движение по линейному Web-узлу превращается в двустороннее. Идея линейной структуры представлена на рисунке 3.



Рисунок 3 - Пример линейной организации узла

Следующий вариант – линейная последовательность с альтернативами: серия вопросов, отвечая на которые "да" или "нет", пользователь переходит к следующему вопросу. Таким образом, создается обманчивое ощущение взаимодействия с узлом, что привлекает посетителя гораздо больше, чем

чисто линейная структура. Схема линейной структуры с альтернативами представлена на рисунке 4.



Рисунок 4 - Пример линейной структуры с альтернативами

вариант организации информации Еще -ЭТО линейная ОДИН последовательность с переходами. Данная структура, представленная на рисунке 5, подходит для представления побочной информации. Например, во анкетах отвечающего просят в зависимости от многих ответа на поставленный вопрос, ответить на группу следующих вопросов или пропустить ее и перейти к следующей группе. Web-форма подобной анкеты замечательно подходит для представления в виде линейной структуры с переходами. При работе с ней, отвечающий может пропустить ненужные вопросы и при этом не потерять общее направление.



Рисунок 5 - Пример линейной структуры с переходами

Следующий вариант - линейная последовательность с ответвлениями прекрасно подходит для представления какой-либо информации, у которой есть полезные дополнения. Например, электронная презентация цветов в Web имеет ответвление на информацию о пчеловодстве. Пользователь может просмотреть ее, а затем вновь вернуться к главному направлению. Схема структуры представлена на рисунке 6. Данная форма позволяет отклоняться от темы, но при этом сохранять общее направление. Линейная структура с ответвлениями схожа с древовидной структурой, положенной на бок.



Рисунок 6 - Пример линейной структуры с ответвлениями

Одной из необычных форм организации информации, по крайней мере, в Web, является табличная структура. В табличной структуре активно используется пространственная организация материала. Это характерно для многих линейных структур. В таблице существуют понятия "вниз", "вверх", "вправо", "влево". Табличная структура, благодаря своей системности, прекрасно подходит для представления связанных тем, как это делается, например, в каталогах (столбцы таблицы - это определенный ассортимент продукции, а каждая ячейка таблицы - это конкретное изделие того или иного ассортимента). Передвижение от столбца к столбцу аналогично переходу от одного изделия к его аналогу из другого ассортимента. Схема табличной структуры представлена на рисунке 7.

Древовидная, или иерархическая, структура менее системная, чем табличная. Узкое древо сужает выбор пользователя, но для того чтобы добраться до нужной информации, нужно сделать множество щелчков мышью. С другой стороны, в широком древе выбор шире, так что для попадания в нужную точку потребуется меньше щелчков мыши, если только посетитель точно знает, куда ему нужно идти. Чем шире древо, тем более размытой становится структурность.



Рисунок 7 - Пример табличной структуры

Для решения этой дилеммы можно обратиться к смешанной форме. Особая ссылка от самого корня дерева может перевести посетителя сразу же к какой-либо важной информации. Этот прием очень часто используют на Web-узлах, например, на главной странице помещают ссылку на архив бесплатных файлов или на какую-либо другую важную страницу. Где-нибудь в глубине узла можно представить посетителю каталог, так что может пригодиться и табличная структура. А в другой части узла для презентации можно использовать линейную структуру. Сложные Web-узлы пользуются именно смешанной иерархической структурой. Ее схема представлена па рисунке 8.

Проблема структуры смешанного дерева состоит в том, что она очень легко может превратиться в сплетение ссылок без ясной организации (Webструктуру). Всякое понятие о пространственности в Web-структуре отсутствует. Разобраться в этом лабиринте можно только с помощью "указателей" или таких ориентиров, как, например, главная страница узла. Преимущества чистой Web-структуры очевидны лишь в том случае, если пользователь уже знаком с данными /2/.



Переход между страницами осуществляется без проблем, потому что они все связаны друг с другом. Однако подобная организация информации может оказаться настолько запутанной, что отпугнет пользователя от узла. Пример чистой Web-структуры представлен на рисунке 9.

При планировании Web-сайта необходимо найти равновесие между предсказуемостью и выразительностью. Web-форма в некоторых случаях может оказаться очень выразительной, но вот со стороны пользователя - абсолютно непредсказуемой.



Рисунок 9 - Пример Web-структуры

С другой стороны линейная форма предсказуема, но совсем невыразительна. Соотношение этих форм изображено на рисунке 10.



Рисунок 10 - Сравнение линейной и Web-структурной форм

Внедрение предложенных форм проходит несколько этапов:

- Испытание. Проверка на разных системах, в разных броузерах. Проверка пользователями.
- Обслуживание. Обслуживание и обновление узла. Переопределение целей и возврат на стадию планирования.

Эти этапы имеют свои подэтапы, каждый из которых имеет свои детали.

Теперь необходимо спроектированный Web-сайт заполнить соответствующей информацией. При наполнении Web-узла всегда нужно помнить два принципа: уникальность и достоверность публикуемых материалов /2/.

Уникальность является первоочередным требованием к содержанию. В WWW уже может существовать немало страниц с похожими материалами. Web-узел должен чем-то отличаться от серверов с аналогичной тематикой, хотя бы для того, чтобы привлечь к себе внимание. Наличие уникальных материалов на Web-странице увеличит ее посещаемость. Для того, чтобы создать уникальный информационный ресурс не обязательно изобретать чтото принципиально новое, можно по-другому оформить уже существующие ресурсы, но при этом не заставлять клиента тратить много времени на их поиск. Проверить же ресурсы на уникальность можно с помощью поисковых серверов. Что касается авторитетности, то все зависит от того, насколько тщательно подобрана и своевременно обновлена информация.

При создании Web-узла необходимо помнить, что составляющие его отдельные документы должны быть объединены общим стилем оформления и средствами навигации. Единый стиль оформления - один из показателей, отличающих любительский Web-узел от профессионального. Благодаря единообразно сделанным документам пользователи будут отличать данный Web-узел от других и запомнят его. Это не означает, что документы должны быть похожи друг на друга как две капли воды, но общая идея, единый стиль должны присутствовать непременно /2/.

То же относится и к средствам навигации по страницам. Не стоит рассчитывать, что посетитель хорошо знает структуру нового Web-узла. Он должен без труда понимать, где он находится сейчас и как можно попасть в любое другое место. Необходимо предусмотреть возможность перехода к первому документу, программе поиска или к схеме Web-узла.

Кроме того, единство стиля позволяет использовать шаблоны страницы, содержащие только общие элементы оформления. При использовании шаблона для получения готовой страницы достаточно лишь внести в него необходимую информацию. Последовательность, логичность, постоянство - вот необходимые качества хорошего Web-узла

Теперь, когда определена структура и содержание будущего Webсайта, необходимо остановится на выборе средств его разработки. В данном учебном пособии мы остановимся на рассмотрении в качестве инструмента разработки Web-узла язык разметки гипертекста HTML.

2 Основы языка НТМL

2.1 Введение в НТМL

В 1989 году Тим Бернерс-Ли предложил руководству международного центра высоких энергий проект распределенной гипертекстовой системы, которую он назвал World Wide Web (WWW), Всемирная паутина. в том, чтобы при Первоначально идея системы состояла помощи навигационной системы объединить гипертекстовой все множество информационных ресурсов в единую информационную систему. Технология оказалась настолько удачной, что дала толчок к развитию одной из самых популярных в мире глобальных информационных систем. Но и популярность самого Internet во многом вызвана появлением World Wide Web, так как это первая сетевая технология, которая предоставила пользователю простой современный интерфейс для доступа к разнообразным сетевым ресурсам. Простота и удобство применения привели к росту числа пользователей WWW и привлекли внимание коммерческих структур. Далее процесс роста числа пользователей стал лавинообразным, и так продолжается до сих пор.

Успех технологии World Wide Web определен двумя основными факторами: во-первых, простотой и использованием протоколов межсетевого обмена семейства TCP/IP, (Transmission Control Protocol, протокол управления передачей, IP (Internet Protocol, протокол Internet), которые являются основой Internet.

Во-вторых, практически все пользователи Сети одновременно получили возможность попробовать себя в качестве создателей и читателей информационных материалов, опубликованных во Всемирной паутине.

Одним из компонентов технологии создания распределенной гипертекстовой системы World Wide Web стал язык гипертекстовой разметки **HTML - HyperText Markup Language**, разработанный Тимом Бернерсом-Ли на основе стандарта языка разметки печатных документов - SGML (Standard Generalised Markup Language, стандартный обобщенный язык разметки).

При этом разработчики HTML смогли решить две задачи:

- предоставить дизайнерам гипертекстовых баз данных простое средство создания документов;
- сделать это средство достаточно мощным, чтобы отразить имевшиеся на тот момент представления об интерфейсе пользователя гипертекстовых баз данных.

Первая задача была решена за счет выбора теговой модели описания документа. Такая модель широко применяется в системах подготовки документов для печати.

Вторым важным моментом, повлиявшим на судьбу HTML, стало то, что в качестве основы был выбран обычный текстовый файл. Выбор был сделан под влиянием следующих факторов:

- такой файл можно создать в любом текстовом редакторе на любой аппаратной платформе, в среде какой угодно операционной системы;
- к моменту разработки HTML существовал американский стандарт для разработки сетевых информационных систем, в котором в качестве единицы хранения указывался простой текстовый файл в кодировке соответствующей US ASCII /3/.

Таким образом, гипертекстовая база данных в концепции WWW - это набор текстовых файлов, размеченных на языке HTML, который определяет форму представления информации (разметку) и структуру связей между этими файлами и другими информационными ресурсами (гипертекстовые ссылки). Гипертекстовые ссылки, устанавливающие связи между текстовыми документами, постепенно стали объединять самые различные информационные ресурсы, в том числе звук и видео; в результате возникло новое понятие - гипермедиа.

2.2 Структура НТМL-файла

Стандартный документ Internet представляет собой текстовый файл, размеченный средствами формата HTML. HTML является описательным языком разметки документов, в нем используются указатели разметки (теги). Любой тег имеет общий вид:

"контейнер"=

<"имя тега" "список атрибутов"> область действия тега

</ "имя тега">

Сам HTML-документ - это один большой контейнер, который начинается с тега <HTML> и заканчивается тегом </HTML>. Создавать HTML-документы можно как в простом текстовом редакторе, например, в Блокноте Windows, так и с помощью специализированных программ, таких как Microsoft Frontpage.

HTML-файл имеет следующую общую структуру:

```
<HTML>
<HEAD>
<Mета-теги>
<TITLE>Заголовок документа</TITLE>
</HEAD>
<BODY>
Ocновная часть документа
</BODY>
</HTML>
```

Обязательным является только открывающий тег <HTML>, который состоит из двух других вложенных контейнеров: заголовка документа (HEAD) и тела документа (BODY).

Заголовок HTML-документа является необязательным элементом разметки. Существование заголовка определяется необходимостью

именования окна броузера. Это достигается за счет элемента разметки TITLE /4/.

Общие сведения о документе размещаются во вложенных тегах <МЕТА>. Перечислим основные из них:

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

В настоящее время широко распространены сразу несколько способов кодирования национальных алфавитов. Тег, указанный выше, управляет кодировкой документа. Обычно используется одна из двух основных русскоязычных кодировок:

- windows-1251 кодировка Windows от Microsoft, по умолчанию используемая для русского языка во всех приложениях под Windows;
- koi8-r кодировка KOI8-R, используемая для русского языка в операционной системе Unix;

Для современных броузеров, поддерживающих несколько кодировок, указание тага кодировки необязательно. Остальные мета-теги:

<meta name="Generator" content="Имя программы, сгенерировавшей документ">

<meta name="Description" content="Краткое описание содержания страницы">

<meta name= "Keywords" content= "Ключевые слова документа, разделенные запятыми">

<meta name= "Owner" content= "Электронный почтовый адрес владельца страницы">

<meta name= "Category" content="Категория страницы, например, home site">

Мета-теги Description и Keywords используются поисковыми машинами и настоятельно рекомендуются к использованию на любой Webстранице, предназначенной для размещения в Internet. Существуют и другие мета-теги, используемые, например, для автоматической переадресации документа, если Web-страница "переехала" на другой сервер: **«meta http-equiv="Refresh" content="10" URL="новый адрес страницы">. Броузеры поймут эту запись как инструкцию ожидать 10 секунд, а затем загрузить новый документ /4/.**

Необходимо отметить, что при подготовке документов HTML используется идентификатор текста DTD (Document Type Declaration, определение типа документа) в качестве первой строки. Это позволяет броузеру идентифицировать документ как соответствующий стандарту HTML. Обычно (но не обязательно) каждый документ HTML начинается со строки типа: <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

Здесь содержится информация о том, что документ соответствует версии HTML 4.0, разработанной W3C, используемый язык - английский.

Также необходимо отметить о таком важном элементе языка HTML как комментарий. Комментарии в HTML обозначаются следующим образом:

<!--Комментарий ->. 3 Оформление текста

3.1 Заголовки

Существует шесть различных уровней заголовков в языке HTML: <H1>, <H2>, <H3>, <H4>, <H5> и <H6>. Эти уровни располагаются в порядке важности, начиная с <H1> - самого важного до <H6> – наименее важного. Большинство броузеров выводят заголовки буквами большего размера и/или используют жирный шрифт. Размер и толщина шрифта связываются с важностью заголовка, поэтому заголовок <H1> выглядит больше, а <H3> – меньше. Текст, оформленный в виде заголовка, выводится другим стилем (крупнее и/или жирнее), причем на отдельной строке. Кроме того, после заголовка вставляется еще несколько строк. Приведем пример использования заголовков в языке HTML.

<HTML>

<HEAD>

<TITLE>Пример использования заголовков</TITLE>

</HEAD>

<BODY>

<Н1>Заголовок первого уровня </Н1>

<H2>Заголовок второго уровня</H2>

<Н3>Заголовок третьего уровня </Н3>

<Н4>Заголовок четвертого уровня</Н4>

<Н5>Заголовок пятого уровня</Н5>

<Н6>Заголовок шестого уровня</Н6>

</BODY>

</HTML>

Реализация данной HTML-страницы в броузере представлена на рисунке 11.

По умолчанию заголовки выравниваются обычно по левому краю, с помощью атрибута ALIGN заголовок любого уровня можно выровнять по левому, правому краю страницы или по центру. Приведем пример HTML-кода с использованием атрибута ALIGN.

<HTML>

```
<HEAD>
```

<TITLE>Пример выравнивания заголовков</TITLE> </HEAD> <BODY> <H1 ALIGN="LEFT"> По левому краю</H1> <H1 ALIGN="CENTER"> По центру</H1> <H1 ALIGN="RIGHT"> По правому краю</H1> </BODY> </HTML> На рисунке 12 представлена реализация данной HTML-страницы в броузере.



Рисунок 11 – Пример реализации заголовков



Рисунок 12 – Пример выравнивания заголовков

Кроме того, заголовки имеют еще и логическое значение и служат не просто для того, чтобы делать текст крупнее. Руководствуясь заблуждением, что заголовки служат только для укрупнения текста, новички в HTML часто пытаются создать с помощью данного элемента заглавные буквы.

Рассмотрим пример неправильного использования заголовка. <HTML> <HEAD>

```
<TITLE>Пример неправильного использования заголовков</TITLE>
</HEAD>
<BODY>
<H1 ALIGN="LEFT">3</H1>
Заголовки не используются для изменения размеров текста,
например, для создания заглавных букв.
</BODY>
</HTML>
```

На рисунке 13 представлена реализация данного HTML-кода в броузере.



Рисунок 13 – Пример неправильного использования заголовков

3.2 Абзацы и переводы строки

Один из наиболее важных элементов структуризации текста – это элемент абзаца – тег <P>. Текст, окруженный тегами <P> и </P>, означает, что он является логическим абзацем и выводится, как обычно, кроме этого, производится его выравнивание по левому краю. С помощью атрибута ALIGN текст абзаца можно выровнять и по правому краю или по центру. Так как по умолчанию ALIGN принимает значение LEFT (по левому краю), это значение никогда не ставят явно. Рассмотрим возможные значения атрибута ALIGN.

ALIGN=justify выравнивание по левому и правому краям (по ширине).

ALIGN=left выравнивание по левому краю. По умолчанию текст HTML выравнивается по левому краю и не выравнивается по правому. Поскольку выравнивание по левому краю задается автоматически, атрибут ALIGN=left можно опустить.

ALIGN=right выравнивание по правому краю. Текст, выровненный по правому краю и не выровненный по левому.

ALIGN=center выравнивание по центру /5/.

Рассмотрим пример HTML-страницы с использованием тега <P>.

- <hr/>
 <hr/>
- <HEAD>

```
<TITLE>Пример использования выравнивания текста</TITLE>
</HEAD>
```

<BODY>

<Р ALIGN=Left>Текст, выровненный по левому краю</Р>

```
<Р ALIGN=Right>Текст, выровненный по правому краю </Р>
```

```
<Р ALIGN=Center>Текст, выровненный по центру </Р>
```

```
<Р ALIGN=Justify >Текст, выровненный по ширине</Р>
```

</BODY >

```
</HTML>
```

Реализация этой HTML-страницы представлена на рисунке 14.

🖄 Пример использования выравнивания текста - Microsoft Inte 🔳 🗖 🔀				
Файл Правка Вид Избранное Сервис Справка 🧗				
🕞 Назад - 🕥 - 🖹 🗟 🏠 🔎 Поиск 👷 Избранное 🥝 🔗 -				
Адрес: и\ПОСОБИЕ ПО НТМL\Пример выравнивания текста. I 💽 💽 Переход Ссылки 🌺				
Текст, выровненный по левому краю Текст, выровненный по правому краю Текст, выровненный по центру				
Текст, выровненный по ширине 🗸				
🛃 Готово 🧕 🧕 Мой компьютер				

Рисунок 14 – Пример реализации тега <P> с атрибутом ALIGN

Так как вставка элемента <P> приводит к появлению пустой строки, некоторые разработчики пытаются вставить с помощью нескольких элементов <P> несколько пустых строк. Это редко приводит к ожидаемому результату. Броузер игнорирует несколько пустых элементов <P>, так как данный элемент обозначает логическую единицу текста, а не ее физический формат.

Для создания нескольких пустых строк нужно пользоваться элементом
. Это элемент уровня текста, который служит для вставки в текст одного перевода строки. Он ничего не окружает и не имеет закрывающего тега. Следующий фрагмент HTML-кода показывает, как можно использовать
 внутри и вне элемента абзаца.

```
<HTML>
<HEAD>
<TITLE>Пример использования
</TITLE>
```

```
</HEAD>
<BODY>
<BR>
<P>Это первый абзац текста<BR>
Он отделен от второго несколькими тегами BR </P>
<BR><BR><BR>
A это второй абзац. Необходимо обратить внимание на свободное место,
появившееся между абзацами. Это сделано с помощью тегов BR.
```

</BODY>

</HTML>

На рисунке 15 представлена реализация данного HTML-кода выводится в броузере.



Рисунок 15- Пример использования тега

А данный фрагмент HTML-кода показывает, что, несмотря на схожесть, элемент <P> все-таки отличается от
.

```
<HTML>
<HEAD>
<TITLE>Пример перевода строки
</TITLE>
</HEAD>
<BODY>
<P> Это первый абзац.
<BR>
Что тут еще можно сказать?
<P><P><P><P><P><</p>
<P>
<P>A это второй абзац. Обратите внимание, что три тега <P> броузером игнорируются, так как они пусты.
```

</HTML>

Реализация данного HTML-кода представлена на рисунке 16.



Рисунок 16 – Пример сравнения элемента
 и <P>

В теге
 используется также атрибут CLEAR для того, чтобы остановить в указанной точке обтекание объекта текстом и затем продолжить текст в пустой области за объектом. Продолжающийся за объектом текст выравнивается в соответствии со значениями LEFT, RIGHT или ALL атрибута CLEAR.

Рассмотрим пример использования атрибута CLEAR.

<HTML>

<HEAD>

<ТІТLЕ>Пример использования выравнивания тега BR

</TITLE>

</HEAD>

<BODY>

<BR CLEAR=left> Текст будет продолжен, начиная с ближайшего пустого левого поля. Текст будет продолжен, начиная с ближайшего пустого левого поля.

<BR CLEAR=right> Текст будет продолжен, начиная с ближайшего пустого правого поля. Текст будет продолжен, начиная с ближайшего пустого правого поля.

<BR CLEAR=all> Текст будет продолжен, как только и левое, и правое поля окажутся пустыми. Текст будет продолжен, как только и левое, и правое поля окажутся пустыми.

</BODY >

</HTML>

Реализация данного HTML-кода представлена на рисунке 17.

Также при работе с абзацами используется тег <NOBR> (No Break, без обрыва), который дает броузеру команду отображать весь текст в одной строке, не обрывая ее. Если текст, заключенный в теги <NOBR>, не поместится на экране, броузер добавит в нижней части окна документа

горизонтальную полосу прокрутки. Если необходимо оборвать строку в определенном месте, можно поставить там тег
.



Рисунок 17 – Пример использования выравнивания тега


```
Рассмотрим пример использования тега <NOBR>.
<HTML>
<HEAD>
<TITLE>Использование NOBR</TITLE>
</HEAD>
<BODY>
<NOBR>
<H3> Пример использования тега NOBR. Весь текст находится
в одной строке, не обрываясь. </NOBR> </H3>
</BODY>
</HTML>
Пример реализации данной HTML-страницы представлен на рисунке 18.
```



Рисунок 18 – Пример использования тега NOBR

3.3 Разделы

В HTML для структуризации документов в виде разделов используется элемент <DIV>. С помощью атрибута ALIGN любую часть или раздел документа можно выровнять по правой, левой стороне или по центру. По умолчанию текст в элементе <DIV> выравнивается по левой стороне.

Разделы также используются совместно со стилями. В примере и на рисунке 9 демонстрируется использование элементов <CENTER> и <DIV>.

<HTML>

<HEAD>

<TITLE>Пример использования элементов DIV и CENTER</TITLE> </HEAD>

<BODY>

<CENTER>

<H1>Пример элемента CENTER</H1>

<P> Данный абзац выровнен по центру с помощью тега CENTER</P> </CENTER>

<DIV ALIGN= "RIGHT">

<H1>Пример элемента DIV</H1>

<P>Одним элементом DIV можно влиять сразу на несколько абзацев и другие элементы.</P>

<P> Необходимо обратить внимание, что все абзацы выровнены по правой стороне.

</P>

</DIV>

</BODY>

</HTML> На рисунке 19 представлена реализация данного HTML-кода.



Рисунок 19 – Пример использования элементов <DIV> и <CENTER> 3.4 Списки в HTML

Списки являются важным средством структурирования текста и применяются во всех языках разметки. В современном языке HTML используются три основных типа списков: упорядоченные списки Ordered Lists (), неупорядоченные списки Unordered Lists () и списки определений (<DL>). Два других элемента списка – <MENU> и <DIR> используются редко, поддерживаются не всеми броузерами и обычно выводятся как неупорядоченные списки /6/.

3.4.1 Упорядоченные списки

Упорядоченный список, определяемый с помощью тегов и , создает список с нумерацией. Для нумерации могут использоваться арабские цифры, буквы и римские цифры. Упорядоченные списки используются для создания простых перечислений или поэтапных указаний, так как каждый из пунктов списка нумеруется броузером автоматически. Пункты в упорядоченном и прочих списках обозначаются с помощью элемента List Item . Элемент не требует закрывающего тега. Пункты списка, как правило, выводятся броузером с отступами. Нумерация начинается с 1. НТМL-код страницы с использованием упорядоченного списка выглядит следующим образом.

 Первый элемент списка

 Второй элемент списка

 Третий элемент списка

...

Сам элемент имеет три основных атрибута – COMPACT, START и TYPE. Все они могут употребляться по желанию. Атрибут COMPACT не требует никаких значений. Он просто просит броузер вывести список компактнее, чтобы тот занимал меньше места на экране.

Атрибут ТҮРЕ элемента задает тип номера элемента списка и может принимать следующие значения: а – для нумерации строчными буквами, А – для нумерации заглавными буквами, І – для нумерации заглавными римскими цифрами, і – для нумерации строчными римскими цифрами или 1 – для обычной нумерации. Атрибут ТҮРЕ в элементе задает схему нумерации для всего списка. Изменить эту схему можно с помощью атрибута ТҮРЕ элемента .

Каждый элемент может иметь свой собственный атрибут ТҮРЕ, с присвоенным ему значением а, А, I, i или 1. Если в элементе схема нумерации была изменена, она продолжается далее во всем списке до тех пор, пока не встретится очередной элемент с другим значением атрибута ТҮРЕ.

Ниже представлен пример задания различных типов номеров элементов списков.

<OL TYPE="l">Тег создает список с нумерацией в формате 1., 2., 3., 4. и т.д.

<OL TYPE="A">Тег создает список с нумерацией в формате А., В., С., D. и т.д.

<OL TYPE="a">Тег создает список с нумерацией в формате a., b., c., d. и т.д.

<OL TYPE="I">Тег создает список с нумерацией в формате I., II., III., IV. и т.д.

<OL TYPE="i" >Тег создает список с нумерацией в формате i., ii., iii., iv. т.д.

В элементе может также использоваться атрибут START, обозначающий, с какого значения начинать нумерацию списка. Как бы ни определялась нумерация атрибутом TYPE – буквами или цифрами, значение атрибута START всегда должно быть числом, например, для того чтобы начать нумерацию с буквы "j", необходимо набрать код <OL TYPE="a" START= "10">, так как буква "j" является десятой в алфавите. Элемент может переопределить свой номер в списке с помощью атрибута VALUE, который тоже должен иметь числовое значение. Нумерация списка продолжается от указанного значения.

Использование упорядоченных списков проиллюстрировано на примере.

<HTML>

<HEAD><TITLE>Пример упорядоченных списков</TITLE></HEAD> <BODY>

<P>Упорядоченные списки могут быть очень простыми</P>

```
<0L>
<LI>Пункт 1
<LI>Пункт 2
<LI>Пункт 3
</OL>
<Р>Упорядоченные списки могут быть разного типа</Р>
<0L>
<LI ТҮРЕ= "a">со строчными буквами
<LI ТҮРЕ= "А">с заглавными буквами
<LI ТҮРЕ= "i">со строчными римскими цифрами
<LI ТҮРЕ= "I">с заглавными римскими цифрами
<LI ТҮРЕ= "1">с обычными цифрами
</OL>
<Р>Упорядоченные списки могут начинаться с различных значений и
различных типов</Р>
<OL START= "10" TYPE= "a">
<LI>Здесь должна быть буква "j"
<LI>Здесь должна быть буква "k"
<0L>
<LI>Списки можно вкладывать друг в друга
<0L>
<LI>на любую глубину
</OL>
</OL>
</OL>
</BODY>
</HTML>
     В броузере эта HTML-страница будет выглядеть следующим образом:
```



Рисунок 20 – Пример упорядоченных списков

3.4.2 Неупорядоченные списки

Неупорядоченный список обозначается тегами и и используется в тех случаях, когда при перечислении нумерация не важна. Его можно использовать при перечислении функций или достоинств объектов. Броузер обычно добавляет к пункту списка какой-нибудь маркер (сплошной круглый маркер, квадрат или пустой круглый маркер) и выводит список с отступом.

В маркированных списках имеется возможность менять тип маркера с помощью атрибута ТҮРЕ. Атрибут ТҮРЕ может ставиться в элементе $\langle UL \rangle$ и определять тип маркера для всего списка. Кроме того, он может ставиться и в любом из пунктов списка $\langle LI \rangle$. Значение атрибута ТҮРЕ в элементе $\langle LI \rangle$ переопределяет маркер для всех идущих за ним пунктов, если только далее не встретится другой элемент $\langle LI \rangle$ с другим значением ТҮРЕ. Атрибут ТҮРЕ может принимать следующие значения: DISC, CIRCLE И SQUARE. Например,

<UL TYPE=CIRCLE>Ter создает маркеры в виде окружностей <UL TYPE=SQUARE>Ter создает сплошные квадратные маркеры <UL TYPE=DISK> Ter создает сплошные маркеры в виде диска

Пример работы с неупорядоченными списками представлен ниже.

<HTML> <HEAD> <TITLE>Пример неупорядоченных списков</TITLE> </HEAD>

```
<BODY>
<UL><LI>Неупорядоченные списки
<UL><LI>можно вкладывать друг в друга
<UL><LI>Менять тип маркера
</UL></UL>
<P>Маркеры задаются с помощью атрибута ТҮРЕ</P>
<UL TYPE= "square">
<LI>Тип можно задать для
<LI>всего списка или
<LI>для каждого пункта.
<LI TYPE= "disc">Заполненный кружок
<LI TYPE= "circle">Пустой кружок
<LI TYPE= "square">Квадрат
</UL>
```

</BODY>

</HTML>

В броузере эта HTML-страница будет выглядеть следующим образом:



Рисунок 21 - Пример неупорядоченных списков

Помимо использования стандартных маркеров (круг, квадрат, диск) в HTML имеется возможность определять свои собственные маркеры. Маркеры задаются в виде рисунка и указываются в теге . Переопределять маркер можно с помощью тега работы с графикой . Рассмотрим пример использования рисунка в качестве маркера для ненумерованного списка.

<HTML>

<HEAD>

<TITLE>Использование рисунков в качестве маркеров</TITLE> </HEAD>

```
<BODY bgcolor=pink text=maroon>
<H2 align= "center"> Магазин "CBET" предлагает:
</H2>
<UL>
<H3>
<IMG SRC="svet.jpg"> Люстры<BR>
<IMG SRC="svet.jpg"> Светильники<BR>
<IMG SRC="svet.jpg"> Светильники<BR>
<IMG SRC="svet.jpg"> Бра<BR>
<IMG SRC="svet.jpg"> Торшеры <BR>
<IMG SRC="svet.jpg"> Торшеры <BR>
<IMG SRC="svet.jpg"> Настольные лампы</H3>
</UL>
</BODY>
</HTML>
```

Реализация данной HTML-страницы в броузере представлена на рисунке 22._____





Список определений представляет собой список терминов вместе с их определениями. Списки определений помещаются между тегами <DL> и </DL>. Каждый из терминов помещается в элемент <DT> (от слов Definition Term – определяемый термин). Каждое определение помещается в элемент <DD>. Ни для <DT>, ни для <DD> закрывающий тег не нужен, но в случае длинных определений он не помещает /7/. Пример использования элемента <DL> представлен ниже. <HTML>

<HEAD>

<TITLE>Пример списка определений</TITLE>

</HEAD>

<BODY><H2 ALIGN= "CENTER">Определения</H2>

<DL>

<DT> Общение </DT>

<DD> непосредственно переживаемая реальность и конкретизация общественных отношений, их персонификация, личностная форма</DD> <DT> Общение в мультикультурной среде </DT>

<DD>социально-психологическое явление, вмещающее в себя богатое многообразие духовных и материальных

форм сосуществования и взаимодействия людей в конкретном

социальном пространстве разнообразных, равноправных и равноценных культур</DD></DL>

</BODY>

</HTML>

В броузере эта HTML-страница будет выглядеть следующим образом:



Рисунок 23 – Пример списка определений **3.4.4 Необычные списки**

Помимо основных – упорядоченного, неупорядоченного списка и списка определения в HTML имеется еще два: список меню и список каталога. Предназначение элемента <DIR> – выводить списки каталогов. Пункты списка определяются с помощью элемента , так же как и в неупорядоченном списке. Элемент <MENU> можно использовать для создания короткого списка. Пример использования этих необычных типов списков представлен ниже.

<HTML> <HEAD> <TITLE> Пример элемента DIR и MENU</TITLE> </HEAD> <BODY><H2>Пример элемента DIR</H2> <DIR> Это должен быть список каталога Он должен быть компактным Может выводиться столбцами Используется редко</DIR> <H2>Пример элемента MENU</H2> <MENU> Кратный список пункта Компактен Очень компактен Используется редко</MENU>

- </BODY>
- </HTML>



Рисунок 24 – Необычные списки

3.5 Escape-последовательности

Кроме тегов, элементами языка HTML являются CER (Character Entity Reference), они предназначены для представления специальных символов в документе HTML, которые могут быть неверно обработаны броузером. Они также имеют название символьные примитивы.

В частности, это относится к символами "<", ">", "&", "_", "©". Для этого используются соответствующие CER. Например, для знака "&" это будет & amp;. CER легко обнаружить, если посмотреть на структуру любого

документа HTML, поскольку каждый из них начинается с амперсанта "&". В отличие от наименований тегов HTML, наименования CER чувствительны к регистру символов.

Также наименования CER могут задаваться не в виде имени, а с помощью трехзначных кодов символов в виде &#nnn;.

В таблице 1 приведены наиболее часто используемые в HTML CER и соответствующие им числовые коды.

Именная	Символ	Описание
замена		
2	3	4
»	»	Правая угловая кавычка
«	»	Левая угловая кавычка
	[Левая квадратная скобка
]	Правая квадратная скобка
	{	Левая фигурная скобка
	}	Правая фигурная скобка
	?	Вопросительный знак
	%	Процент
&	&	Амперсант
<	<	Меньше
>	>	Больше
		Неразрывный пробел
¡	i	Перевернутый восклицательный знак
§	Ş	Параграф
	\$	Доллар
€	€	Знак евро
¢	¢	Цент
ние таблицы	1	-
2	3	4
£	£	Фунт
¤	¤	Валюта
¥	¥	Йена
©	©	Копирайт
	a	Коммерческое "at"
®	®	Зарегистрированная торговая марка
	+	Знак плюс
	Именная замена 2 » « &laq	Именная заменаСимвол23 $\&$ raquo;> $\&$ laquo;> $\&$ laquo; $\&$ $\&$ laquo; $\&$ $\&$ laquo; $\&$ $\&$ laquo; $\&$ $\&$ amp; $\&$ $\&$ cent; \wp $µ$ $\&$ ceurren; \square $\&$ ceurren; \square \bigotimes $\&$ ven; \pounds $\&$ copy; \bigcirc $@$ $\&$ reg; \mathbb{R} $=$ $=$ \square \blacksquare

Таблица 1 – Наиболее известные Escape-последовательности

& #043;		-	Дефис
& #092;		/	Косая черта (слеш)
=	≡	=	Знак равенства
& #177;	±	±	Плюс или минус

В приведенном ниже примере реализована HTML-страница, на которой использованы различные ессаре-последовательности, а на рисунке 25 представлена реализация данной HTML-страницы в броузере.

<hr/>HTML>

<HEAD> <TITLE> Примеры символьных примитивов</TITLE> </HEAD> <BODY> <H4><P>Символьные примитивы позволяют вставлять в текст специальные символы типа &сору; & ® ¥ </P>
<P>Можно вставлять П P O Б E Л Ы с помощью </Р> <**BR**> <P>Также можно вставлять знаки < > ¡ « » ± ¢ £ ¤

</H4>

</P>

</BODY>

</HTML>



Рисунок 25 - Примеры символьных примитивов

3.6 Логическое форматирование текста

Для логического форматирования текста HTML-документа предусмотрена целая группа тегов. Теги логического форматирования обозначают своими именами структурные типы текстовых фрагментов, такие, например, как программный код, цитата, аббревиатура и т. д.

Логические элементы характеризуют тип окруженного ими содержимого. В таблице 2 представлены наиболее часто употребляемые элементы логического форматирования текста, поддерживаемые большинством броузеров.

Таблица 2 – Таблица элементов логического форматирования текста

Элемент	Свойство элемента
<abbr></abbr>	Сокращение, аббревиатура
<pre><acronym></acronym></pre>	Акроним, аббревиатура
<cite></cite>	Цитата
<code></code>	Исходный код
	Удаленный
<pre>Comparison </pre>	Определение
<ins></ins>	Вставка
	Выделение
<kbd></kbd>	Набор на клавиатуре
<q></q>	Короткие цитаты
<samp></samp>	Пример, образец
	Сильное выделение
<var></var>	Переменная в программе

Рассмотрим пример HTML-страницы, в которой используются все рассмотренные элементы логического форматирования текста.

<HTML>

<HEAD>

<TITLE>Элементы логического форматирования текста</TITLE> </HEAD>

<BODY><H3 ALIGN="CENTER">Элементы логического форматирования текста</H3>

```
<HR>Это <EM>EM - выделение</EM><BR>
Это <STRONG>STRONG - сильное выделение</STRONG><BR>
Это <CITE>CITE - цитата</CITE><BR>
Это <CODE>CODE - исходный код</CODE><BR>
Это <DFN>DFN - определение</DFN><BR>
Это <KBD>KBD - набор на клавиатуре</KBD><BR>
Это <SAMP>SAMP - пример</SAMP><BR>
Это <VAR>VAR - переменная в программе</VAR><BR>
</BODY>
```

```
</HTML>
```

Пример реализации данной HTML-страницы в броузере представлен на рисунке 26.

🚰 Элементы логического форматирования текста - Micros 🔳 🗖	×
Файл Правка Вид Избранное Сервис Справка 🥻	
🕒 Назад - 🐑 - 💽 😭 🏠 🔎 Поиск 👷 Избранное 🪱	>>
Адрес: 🕘 мер логического форматирования текста.html 😒 ラ Переход Ссылки	»
Элементы логического форматирования текста Это ЕМ - выделение Это STRONG - сильное выделение Это CITE - цитата Это CODE – исходный код Это CODE – исходный код Это DFN - определение Это KBD – набор на клавиатуре Это SAMP – пример Это VAR - переменная в программе	
🔊 Farran	Y
ет і отово	117

Рисунок 26 – Элементы логического форматирования

Существуют и другие элементы логического форматирования. форматирования. К ним относят <BLOCKQUOTE>, <Address >, <PRE>.

Блоки цитат. Иногда при цитировании текста большого объема нужно сделать так, чтобы он выделялся из остального текста. Для этой цели может служить элемент <BLOCKQUOTE>. Текст, помещаемый между тегами

<BLOCKQUOTE> и </BLOCKQUOTE>, выводится с отступом. Ниже приведен пример HTML-страницы с использованием тега<BLOCKQUOTE>. <hr/><hr/>HTML>

<HEAD>

<ТІТLЕ>Пример элемента BLOCKQUOTE

</TITLE>

</HEAD>

<BODY><H2 ALIGN="CENTER">Выдержка из автореферата

диссертации на соискание ученой степени кандидата педагогических наук </H2>

<ВLOCKQUOTE> Цель исследования – теоретическое обоснование и экспериментальная проверка комплекса организационнопедагогических требований формирования готовности студентов вуза к общению в мультикультурной среде.

<P>Объект исследования – процесс формирования готовности студентов вуза к общению в мультикультурной среде. </P>

<ВLOCKQUOTE> Предмет исследования – комплекс организационнопедагогических требований формирования готовности студентов вуза к общению в мультикультурной среде как элемент общей и профессиональной подготовки.

</BLOCKQUOTE>

</BODY>

</HTML>

Пример реализации данной HTML-страницы в броузере представлен на рисунке 27.



Рисунок 27 - Пример элемента BLOCKQUOTE

На рисунке видно, что второй и третий абзацы отличаются от второго. В коде HTML-страницы для второго абзаца задан тег $\langle P \rangle$, а для первого и третьего задан тег $\langle BLOCKQUOTE \rangle$, который логически форматирует текст, делая его с отступом.

Адрес. Тег <ADDRESS> также относится к элементам логического форматирования текста. Элементом <ADDRESS> окружают информацию, например, подпись автора страницы или адрес организации, которой посвящена страница. Элемент <ADDRESS> обычно окружен текстом и выводится курсивом. Пример использования тега <ADDRESS> представлен ниже.

<HTML>

<HEAD>

<TITLE>Примеры элемента ADDRESS</TITLE>

</HEAD><BODY><H2 ALIGN= "CENTER">Добро пожаловать на наш информационный сайт</H2><HR>

Благодарим вас за посещение нашего Web-узла. Если у вас возникнут какие-либо вопросы, обращайтесь к нам по адресу:

<HR><ADDRESS>

<P>Кафедра ПОВТАС

Учебно-научно-производственный комплекс

Оренбургского государственного университета (УНПК ОГУ)

460019 г. Оренбург

Шарлыкское шоссе 5</ADDRESS>

</BODY>

</HTML>

Пример реализации данной HTML-страницы в броузере представлен на рисунке 28.



Рисунок 28 – Пример элемента <ADDRESS>
Преформатированный текст. Преформатированный текст также является элементом логического форматирования текста. Применение этого тега позволяет отобразить текст "как есть" (без форматирования), теми же символами и с тем же разбиением на строки.

В некоторых случаях пробелы, знаки табуляции и перевода строки в тексте настолько важны, что принятое в броузере отношение к ним может полностью исказить вид текста. В данном случае необходимо как-то сообщить броузеру, что текст уже отформатирован. Для этого текст следует поместить между тегами <PRE> и </PRE>, и тогда он не будет отформатирован броузером.

В тексте, помещенном между тегами элемента <PRE>, сохраняются все пробелы и символы перевода строк, при изменении размера окна появляются горизонтальная и вертикальная линейка прокрутки. Как правило, броузер выводит преформатированный текст с использованием моноширинного шрифта. Обычно это Courier. Внутри элемента <PRE> можно пользоваться некоторыми элементами форматирования, например , <I>, помещать ссылки. Пример использования элемента <PRE> в сравнении с обычным текстом представлен на примере.

```
<HTML>
```

```
<HEAD><TITLE>Пример элемента PRE</TITLE></HEAD>
<BODY>
<PRE>
 Это ПРЕФОРМАТИРОВАННЫЙ
  Т
   Е
     К
       С
         Т
Все пробелы на месте! А также и символы перевода строки!
</PRE>
<BR><BR>
<P>
А это НЕПРЕФОРМАТИРОВАННЫЙ
 Т
   E
     К
       С
        Т
Пробелы и символы перевода строки пропали.
</P>
</BODY>
</HTML>
```

Пример реализации данной HTML-страницы в броузере представлен на рисунке 29.

🗿 Пример элемента PRE - Microsoft Internet Explorer 📃 🗖 🔯
Файл Правка Вид Избранное Сервис Справка 🦧
🕜 Назад - 🕥 - 💽 🛃 🏠 🔎 Поиск 👷 Избранное 🪱 🧳
Адрес: 🙋 гии\ПОСОБИЕ ПО НТМL\Пример элемента PRE.html 💌 🂽 Переход 🛛 Ссылки 🤌
Это ПРЕФОРМАТИРОВАННЫЙ Т К С Т Все пробелы на месте! А также и символы перевода строки! А это НЕПРЕФОРМАТИРОВАННЫЙТЕКСТ Пробелы и символы перевода строки пропали.
🕙 Готово 🤤 Мой компьютер

Рисунок 29 – Сравнение тега <PRE> с обычным текстом

3.7 Физическое форматирование текста

Иногда для выделения какого-либо текста, например кода программы, бывает необходимо вывести его жирным шрифтом или курсивом или другим способом. Язык HTML имеет несколько элементов, которые можно использовать для физического форматирования. Они не несут никакого другого смысла кроме назначения определенным образом изменить вывод текста на экран. Базовые элементы физического форматирования представлены в таблице 3.

Элемент	Свойство элемента
<i></i>	Курсив
	Жирный
<tt></tt>	Печатная машинка (моноширинный)
<u></u>	Подчеркнутый
<strike></strike>	Перечеркнутый
<s></s>	Еще один элемент перечеркивания
	Нижний индекс
	Верхний индекс
<big></big>	Увеличенный шрифт (на один пункт больше)
<small></small>	Уменьшенный шрифт (на один пункт меньше)
<blink></blink>	Мигающий текст
<pre></pre> /SPAN>	Блок

Таблица 3 – Таблица элементов физического форматирования текста

Рассмотрим пример HTML-страницы с использованием физического форматирования текста.

<HTML>

<HEAD>

<TITLE> Элементы физического форматирования текста</TITLE> </HEAD>

<BODY>

<H3 ALIGN="CENTER">Элементы физического форматирования текста</H3>

<HR>

Это жирный

Это <I>курсив</I>

Это <TT>моноширинный </TT>

Это <U>подчеркнутый</U>

Это <STRIKE>перечеркнутый </STRIKE>

Это тоже <S>перечеркнутый

Это <BIG>большой шрифт </BIG>

Это <SMALL>маленький шрифт </SMALL>

Это текст _{в нижнем индексе}

Это текст ^{в верхнем индексе}

</BODY>

</HTML>

Пример реализации данной HTML-страницы в броузере представлен на рисунке.



Рисунок 30 – Пример физического форматирования текста 3.8 Управление цветами в HTML

В некоторых тегах языка HTML могут быть использованы цвета: например, цвет фона, цвет текста по умолчанию или цвет текста связи. Для спецификации цвета, как значения атрибута, в HTML существуют два варианта:

- символьная нотация, например, MAROON;
- цифровое обозначение, которое контролирует, каким образом цвет формируется из основных цветов - красного, зеленого и голубого - в так называемое RGB цветовое пространство, при этом обозначение должно быть взято в кавычки (комбинация RGB: "#RRGGBB"). Для задания цифрового обозначения цвета в HTML используются шестнадцатеричная и десятичные формы записи. Например, "#800000" и rgb (128,0,0).

Символьные нотации намного легче и более понятны. С другой стороны численные обозначения дают больше возможностей. К ним, в частности относится возможность подбора наиболее близких по цветовой гамме цветов при создании единого стиля страницы.

Чтобы пользоваться предварительно определенными цветовыми именами, необходимо знать их числовые эквиваленты. Наиболее популярные

цвета, используемые в HTML, их символьные и числовые эквиваленты представлены в таблице 4.

Символьн	Шестнад-	Десятерич-	Символьно	Шестнад-	Десятеричная
oe	цатеричная	ная запись	е	цатерич-ная	запись
обозначен	запись		обозначени	запись	
ие			е		
Aqua	"#00FFFF"	0,255,255	Mistryrose	"#FFE4E1"	255,228,225
Black	"#000000"	0,0,0,0	Navy	"#000080"	0,0,128
Blue	"#0000FF"	0,0,255	Olive	"#808000"	128,128,0
Brown	"#A52A2A"	165,42,42	Orange	"#FFA500"	255,165,0"
Coral	"#FF7F50"	255,127,80	Orchid	"#DA70D6"	218,112,214
Darkblue	"#00008B"	0,0,139	Peru	"#CD853F"	205,133,63
Darkgreen	"#006400"	0,100,0	Plum	"#DDA0DD"	221,160,221
Darkred	"#8B0000"	139,0,0	Powderblue	"#B0E0E6"	176,224,230
Fuchsia	"#FF00FF"	255,0,255	Pink	"#FFC0CB"	255,192,203
Gray	"#808080"	128,128,128	Red	"#FF0000"	255,0,0
Green	"#008000"	0,128,0	Royalblue	"#4169E1"	65,105,225"
Hotpink	"#FF69B4"	255,105,180	Seagreen	"#2E8B57"	46,139,87
Lavender	"#E6E6FA"	230,230,250	Skyblue	"#87CEEB"	135,206,235
Lightblue	"#ADD8E6"	173,216,230	Silver	"#COCOCO"	192,192,192
Lime	"#00FF00"	00FF00	Tomato	"#FF6347"	255,99,71
Maroon	"#800000"	128,0,0	Violet	"#EE82EE"	238,130,238
Magenta	"#FF00FF"	255,0,255	White	"#FFFFFF"	255,255,255,2555

Таблица 4 – Стандартные цвета в HTML

Применение цветов в HTML достаточно широко. Можно залить цветом фон HTML-страницы или таблицу, а можно задать цвет для текста, ссылок или рамок. Чтобы задать цвет фона в HTML необходимо в теге <BODY> использовать атрибут BGCOLOR, чтобы задать цвет текста документа необходимо использовать атрибут TEXT. При этом, цвет можно задавать любыми способами: символьно BGCOLOR=Violet, шестнадцатеричной записью BGCOLOR=rgb(238,130,238), либо десятеричной записью BGCOLOR="#EE82EE". Рассмотрим пример простой HTML-страницы с использованием цветов. На странице задан цвет фона небесно-голубой, цвет текста темно-синий.

<HTML>

<HEAD>

<TITLE>Цвета в HTML</TITLE> </HEAD><BODY BGCOLOR="#87CEEB" TEXT="#00008B"> <H2>Использование цвета в HTML</H2> <H3>На данной странице цвет фона задан "#87CEEB" - Skyblue, цвет текста задан "#00008B" - Darkblue </H3> </BODY> </HTML> Пример реализации данной HTML-страницы в броузере представлен на рисунке 31.



Рисунок 31 - Использование цвета в НТМL

Помимо фона страницы и цвета текста можно задавать цвет для гипертекстовых ссылок. Причем для каждого типа ссылок можно задавать отдельный цвет. Простая гиперссылка задается с помощью атрибута LINK, активная гиперссылка задается с помощью атрибута ALINK, и наконец, посещенная с помощью атрибута VLINK тега <BODY>. Например,

<BODY LINK="#0000FF" ALINK="#FF00FF" VLINR= "00FF00"> 3.9 Форматирование шрифта

Помимо рассмотренных выше возможностей форматирования текста, в HTML имеется возможность изменения одновременно всех параметров шрифта (цвет, размер, начертание). Эта возможность реализована в теге .

 TEKCT

Рассмотрим возможные атрибуты.

Size - устанавливает размер шрифта, который будет задан тексту, содержащемуся в пределах элемента . Можно задать абсолютный размер шрифта, указав какое-либо целое число от 1 до 7. Также можно указывать относительный размер, сообщая атрибуту целое число со знаком (это может быть size= "+1" или size= "-2").

Color - указывает цвет, которым будет выделен данный кусок текста. Цвета задаются в виде RGB-значения с шестнадцатеричной нотацией, либо выбирается один из стандартных цветов.

Face - задает гарнитуру шрифта. К ним относятся наиболее популярные семейства шрифтов, такие как Times New Roman, Courier, Arial и т.д

Пример использования тега Font с атрибутом Size представлен ниже. Здесь представлены все известные размеры шрифтов от 1 до 7, другие атрибуты не используются.

<hr/>HTML</hr>

<HEAD>

```
<TITLE>Пример тега font_size </TITLE>
</HEAD>
<BODY BGCOLOR= "#fffff00" TEXT= "#0000ff">
<P><Font size= "1"> Шрифт размера 1</font></P>
<Font size= "2"> Шрифт размера 2</font>
<Font size= "3"> Шрифт размера 3</font>
<Font size= "4"> Шрифт размера 3</font>
<Font size= "4"> Шрифт размера 3</font>
<Font size= "5"> Шрифт размера 4</font>
<Font size= "6"> Шрифт размера 5</font>
<Font size= "6"> Шрифт размера 5</font>
<Font size= "7"> Шрифт размера 7</font>
```

</BODY>

</HTML>

Реализация данного HTML-кода представлена на рисунке 32.

Все атрибуты тега Font можно использовать каждый отдельно, как в ниже представленном примере.

 Текст размера 1

Текст красного цвета

Текст с гарнитурой Arial

Так же можно использовать все атрибуты одновременно. Пример использования всех атрибутов вместе приведен ниже.

 Текст зеленого цвета гарнитуры Courier



Рисунок 32 – Пример работы тега

Рассмотрим пример HTML-страницы с использованием всех атрибутов тега Font.

```
<HTML>
<HEAD>
<TITLE>Пример тега font size color face
</TITLE>
</HEAD>
<BODY>
<Font size= "5" color="#000080" face= "Comic Sans MS">
Текст темно-синего цвета гарнитуры Comic Sans MS</Font>
<Font size= "6" color=Red face= "Mistral" > Текст красного цвета
гарнитуры Mistral </Font>
<Font size= "5" color= "FF00FF" face= "Monotype Corsiva">
Текст цвета фуксии гарнитуры Corsiva </Font>
<Font size= "5" color="Maroon" face="Impact">
Текст темно-красного цвета гарнитуры Impact
</Font>
</BODY>
</HTML>
    В броузере этот HTML-код будет выглядеть следующим образом:
```



Рисунок 33 – Пример работы тега font с различными атрибутами

3.10 Горизонтальные линейки

Горизонтальное отчеркивание (Horizontal Rule) применяется для разделения документа на части. Для этого используется элемент уровня блока, который называется горизонтальная линейка и обозначается тегом <HR>. Элемент <HR> – пустой и не требует закрывающего тега, так как не окружает никаких данных. Имеются несколько атрибутов элемента <HR>. Атрибут SIZE устанавливает толщину (высоту) линейки, WIDTH – ширину, ALIGN управляет выравниванием по горизонтали. Атрибут NOSHADE создает линейку без тени. Наиболее употребительные атрибуты <HR> представлены в таблице 5.

Атрибут	Описание	Значение	Пример
ALIGN	Выравнивает линейку по	LEFT,	<hr< td=""></hr<>
	левой, правой стороне или	RIGHT или	ALIGN="LEFT"
	по центру экрана или	CENTER	WIDTH="50%">
	таблицы		
NOSHADE	Линейка выводится без тени	Нет	<hr noshade=""/>
SIZE	Устанавливает	В пикселях	<hr size="10"/>
	вертикальный размер		
	линейки в пикселях		
WIDTH	Устанавливает ширину	В пикселях	<hr< td=""></hr<>
	линейки относительно	или в	WIDTH="50%">
	ширины экрана в процентах	процентах	<hr/> HR
	или в пикселях		WIDTH="200">

Таблица 5 – Стандартные атрибуты элемента <HR>

Ниже представлены примеры использования горизонтальной линейки в HTML-документе с использованием основных атрибутов.

<HTML> <HEAD> **<TITLE>** Примеры горизонтальных линеек </TITLE> </HEAD> <BODY> HR с толшиной 10 <HR SIZE= "10"> HR толшиной 50% и без тени <HR WIDTH=50% NOSHADE> Ширина 200 пикселей, толщина 3 пикселя, без тени <HR WIDTH= "200" SIZE= "3" NOSHADE> Ширина 100 пикселей, выравнивание по правому краю <HR ALIGN="RIGHT" WIDTH="100"> Ширина 100 пикселей, выравнивание по левому краю <HR ALIGN= "LEFT" WIDTH="100"> Ширина 100 пикселей, выравнивание по центру <HR ALIGN= "CENTER" WIDTH="100"> </BODY> </HTML>

Пример реализации данной HTML-страницы в броузере представлен на рисунке 34.



Рисунок 34 – Примеры использования горизонтальных линеек

3.11 Бегущая строка

Internet Explorer поддерживает тег <MARQUEE>, который позволяет создать, так называемую бегущую строку, то есть эффект прокручивания текста в заданном поле. Характеристики бегущей строки задаются следующими атрибутами:

- Width ширина бегущей строки в пикселях или процентах от ширины окна;
- Height высота поля бегущей строки (в пикселях);
- Hspace, Vspace интервалы по горизонтали и вертикали между текстом строки и краями ее поля (в пикселях);
- Align определяет положение текста бегущей сроки в ее поле, возможные аргументы: top (вверху), bottom (внизу), middle (посередине);
- Direction определяет направление движения, возможные аргументы: left (справа налево), right (слева направо);
- Behavior характер движения строки, возможные аргументы: scroll (текст появляется от одного края и скрывается за другим), slide (строка вытягивается из одного края поля и останавливается у другого края), alternate – задает переменное направление движения, от одного края к другому, а затем обратно;
- Loop число повторения текста в бегущей строке, задаваемое числом, если необходимо бесконечное число повторений, то следует задать аргументов в виде ключевого слова Infinity;
- Scrollamount устанавливает длину в пикселях "прыжка текста" за один такт, при большом значении этого параметра текст движется рывками;
- Scrolldelay определяет величину паузы между тактами перемещения текста в милисекундах;
 - Bgcolor устанавливает цвет поля бегущей строки.

Рассмотрим пример, в котором бегущая строка "бесконечно" прокручивается на желтом поле шириной 75% от ширины окна.

<HTML>

<HEAD>

<ТІТLЕ>Пример использования бегущей строки</ТІТLЕ>

</HEAD>

<BODY>

Пример бегущей строки, которая "бесконечно" прокручивается на желтом поле шириной 90% от ширины окна:

<MARQUEE HEIGHT= 50 WIDTH=75% HSPACE=5 VSPACE=5
ALIGN=TOP BGCOLOR=YELLOW DIRECTION = LEFT</pre>

LOOP=INFINITY BEHAVIOR=SCROLL SCROLLAMOUNT=5 SCROLLDELAY=100> <H1> Это сообщение будет прокручиваться на экране справа на лево. </H1>

```
</MARQUEE>
</BODY>
</HTML>
```

В броузере эта HTML-страница будет выглядеть следующим образом.



Рисунок 35 - Пример использования бегущей строки с параметром DIRECTION = LEFT

В приведенном ниже примере бегущая строка появляется с правого края, т.к. параметр DIRECTION равен RIGHT



Рисунок 36 - Пример использования бегущей строки с параметром DIRECTION = RIGHT

4 Таблицы в HTML

По мере развития WWW стало ясно, что средств, которые заложены в HTML, недостаточно для качественного отображения различного типа документов. Недостатком HTML было отсутствие в его составе средств отображения таблиц. Для этой цели обычно использовался преформатированный текст (тег <PRE>), в котором таблица обрисовывалась символами ASCII. Но такая форма представления таблиц была недостаточно высокого качества и выбивалась из общего стиля документа. После введения таблиц в HTML у Web-мастеров появился не просто инструмент для размещения текстовых и числовых данных, а мощное средство дизайна для расположения в нужном месте экрана графических образов и текста.

Для описания таблиц в HTML используется тег <TABLE>. Тег <TABLE>, как и многие другие, автоматически переводит строку до и после таблицы. Внутри тега <TABLE> используются теги <TR> и <TD>.

Ter < TR > (Table Row, строка таблицы) создает строку таблицы. Весь текст, другие теги и атрибуты, которые требуется поместить в одну строку, должны размещаться между тегами < TR > </TR >.

Внутри строки таблицы обычно размещаются ячейки с данными. Каждая ячейка, содержащая текст или изображение, должна быть окружена тегами <TD></TD>. Число тегов <TD></TD> в строке определяет число ячеек. Рассмотрим пример простой таблицы.

<HTML> <HEAD> <ТІТLЕ>Простая таблица </TITLE> </HEAD> <BODY> <H1 ALIGN= "center">Таблица</H1> <CENTER> <TABLE BORDER= "2"> <TR> <TD COLSPAN=3>Если в таблице два тега TR, то в ней две строки.</TD> </TR></TR> <TD>Если в строке три тега TD, </TD> <TD>то в ней</TD> <TD>три столбца.</TD> </TR> </TABLE> </CENTER> </BODY> </HTML> Пример реализации данной HTML-страницы представлен на рисунке 37.

5

🚰 Простая таблица - Microsoft Internet Explorer	
Файл Правка Вид Избранное Сервис Справка	1
🔇 Назад 🔹 🕥 - 💽 🛃 🚺 🔎 Поиск 🔶 Избранное	
Адрес: 🕘 БИЕ ПО НТМL\Таблицы\Простая таблица.html 💌 🎅 Переход	Ссылки »
Е сли в таблице два тега TR, то в ней две строки.	
Если в строке три тега TD, то в ней три столбца.	
	1

Рисунок 37 – Пример простой таблицы

Помимо тегов <TD> данные таблиц могут быть занесены в теги <TH> <TH></TH> (Table Header, заголовок таблицы). Эти теги подобны <TD></TD>. Отличие состоит в том, что текст, заключенный между тегами <TH></TH> автоматически записывается жирным шрифтом и по умолчанию располагается посередине ячейки. Центрирование можно отменить и выровнять текст по левому или правому краю. Если воспользоваться <TD></TD> с тегом и атрибутом <ALIGN= "center">, текст тоже будет выглядеть как заголовок. Однако следует иметь в виду, что не все броузеры поддерживают в таблицах жирный шрифт, поэтому лучше задавать заголовки таблиц с помощью <TH>. Рассмотрим пример использования заголовков в ячейках таблиц.

<HTML>

<HEAD>

<ТІТLЕ>Заголовки в таблицах</ТІТLЕ>

</HEAD>

<BODY>

<TABLE BORDER= "1">

<TR> <TH>Заголовок центрирован по умолчанию</TH>

<TH COLSPAN=2>Заголовок может объединять столбцы</TH></TR>

<TR> <TH>Заголовок может быть расположен перед столбцами</TH> <TD>Текст или данные</TD>

<TD>Текст или данные (/TD>

</TR>

<TR> <TH ROWSPAN=3>Заголовок может объединять строки</TH>

<TD>Tекст или данные</TD>

TD>Текст или данные</TD>

```
</TR>
<TR>
<TD>Текст или данные</TD>
<TD>Текст или данные</TD>
</TR>
</TR>
<TR> <TD>Текст или данные</TD>
</TR>
</TR>
</TR>
</TR>
</TR>
</HTR>
</HTML>
```

Реализация данной HTML-страницы в броузере представлена на рисунке 38.



Рисунок 38 – Заголовки в ячейках таблиц

Обычно любой текст, не помещающийся в одну строку ячейки таблицы, переходит на следующую строку. Однако можно использовать атрибут NOWRAP с тегами <TH> или <TD>, тогда длина ячейки расширяется настолько, чтобы заключенный в ней текст поместился в одну строку.

4.1 Использование заголовков таблицы

В таблицах для задания заголовков используется тег <CAPTION>. Заголовки центрируются и размещаются либо над (<CAPTION ALIGN= "top">), либо под таблицей (<CAPTION ALIGN= "bottom">). Заголовок

может состоять из любого текста и изображений. Ниже приведен пример использования тега <CAPTION> а таблицах. <HTML> <HEAD> <TITLE>Caption в таблицах</TITLE> </HEAD> <BODY> <TABLE BORDER="3"> <CAPTION ALIGN= "top">Заголовок над таблицей</CAPTION> <TR><TD>Текст или данные</TD> <TD>Текст или данные</TD> <TD>Текст или данные</TD> <TD>Текст или данные</TD> </TR> </TABLE>

 <TABLE BORDER="3"> <CAPTION ALIGN="bottom">Заголовок под таблицей</CAPTION> <TR> <TD>Текст или данные</TD> <TD>Текст или данные</TD> <TD>Текст или данные</TD> </TR> </TABLE> </BODY> </HTML> Так эта HTML-страница будет выглядеть в броузере:



Рисунок 39 - Caption в таблицах

Помимо заголовков, HTML-таблицам можно задавать и другие свойства, такие как ширина таблицы, выравнивание данных в таблицах, толщина рамки и т.д. Все это можно задать, используя основные атрибуты

таблиц. К ним, в частности относятся COLSPAN, ROWSPAN, WIDTH, ALIGN и VALIGN, CELLPADDING, CELLSPACING, BORDER, BGCOLOR, BACKGROUND. Рассмотрим каждый из них подробнее.

4.2 Атрибуты таблиц

4.2.1 Атрибут COLSPAN

Теги <TD> и <TH> модифицируются с помощью атрибута COLSPAN (Column Span, соединение столбцов). Если необходимо сделать какуюнибудь ячейку шире, чем верхняя или нижняя, можно воспользоваться атрибутом COLSPAN, чтобы растянуть ее над любым количеством обычных ячеек. Рассмотрим пример использования атрибута COLSPAN в таблицах.

<HTML> <HEAD> <TITLE>Colspan в таблицах</TITLE> </HEAD> <BODY> <CENTER> <TABLE BORDER="5"> <TR><TD>Если необходимо сделать какую-нибудь ячейку шире, чем верхняя или нижняя,</TD> <TD>можно воспользоваться атрибутом COLSPAN=, </TD></TR> <TR><TD BGCOLOR=white COLSPAN=2>чтобы растянуть ее над любым количеством обычных ячеек. </TD></TR> </TABLE> </CENTER> </BODY> </HTML> В броузере эта HTML-страница будет выглядеть следующим образом: 🖄 Colspan в таблицах - Microsoft Internet Explorer



Рисунок 40 – Использование Colspan в таблицах

4.2.2 Атрибут ROWSPAN

Атрибут ROWSPAN, используемый в тегах <TD> и <TH>, подобен атрибуту COLSPAN, только он задает число строк, на которые растягивается ячейка. Если указано в атрибуте ROWSPAN число большее единицы, то соответствующее количество строк должно находиться под растягиваемой ячейкой. Внизу таблицы ее поместить нельзя.

Нижеследующий пример и рисунок 41 отображает использование атрибутов COLSPAN и ROWSPAN одновременно.

<hr/>HTML>

<HEAD>

<TITLE> Использование параметров COLSPAN и ROWSPAN</TITLE> </HEAD> <BODY> <**BR**> <TABLE BORDER=1 BORDERcolor=black> <TR><TD ROWSPAN=2 > Ячейка, захватывающая две строки </TD> <TD COLSPAN =2> Ячейка, захватывающая два столбца </TD> </TR> <TR> <TD> Ячейка 3 </TD> <TD> Ячейка 4 </TD> </TR> <TR><TD> Ячейка 5 </TD> <TD> Ячейка 6 </TD> <TD> Ячейка 7 </TD> </TR> </TABLE> </BODY> </HTML>

🗿 Использование параметров COLSPAN	и ROWSPAN - Micro	osoft Internet 💶 🗖	×
Файл Правка Вид Избранное Сервис	Справка	_	E
🜀 Назад - 🕥 - 💌 🖻 🏠 🔎	🔵 Поиск	ное 🚱 🔗 🦓	**
Адрес: 🦉 Е ПО НТМL\Использование параметро	в colspan и rowspan.htm	💽 🔁 Переход 🛛 Ссылки	»
			~
[1		
Янейка задатырающая пре столки	Ячейка, захватые	ающая два столбца	
ласика, захватывающая две строки	Ячейка З	Ячейка 4	
Ячейка 5	Ячейка б	Ячейка 7	
			2
A FOTORO	S Mor	X KOMPLIOTOD	
	3 1101	и компьютер	- 44

Рисунок 41 – Использование параметров COLSPAN и ROWSPAN

4.2.3 Атрибут WIDTH

Атрибут WIDTH применяется в двух случаях. Можно поместить его в тег <TABLE>, чтобы дать ширину всей таблицы, а можно использовать в тегах <TR> или <TH>, чтобы задать ширину ячейки или группы ячеек.

Ширину можно указывать в пикселах или в процентах, например, WIDTH=250 либо WIDTH=50%. При задании ширины таблицы в пикселах, необходимо помнить, что различные броузеры могут по-разному отобразить таблицу, тогда проценты всегда отображаются верно.

В зависимости от поставленных задач и тот, и другой способ задания ширины таблицы может оказаться полезным.

Рассмотрим пример использования атрибута WIDTH в таблицах. <HTML> <HEAD> <TITLE> WIDTH в таблицах</TITLE> </HEAD> <BODY> <TABLE BORDER= "3" WIDTH=100%> <TR> <TD ALIGN=center>Текст или данные — ширина 100%</TD> </TR> </TABLE> или
 <TABLE BORDER WIDTH=50%> <TR> <TD ALIGN=center>Текст или данные — ширина 50%</TD> </TR> </TABLE> или
 <TABLE BORDER WIDTH=200> <TR> <TD ALIGN=center>Teкст или данные — ширина 200 пикселов</TD> </TR> </TABLE> или <**BR**> <TABLE BORDER WIDTH=100> <TR> <TD ALIGN=center>Teкст или данные — ширина 100 пикселов</TD> </TR> </TABLE> </BODY> </HTML>

В броузере эта HTML-страница будет выглядеть следующим образом:

🖉 WIDTH в таблицах - Microsoft Internet Explorer 📃 🗖	×
Файл Правка Вид Избранное Сервис Справка 🦧	1
🜀 Назад - 🕥 - 💽 🛃 🏠 🔎 Поиск 👷 Избранное 🤣 🔗 - 🌺	»
Адрес: 🕗 ологии\ПОСОБИЕ ПО HTML\Таблицы\Width в таблицах.html 💌 🋃 Переход 🛛 Ссылки	»
Текст или данные — ширина 100%	~
или	
Текст или данные — ширина 50%	
или	
Текст или данные — ширина 200 пикселов	
или	
Текст или	
данные —	
ширина 100	
Пикселов	-
	~
🕮 Готово 🤤 Мой компьютер	11

Рисунок 42 – Использование атрибута WIDTH в таблицах

Если ячейка не содержит данных, она не будет иметь границ. Если требуется, чтобы у ячейки были границы, но не было содержимого, необходимо поместить в нее что-то, что не будет видно при просмотре. Можно воспользоваться пустой строкой
. Можно даже задать пустые столбцы, определив их ширину в пикселах или относительных единицах и не введя в полученные ячейки никаких данных. Это средство может оказаться полезным при размещении на странице текста и графики.

4.2.4 Атрибуты ALIGN и VALIGN

Теги <TR>, <TD> и <TH> можно модифицировать с помощью атрибутов ALIGN и VALIGN. Атрибут ALIGN определяет выравнивание текста и графики по горизонтали, то есть по левому, правому краю, либо по центру. Горизонтальное выравнивание может быть задано несколькими способами:

ALIGN=bleedleft прижимает содержимое ячейки вплотную к левому краю.

ALIGN=left выравнивает содержимое ячейки по левому краю с учетом отступа, заданного атрибутом CELLPADDING.

ALIGN=center располагает содержимое ячейки по центру.

ALIGN=right выравнивает содержимое ячейки по правому краю с учетом отступа, заданного атрибутом CELLPADDING /7/.

Рассмотрим простой пример использования тега ALIGN.

```
<HTML>
<HEAD>
<TITLE> Выравнивание элементов таблицы</TITLE>
</HEAD>
<BODY>
<TABLE BORDER= "3" bordercolor=black WIDTH=100%>
<TR>
<TD ALIGN=RIGHT> Ячейка 1 </TD>
<TD ALIGN=CENTER> Ячейка 2 </TD>
<TD> Ячейка 3 </TD>
</TR>
<TR>
<TD ALIGN=RIGHT> Ячейка 4 </TD>
<TD ALIGN=CENTER> Ячейка 5 </TD>
<TD> Ячейка 6 </TD>
</TR>
</TABLE>
</BODY>
```

```
</HTML>
```

Пример реализации данной HTML-страницы представлен ниже.

🗿 Выравнивание элементов та	блицы - Microsoft Inter	net Explorer	
Файл Правка Вид Избранное	Сервис Справка		1
🚱 Назад - 🕥 - 💌 😰	🏠 🔎 Поиск 👷 И	збранное 🧭 🎯 🍓	× *
Адрес: хнологии\ПОСОБИЕ ПО НТМ	ML\Выравнивание элементов	таблицы.html 🔽 🎦 Перехо,	д Ссылки »
			~
Ячейка 1	Ячейка 2	Ячейка 3	
Ячейка 4	Ячейка 5	Ячейка б	
			1
🙆 Готово		🚽 Мой компьютер	

Рисунок 43 - Пример использования атрибута ALIGN

Рассмотрим другой пример использования тега Align. <HTML> <HEAD> <TITLE> ALIGN в таблицах</TITLE> </HEAD> <BODY>
 <TABLE BORDER="2" WIDTH=100% BORDERcolor=black> <TR> <TD ALIGN=left>Teкст по левому краю</TD> <TD ALIGN=center>Tекст по центру</TD>

<TD ALIGN=right>Текст по правому краю</TD>

```
</TR>
<TR> <TD ALIGN=right>Teкст по правому краю</TD>
<TD ALIGN=center>Tекст по центру</TD>
<TD ALIGN=left>Teкст по левому краю</TD>
</TR>
<TR> <TD ALIGN=right>Teкст по правому краю</TD>
<TD ALIGN=right>Текст по правому краю</TD>
<TD ALIGN=right>Текст по правому краю</TD>
</TR>
<TR> <TD ALIGN=center>Teкст по центру </TD>
<TD ALIGN=center>Tекст по центру</TD>
<TD ALIGN=center>Текст по центру</TD>
</TR>
<TR>
<TD ALIGN=left>Teкст по левому краю</TD>
<TD ALIGN=left>Tекст по левому краю</TD>
<TD ALIGN=left>Tекст по левому краю</TD>
</TR>
</TABLE>
</BODY>
```

</HTML>

Пример реализации данной HTML-страницы представлен ниже.

ALIGN в таблицах - Microsoft In	ternet Explorer	
Файл Правка Вид Избранное С	ервис Справка	
🔇 Назад 🕤 🕥 - 💌 🛃 🃢	🏠 🔎 Поиск לא	3 🗟 - 😓 🔯 - 🔂 🚳
дрес: 🙋 информационные технологии	и\ПОСОБИЕ ПО HTML\Таблицы\Align в та	блицах.html 🔽 🋃 Переход 🔹 Ссылки 🤅
		2
Текст по левому краю	Текст по центру	Текст по правому краю
Текст по правому краю	Текст по центру	Текст по левому краю
Текст по правому краю	Текст по правому краю	Текст по правому краю
Текст по центру	Текст по центру	Текст по центру
Текст по левому краю	Текст по левому краю	Текст по левому краю
		3
Готово		😼 Мой компьютер

Рисунок 44 – Использование ALIGN в таблицах

Атрибут VALIGN аналогичен ALIGN, однако осуществляет выравнивание текста и графики внутри ячейки по вертикали. Вертикальное выравнивание может быть задано несколькими способами:

VALIGN=top выравнивает содержимое ячейки по ее верхней границе.

VALIGN=middle центрирует содержимое ячейки по вертикали.

VALIGN=bottom выравнивает содержимое ячейки по ее нижней границе.

Рассмотрим пример использования атрибута VALIGN в таблицах. <HTML> <HEAD> <TITLE> VALIGN в таблицах </TITLE> $\langle HEAD \rangle$ <BODY> <CENTER> <TABLE BORDER WIDTH=90%> <TR> **<TD WIDTH=100>Атрибут VALIGN осуществляет выравнивание текста** и графики внутри ячейки по вертикали.</TD> <TD VALIGN=top>Bepx,</TD> <TD VALIGN=middle>середина,</TD> <TD VALIGN=bottom>низ.</TD> </TR> <TR VALIGN=top> <TD> VALIGN=top Выравнивает содержимое ячейки по ее верхней границе.</TD> <TD>верх,</TD> <TD>Bepx,</TD> <TD>верх.</TD> </TR> <TR VALIGN=middle> <TD>VALIGN=middle Центрирует содержимое ячейки по вертикали.</TD> <TD>середина,</TD> <TD>середина,</TD> <TD>середина.</TD> </TR> <TR VALIGN=bottom> <TD>VALIGN=bottom Выравнивает содержимое ячейки по ее нижней границе.</TD> <ТD>низ.</ТD> <TD>низ,</TD> <ТD>низ.</ТD> </TR> </TABLE> </CENTER>

</BODY>

</HTML>

Пример реализации данной HTML-страницы представлен ниже.

🚰 VALIGN в таблицах - Microsoft Internet Exp	lorer			
Файл Правка Вид Избранное Сервис Справ	жа			<u></u>
🔇 Назад - 🚫 - 💌 😰 🏠 🔎 Пои	іск 👷 Избра	анное 🧭	8.3	×
	/align в таблица	av btml 🔽 🖪	Переход	Ссылки Х
Атрибут VALIGN	верх,			
осуществляет выравнивание текста и графики		середина,		
внутри ячеики по вертикали.			низ.	
VALIGN=top Выравнивает содержимое ячейки по ее верхней границе.	верх,	верх,	верх.	
VALIGN=middle Центрирует содержимое ячейки по вертикали.	середина,	середина,	середина.	
VALIGN=bottom Выравнивает содержимое ячейки по ее нижней границе.	низ,	низ,	низ	
Готово		😼 Мой компы	ютер	2 N

Рисунок 45 – Использование VALIGN в таблицах

4.2.5 Атрибут CELLPADDING

Данный атрибут определяет ширину пустого пространства между содержимым ячейки и ее границами, то есть задает поля внутри ячейки. Пример таблицы с использованием атрибута CELLPADDING представлен ниже.

```
<HTML>
<HEAD>
<TITLE> Использование CELLPADDING </TITLE>
</HEAD>
<BODY>
<TABLE BORDER="1" BORDERCOLOR=BLACK CELLPADDING=10
WIDTH=70%>
<TR ALIGN="CENTER"><TH>Год</TH><TH>Продажи</TH></TR>
<TR ALIGN="CENTER"><TD>2000</TD><TD>18$</TD></TR>
<TR ALIGN="CENTER"><TD>2000</TD><TD>18$</TD></TR>
<TR ALIGN="CENTER"><TD>2001</TD><TD>20$</TD></TR>
<TR ALIGN="CENTER"><TD>2001</TD><TD>20$</TD></TR>
<TR ALIGN="CENTER"><TD>2002</TD><TD>22$</TD></TR>
<TR ALIGN="CENTER"><TD>2003</TD><TD>24$</TD></TR>
<TR ALIGN="CENTER"><TD>2004</TD><TD>26$</TD></TR>
<TR ALIGN="CENTER"><TD>2004</TD><TD>26$</TD></TR>
```

</TABLE> </BODY> </HTML>

В броузере эта HTML-страница будет выглядеть следующим образом:

🚰 Использования CEL	LPADDING - Microsoft Internet Explorer	
Файл Правка Вид И	1збранное Сервис Справка	20
🕝 Назад 👻 🕥 т	🖹 🛃 🏠 🔎 Поиск 🐈 Избранное	🚱 🔗 🍓 🖸 • 🎽
Адрес: 🖉 ционные техно	логии\ПОСОБИЕ ПО HTML\использование cellpadd	ing.html 🔽 🛃 Переход 🛛 Ссылки 🌺
		7
Год	Продажи	
2000	18\$	
2001	20\$	
2002	22\$	
2003	24\$	
2004	26\$	
2005	28\$	
🙆 Готово		🔮 Мой компьютер 🦪

Рисунок 46 – Использование CELLPADDING

А теперь в качестве сравнения приведем две таблицы. В одной из них используется атрибут Cellpadding, а в другой не используется.

```
<HTML>
<HEAD>
<TITLE> CELLPADDING в таблицах</TITLE>
</HEAD>
<BODY>
<CENTER>
<TABLE BORDER="3" CELLPADDING=20>
<TR>
<TD>Текст или данные</TD>
<TD>Текст или данные</TD>
<TD>Текст или данные</TD>
</TR>
<TR>
<TD>Текст или данные</TD>
<TD>Текст или данные</TD>
<TD>Текст или данные</TD>
</TR>
</TABLE>
<BR>
```

<TABLE BORDER= "3" CELLPADDING=0> <TR> <TD>Tекст или данные</TD> <TD>Tекст или данные</TD> <TD>Tекст или данные</TD> </TR> <TD>Tекст или данные</TD> <TD>Tекст или данные</TD> <TD>Tекст или данные</TD> </TR> </TR> </CENTER> </BODY> </HTML>

В броузере эта HTML-страница будет выглядеть следующим образом:

Текст или данные Текст или данные Текст или данные Текст или данные Текст или данные Текст или данные) Назад — 💽 — 🛓 ес: 🛃 особие по нтмі	🚺 🛃 🎧 🔎 Поиск .\Таблицы\CeLLpadding в табл	🦟 Избранное 🧭 🖾 🕶 ицах.html 🔽 🂽 Переход 🛛 Ссыл
Текст или Текст или Текст или данные данные	Текст или	Текст или	Текст или
	данные	данные	данные
	Текст или	Текст или	Текст или
	данные	данные	данные

Рисунок – 47 Сравнение таблиц с использование атрибута CELLPADDING

A теперь приведем пример таблицы, в которой используются атрибуты COLSPAN, ROWSPAN и CELLPADDING вместе. <HTML> <HEAD> <TITLE> Использование параметров COLSPAN и ROWSPAN и CELLPADDING </TITLE> </HEAD> <BODY>

```
<TABLE BORDER="1" CELLPADDING="10" WIDTH="80%"
BORDERCOLOR = black>
<TR align="center">
<TH ROWSPAN="2">Год</TH><TH COLSPAN="3">Продажи</TH>
</TR>
<TR align="center"><TH>Ceвер</TH><TH>Юг</TH>
<TH>Итого</TH></TR>
<TR align="center"><TD>2000</TD><TD >20$</TD><TD>25$</TD>
<TD>45$</TD></TR>
<TR align="center"><TD>2001</TD><TD >30$</TD><TD>26$</TD>
<TD>56$</TD></TR>
<TR align="center"><TD>2002</TD><TD >40$</TD><TD>27$</TD>
<TD>67$</TD></TR>
<TR align="center"><TD>2003</TD><TD>50$</TD><TD>28$</TD>
<TD>78$</TD></TR>
<TR align="center"><TD>2004</TD><TD >60$</TD><TD>29$</TD>
<TD>89$</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

Пример реализации данной HTML-страницы представлен ниже.

ил Правка Назад - (ес: 🛃 по нті	ыд Избранное	Сервис Спр	равка оиск 🔶 Избран Блицах.html 💙 통	нное 🧭	Ссылк
		Продажи]	
Год	Север	Юг	Итого		
2000	20\$	25\$	45\$		
2001	30\$	26\$	56\$		
2002	40\$	27\$	67\$		
2003	50\$	28\$	78\$		
2004	60\$	29\$	89\$	1	

Рисунок 48 – Использование атрибутов COLSPAN, ROWSPAN и CELLPADDING

4.2.6 Атрибут CELLSPACING

Атрибут CELLSPACING определяет ширину промежутков между ячейками в пикселах. Если этот атрибут не указан, по умолчанию задается величина, равная двум пикселам. С помощью атрибута CELLSPACING можно размещать текст и графику там, где вам нужно. Если необходимо оставить пустое место, можно вписать в ячейку пробел.

Рассмотрим пример использования атрибута CELLSPACING.

```
<HTML>
<HEAD>
<TITLE> CELLSPACING в таблицах</TITLE>
</HEAD>
<BODY>
<CENTER>
<TABLE BORDER= "2" CELLSPACING=20>
<TR>
<TD>Текст или данные</TD>
<TD>Текст или данные</TD>
<TD>Текст или данные</TD>
</TR>
<TR>
<TD>Текст или данные</TD>
<TD>Текст или данные</TD>
<TD>Текст или данные</TD>
</TR>
</TABLE>
<BR><BR>
<TABLE BORDER= "2" CELLSPACING=0>
<TR>
<TD>Текст или данные</TD>
<TD>Текст или данные</TD>
<TD>Текст или данные</TD>
</TR>
<TR>
<TD>Текст или данные</TD>
<TD>Текст или данные </TD>
<TD>Текст или данные</TD>
</TR>
</TABLE>
</CENTER>
</BODY>
</HTML>
    Пример реализации данной HTML-страницы представлен на рисунке
49.
```



Рисунок 49 – Использование атрибута CELLSPACING

4.2.7 Атрибут BORDER

В теге <TABLE> часто определяют, как будут выглядеть рамки, то есть линии, окружающие ячейки таблицы и саму таблицу. Если не задать рамку, то получится таблица без линий, но пространство под них будет отведено. Того же результата можно добиться, задав <TABLE BORDER=0>. Иногда хочется сделать границу потолще, чтобы она лучше выделялась. Можно для привлечения внимания к рисунку или тексту задать исключительно жирные границы. При создании вложенных таблиц приходится делать для разных таблиц границы различной толщины, чтобы их легче было различать.

Рассмотрим пример HTML-страницы с использованием различных значений атрибута BORDER. На странице использованы три аналогичные таблицы, однако у первой таблицы атрибут BORDER равен 1, у второй 21, а у третьей отсутствует, BORDER=0.

```
<HTML>
<HEAD>
<TITLE>Пример использования Border
</TITLE>
<BODY>
<P>У первой таблицы атрибут BORDER равен 1</P>
<TABLE BORDER= "1" BORDERCOLOR=BLACK>
<TR>
<TD>Фамилия</TD>
<TD>Фамилия</TD>
<TD>Имя</TD>
```

```
<ТD>Стаж</ТD>
</TR>
<TR>
<TD>Иванов</TD>
<TD>Иван </TD>
<TD>Иванович</TD>
<TD>10 лет</TD>
</TR>
</TABLE>
<P>У второй таблицы атрибут BORDER равен 21
</P>
<TABLE BORDER= "21" BORDERCOLOR=BLACK>
<TR>
<TD>Фамилия</TD>
<TD>Имя</TD>
<ТD>Отчество</ТD>
<ТD>Стаж</ТD>
</TR>
<TR>
<TD>Иванов</TD>
<ТD>Иван </ТD>
<TD>Иванович</TD>
<TD>10 лет</TD>
</TR>
</TABLE>
<Р>У третьей таблицы атрибут BORDER равен 0
</P>
<TABLE BORDER= "0" BORDERCOLOR=BLACK>
<TR>
<TD>Фамилия</TD>
<TD>Имя</TD>
<ТD>Отчество</ТD>
<ТD>Стаж</ТD>
</TR>
<TR>
<TD>Иванов</TD>
<TD>Иван </TD>
<TD>Иванович</TD>
<TD>10 лет</TD>
</TR>
</TABLE>
</BODY>
</HTML>
    В броузере эта HTML-страница будет выглядеть следующим образом.
```

🗿 Пример использования Border - Microsoft Internet Explorer 📃 🗖	K
Файл Правка Вид Избранное Сервис Справка 🥂	1
🜀 Назад - 🕥 - 🖹 🗟 🏠 🔎 Поиск 👷 Избранное 🧭 🔗 - 🌺	»
Адрес: формационные технологии\ПОСОБИЕ ПО HTML\Пример Border.html 💌 🎅 Переход Ссылки	»
У первой таблицы атрибут BORDER равен 1 Фамилия Имя Отчество Стаж Иванов Иван Иванович 10 лет У второй таблицы атрибут BORDER равен 21	<
Фамилия Имя Отчество Стаж Иванов Иван Иванович 10 лет	
У третьей таблицы атрибут BORDER равен 0	
Фамилия Имя Отчество Стаж	3
Иванов Иван Иванович 10 лет	~
🕘 Готово 🧕 😼 Мой компьютер	

Рисунок 50 - Пример использования атрибута BORDER

4.2.8 Атрибут BGCOLOR

Данный атрибут позволяет установить цвет фона в таблице. В зависимости от того, с каким тегом (TABLE, TR, TD) он применяется, цвет фона может быть установлен для всей таблицы, для строки или для отдельной ячейки. Значением данного атрибута является RGB-код или символьное название цвета.

Приведем простой пример таблицы, где фон заголовка таблицы окрашен одним цветом, остальные ячейки другим цветом, а рамка третьим цветом.

<HTML>
<HTML>
<HEAD>
<TITLE> Применение цвета в таблицах</TITLE>
</HEAD>
<BODY>
<TABLE border= "2" bordercolor= blue cellpadding=10 cellspacing=10>
<TR><TH BGCOLOR="#CCCCC99">Year</TH>
<TH BGCOLOR="#CCCCC99">Sales</TH>
<TH BGCOLOR= "#CCCCC99">Model</TH>
</TR>
</TR>
</TR>
</TR>
</TR>
</TR>
</TR>

```
<TD BGCOLOR= "#FFFF666">BMV-6</TD>
</TR>
<TR>
<TD BGCOLOR= "#FFFF66">2001</TD>
<TD BGCOLOR= "#FFFF66">$21000</TD>
<TD BGCOLOR= "#FFFF66">Mersedes</TD>
</TR>
<TR>
<TD BGCOLOR= "#FFFF66">$2002</TD>
<TD BGCOLOR= "#FFFF66">$2000</TD>
<TD BGCOLOR= "#FFFF66">Mazda</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

В броузере эта HTML-страница будет выглядеть следующим образом:



Рисунок 51 – Использование цвета в таблицах

Рассмотрим пример HTML-страницы, где каждая ячейка закрашена разным цветом.

```
<HTML>
<HEAD>
<TITLE> Цвета в таблицах</TITLE>
</HEAD>
<BODY>
<CENTER>
<TABLE BORDER= "3" BGCOLOR=yellow>
<TR BGCOLOR=blue>
<TD>Teкст или данные</TD>
```

<td bgcolor="red">Текст или данные</td>	Текст или данные
<td>Текст или данные</td>	Текст или данные
<tr bgcolor="green"></tr>	
<td>Текст или данные</td>	Текст или данные
<td>Текст или данные</td>	Текст или данные
<td bgcolor="lime">Tекст или данные</td>	Tекст или данные

| |
| |
| |
Пример реализации данной HTML-страницы представлен ниже.

🖆 Цве	та в в та	блиц	ax - Microso	oft Intern	net Explo	rer		
Фаил	правка	БИД	изоранное	Сервис	Справка			iii ii
G	Іазад т	0	💌 😰	6	🔎 Поиск	📌 Избранн	oe	**
Адрес:	🖉 посс	БИЕ ПС) HTML\Цвета	в таблица	ix.html 💌	🔁 Переход	Ссылки	»
_								~
Te	жот или	дани	Me Teket 1	или дан	ные Тек	ст или дани	(kie	
Te	кст или	данн	ые Текст і	или дан	ные Тек	ст или дани	ные	
								×
🙆 Гото	во			1	Moй ком	пьютер		

Рисунок 52 – Использование атрибута Bgcolor

4.2.9 Атрибут BACKGROUND

Данный атрибут задает фоновое изображение для таблиц применяется к тегам TABLE и TD. Его значением является URL файла с фоновым изображением. Рассмотрим пример HTML-страницы с использованием атрибута background.

```
<HTML>
<HEAD><TITLE>Пример использования Background</TITLE>
<BODY>
<P></P>
<TABLE BORDER="1" bordercolor=black background= "lips.jpg">
<TR>
<TR>
<TH>Фамилия</TH>
<TH>Фамилия</TH>
<TH>Имя</TH>
<TH>>Имя</TH>
<TH>> Отчество</TH>
```

<ТН>Премия</ТН> </TR> <TR> <TD>Иванов</TD> <TD>Иван </TD> <TD>Иванович</TD> <TD>10 лет</TD> <TD>1000 p.</TD> </TR> <TR> <TD>Петров</TD> <TD>Петр </TD> <TD>Петрович</TD> <TD>5 лет</TD> <TD>500 P.</TD> </TR> <TR> <TD>Сидоров</TD> <TD>Сидор </TD> <ТD>Сидорович</ТD> <TD>12 лет</TD> <TD>1200 p.</TD> </TR> </TABLE> </BODY> </HTML>

Пример реализации данной HTML-страницы в броузере представлен на рисунке 53.

🛓 Пример	использо	вания Backg	round - I	Mi	
Файл Пра	вка Вид	Избранное	Сервис	Справка	
🕝 Назад	• @ •	🖹 💈 (() <i>(</i>)	Поиск	
Адрес: 🛃	Пример Back	ground.html 🔽	🔁 Пер	еход Ссь	лки
-					
Фамил	ия Имя	Отчество	Стаж	Преми	я
Иванов	Иван	Иванович	10 лет	1000 p.	
Петров	Петр	Петрович	5 дет	500 p.	
Сидоро	в Сидор	Сидорови	ч лет	1200 p.	
ПГс		📃 😡 Мой ко	мпьютер		

Рисунок 53 – Пример использования атрибута Background

4.3 Сложные таблицы

4.3.1 Обтекание таблиц

```
Для задания обтекания таблицы текстом используется известный уже
атрибут ALIGN, который теперь применяется не в тегах <TR>, а
непосредственно в теге <TABLE>. Рассмотрим пример обтекания сложной
таблицы текстом справа, тогда атрибут Align будет иметь значение left.
<HTML>
<HEAD>
<TITLE> Таблица с обтекающим текстом </TITLE>
</HEAD>
<BODY>
<TABLE border= "1" bordercolor=black ALIGN=LEFT WIDTH=70%>
<CAPTION>
<H3> Наиболее употребительные мужские имена <BR>
взрослого населения России
</H3>
</CAPTION>
<UL>
<TR>
<TD VALIGN=TOP>
<LI>Александр<LI>Андрей<LI>Анатолий<LI>Аркадий<LI>Борис<LI>В
адим<LI>Валерий<LI>Виталий<LI>Владимир<LI>Владислав<LI>Григо
рий<LI>Дмитрий<LI>Денис
</TD>
<TD VALIGN=TOP>
<LI>Евгений<LI>Иван<LI>Игорь<LI>Илья<LI>Константин<LI>Леони
д<LI>Михаил<LI>Николай<LI> Олег<LI>Роман<LI>Семен <LI>Сергей
<LI>Юрий <LI>Яков
</TD>
</UL>
</TR>
</TABLE>
<BR><BR>
<BR>
<BR><BR>
Приведенные данные получены на основе анализа выборки, содержащей
сведения о 5000 мужчин в возрасте старше 18 лет. Указанные имена
охватывают 92% выборки. Частота встречаемости каждого из
остальных имен не превосходит 0.3%.
</BODY>
</HTML>
    В броузере эта страница будет выглядеть следующим образом:
```


Рисунок 54 – Обтекание таблицы текстом

4.3.2 Вложенность таблиц

Помимо всех рассмотренных свойств таблиц есть еще одно свойство. Это возможность использования вложенных таблиц, когда используется несколько тегов <TABLE>. Рассмотрим пример такой таблицы.

```
<HTML>
<HEAD>
<TITLE> Вложенные таблицы</TITLE>
</HEAD>
<BODY>
<TABLE BORDER= "0" CELLSPACING=0 CELLPADDING=0>
<CAPTION><H3>Города Ленинградской области</H3>
H – население города (тыс. жит., 1992г.)       P –
расстояние от Санкт-Петербурга (км) </CAPTION>
<TR><TD VALIGN=TOP>
<TABLE BORDER="1" BORDERCOLOR=black CELLPADDING=3
CELLSPACING=0>
<CAPTION><STRONG> Города, подчиненные Санкт-Петербургу
</STRONG></CAPTION>
```

```
<TR><TH>Город</TH><TH>H</TH><TH>P</TH></TR>
<TR><TD>Зеленогорск</TD><TD ALIGN=RIGHT>13,6</TD><TD
ALIGN=RIGHT>50</TD></TR>
<TR><TD>Kолпино</TD><TD ALIGN=RIGHT>144,6</TD><TD
ALIGN=RIGHT>26</TD></TR>
<TR><TD>Kpohштaдт</TD><TD ALIGN=RIGHT>45,2</TD><TD
ALIGN=RIGHT>48</TD></TR>
<TR><TD>Ломоносов</TD><TD ALIGN=RIGHT>42</TD><TD
ALIGN=RIGHT>40</TD></TR>
<TR><TD>Павловск</TD><TD ALIGN=RIGHT>25,4</TD><TD
ALIGN=RIGHT>30</TD></TR>
<TR><TD>Петродворец</TD><TD ALIGN=RIGHT>83,8</TD><TD
ALIGN=RIGHT>29</TD></TR>
<TR><TD>Пушкин</TD><TD ALIGN=RIGHT>95,1</TD><TD
ALIGN=RIGHT>50</TD></TR>
<TR><TD>Cестрорецк </TD><TD ALIGN=RIGHT>34,9</TD><TD
ALIGN=RIGHT>35</TD></TR>
</TABLE>
<P>
<CENTER>
Все города, подчиненные администрации Санкт-Петербурга, имеют
прямые городские телефонные номера.
</CENTER>
</TD>
<TD WIDTH=50></TD>
<TD VALIGN=TOP>
<TABLE BORDER= "1" BORDERCOLOR=black CELLPADDING=3
CELLSPACING=0>
<CAPTION><STRONG> Города областного подчинения</CAPTION>
<TR><TH><Fodoa></TH><TH>H</TH><TH>P</TH></TR>
<TR><TD>Бокситогорск</TD><TD ALIGN=RIGHT>21,6</TD><TD
ALIGN=RIGHT>245</TD></TR>
<TR><TD>B0лхов</TD><TD ALIGN=RIGHT>50,3</TD><TD
ALIGN=RIGHT>122</TD></TR>
<TR><TD>Bceboлoжcк</TD><TD ALIGN=RIGHT>32,9</TD><TD
ALIGN=RIGHT>24</TD></TR>
<TR><TD>Bыборг</TD><TD ALIGN=RIGHT>80,9</TD><TD
ALIGN=RIGHT>130</TD></TR>
<TR><TD>Bысоцк</TD><TD ALIGN=RIGHT>1,0</TD><TD
ALIGN=RIGHT>159</TD></TR>
<TR><TD>Гатчина</TD><TD ALIGN=RIGHT>80,9</TD><TD
ALIGN=RIGHT>46</TD></TR>
<TR><TD>Ивангород</TD><TD ALIGN=RIGHT>11,9</TD><TD
ALIGN=RIGHT>147</TD></TR>
```

<TR><TD>Kamehhoropcк</TD><TD ALIGN=RIGHT>80,9</TD><TD ALIGN=RIGHT>157</TD></TR> <TR><TD>Кингисепп</TD><TD ALIGN=RIGHT>51,5</TD><TD ALIGN=RIGHT>138</TD></TR> <TR><TD>Кириши</TD><TD ALIGN=RIGHT>53,8</TD><TD ALIGN=RIGHT>115</TD></TR> <TR><TD>Кировск</TD><TD ALIGN=RIGHT>23,8</TD><TD ALIGN=RIGHT>55</TD></TR> <TR><TD>Линейное поле</TD><TD ALIGN=RIGHT>27,3</TD><TD ALIGN=RIGHT>244</TD></TR> </TABLE> </TD> <TD WIDTH=50></TD> <TD VALIGN=TOP> <TABLE BORDER="1" BORDERCOLOR=black CELLPADDING=3 CELLSPACING=0> <CAPTION>(продолжение таблицы) </CAPTION> <TR><TH>Город></TH><TH>H</TH><TH>P</TH></TR> <TR><TD>Любань</TD><TD ALIGN=RIGHT>4,7</TD><TD ALIGN=RIGHT>85</TD></TR> <TR><TD>Hовая Ладога</TD><TD ALIGN=RIGHT>11,2</TD><TD ALIGN=RIGHT>141</TD></TR> <TR><TD>OTPAдное</TD><TD ALIGN=RIGHT>22,9</TD><TD ALIGN=RIGHT>40</TD></TR> <TR><TD>Пикалево</TD><TD ALIGN=RIGHT>25,1</TD><TD ALIGN=RIGHT>246</TD></TR> <TR><TD>Подпорожье</TD><TD ALIGN=RIGHT>23,1</TD><TD ALIGN=RIGHT>285</TD></TR> <TR><TD>Приморск</TD><TD ALIGN=RIGHT>6,7</TD><TD ALIGN=RIGHT>137</TD></TR> <TR><TD>Приозерск</TD><TD ALIGN=RIGHT>20,5</TD><TD ALIGN=RIGHT>145</TD></TR> <TR><TD>Cветогорск</TD><TD ALIGN=RIGHT>15,8</TD><TD ALIGN=RIGHT>201</TD></TR> <TR><TD>Cланцы</TD><TD ALIGN=RIGHT>42,6</TD><TD ALIGN=RIGHT>192</TD></TR> <TR><TD>Cосновый бор</TD><TD ALIGN=RIGHT>57,6</TD><TD ALIGN=RIGHT>81</TD></TR> <TR><TD>Tихвин</TD><TD ALIGN=RIGHT>72,0</TD><TD ALIGN=RIGHT>200</TD></TR> </TABLE> </TD> </TR></TABLE> </BODY> </HTML>

	пэац	ri ri	данно		pam	щы	пред		I/KC.		
🖀 Вложенные таблицы - Microsoft Internet Explorer 📃 🔲 🔀											
Файл Правка Вид Избранное Сервис Справка 🥂											
🕞 Назад 🔹 📀 т 💽 🛃 😭 🔎 Поиск 👷 Избранное 🤣 😒 т 🌄 🦓											
Адрес: 餐 тевые и	нформац	ионны	ые технолог	ИИ/ПОСОБИЕ ПО НТИ	ЧL\Вло>	кенные	е таблицы	.html 🔽 🔁 Пер	еход	Ссылк	ки »
			тород	а ленинградо	скои	001	асти				
Н — населен	ие гор	ода	(тыс. жиз	r., 1992r.) 🛛 P	-pac	стоян	ние от С	анкт-Петербу	рга (к	:м)	
Города, подч	иненн	ые		Города обла	стно	0		(продолж	кение	•	
Санкт-Пет	ербург	у	· •	подчине	ния			таблиц	цы) 		
Город	Н	Р		<Город>	Н	Р		Город>	Н	Р	
Зеленогорск	13,6	50		Бокситогорск	21,6	245		Любань	4,7	85	
Колпино	144,6	26		Волхов	50,3	122		Новая	11.2	141	
Кронштадт	45,2	48		Всеволожск	32,9	24		Ладога	,-		
Ломоносов	42	40		Выборг	80,9	130		Отрадное	22,9	40	≡
Павловск	25,4	30		Высоцк	1,0	159		Пикалево	25,1	246	
Петродворец	83,8	29		Гатчина	80,9	46		Подпорожье	23,1	285	
Пушкин	95,1	50		Ивангород	11,9	147		Приморск	6,7	137	
Сестрорецк	34,9	35		Каменногорск	80,9	157		Приозерск	20,5	145	
				Кингисепп	51.5	138		Светогорск	15,8	201	
Все города, подчиненные			Кириши	53.8	115		Сланцы	42,6	192		
администрации Санкт- Петербурга, имеют		Кировск	23,8	55		Сосновый бор	57,6	81			
прямые гор телефонные	одски номер	e Da.		Линейное поле	27,3	244		тихвин	72,0	200	
ј 🕘 Готово			•					Мой компьютер			

* IITMI

Рисунок 55 - Вложенные таблицы

4.3.3 Структурированные таблицы

В HTML имеется возможность структурировать таблицы и гибко управлять прорисовкой рамок и линий сетки. Для этого используются теги <THEAD>, <TBODY>, <TFOOT>.

Теги <THEAD>, <TBODY> и <TFOOT> более строго задают структуру описания таблицы, выделяя ячейки заголовка таблицы, основное содержимое таблицы и итоговую строку. Эти теги могут встречаться только в описании таблиц внутри пары <TABLE> и </TABLE>.

Теги <THEAD> и <TFOOT> используются для описания верхнего и нижнего колонтитулов таблицы. Эти теги могут встречаться в таблице не раза. Завершающий тег более одного для них можно опускать. Использование данных тегов удобно при создании больших таблиц, выходящих за пределы одной страницы.

Тег <TBODY> может встречаться многократно в описании таблицы, при этом требуется использование завершающего тега </TBODY>.Этот тег группирование данных выполняет логическое так же, как И тег <COLGROUP>, выполняющий группирование смежных столбцов.

При использовании новых тегов появляется возможность более гибко управлять рамками и линиями сетки таблицы.

Управление прорисовкой рамок вокруг таблицы осуществляется параметром FRAME тега <TABLE>, а линии сетки таблицы – параметром – RULES. Например, становится возможным провести только вертикальные линии между колонками и вместо рамки вокруг всей таблицы дать горизонтальные линии сверху и снизу таблицы.

Параметр FRAME может принимать следующие значения:

- BOX или BORDER рамка рисуется со всех четырех сторон;
- ABOVE рамка рисуется только с верхней стороны;
- BELOW рамка рисуется только с нижней стороны;
- HSIDES рисуется и нижняя и верхняя сторона;
- VSIDES рисуется левая и правая сторона;
- LHS только с левой стороны;
- RHS –только с правой стороны;
- VOID таблица без внешних рамок.

Параметр RULES управляет прорисовкой внутренних линий сетки таблицы и может принимать следующие значения:

- ALL рисуются все внутренние линии;
- GROUPS рисуются только линии, разделяющие группы;
- ROWS рисуются линии, разделяющие строки;
- COLS рисуются линии, разделяющие столбцы;

- NONE – внутренние линии не рисуются.

Пример: <TABLE BORDER FRAME=HSIDES RULES=GROUPS>

Причем прорисовка линий сетки таблицы и рамок будет осуществляться при наличии параметра BORDER. При отсутствии этого параметра или при нулевом значении линии сетки и рамки будут отсутствовать /8/.

Приведем пример полного HTML-кода, создающего таблицу с использованием описанных возможностей.

<HTML>

<HEAD>

<TITLE> Выделение заголовка и итоговой строки</TITLE>

</HEAD>

<BODY>

<TABLE BORDER=5 bordercolor=black WIDTH=100% RULES=GROUP FRAME=HSIDES>

```
<COLGROUP ALIGN=CENTER>
```

<COLGROUP ALIGN=CENTER>

<COLGROUP ALIGN=CENTER>

<CAPTION>

<H3>

Пример гибкого управления линиями сетки таблицы</Н3>

</CAPTION>

<THEAD>

```
<TR>
<TH>Заголовок столбца 1 </TH>
<TH> Заголовок столбиа 2</TH>
<TH> Заголовок столбца 3 </TH>
</TR>
</THEAD>
<TBODY>
<TR><TD>Данные</TD><TD>Данные</TD></TR>
<TR><TD>Данные</TD><TD>Данные</TD></TR>
<TR><TD>Данные</TD><TD>Данные</TD></TR>
<TR><TD>Данные</TD><TD>Данные</TD></TR>
<TR><TD>Данные</TD><TD>Данные</TD></TR>
<TR><TD>Ланные</TD><TD>Ланные</TD></TR>
</TBODY>
<TFOOT>
<TR><TD>Итог</TD><TD>Итог</TD>Итог
</TD>
</TR>
</TFOOT>
</TABLE>
</BODY>
</HTML>
    Реализация данного HTML-кода представлена на рисунке 56.
         🚰 Выделение заголовка и итоговой строки - Microsoft Internet E... 💶
                  Вид Избранное
                             Сервис
                                  Справка
         Файл Правка
            Назад
                       ×
                                  🕽 Поиск 🛛 🤫 Избранное
         Адрес: 🥔 О HTML\Выделение заголовка и итоговой стоки.html 🗙
                                            📄 Переход
                                                    Ссылки
          Пример гибкого управления линиями сетки таблицы
          Заголовок столбца 1 Заголовок столбца 2 Заголовок столбца 3
              Данные
                              Данные
                                             Данные
               Итог
                               Итог
                                              Итог
```

Рисунок 56 – Пример гибкого управления линиями сетки таблицы

🚽 Мой компьютер

🗐 Готово

5 Гиперссылки

Гипертекстовые связи (ссылки) являются наиболее важным элементом Web-страниц. С их помощью делают документ связанным и структурированным, что позволяет пользователю получать необходимую ему информацию максимально быстро и удобно.

Ссылки могут указывать на другой документ, специальное место данного документа или выполнять другие функции, например, запрашивать файл по FTP-протоколу для отображения его броузером.

Ссылка состоит из двух частей. Первая из них - это то, что видно на Web-странице, она называется указатель ссылки (anchor). Вторая часть, дающая инструкцию броузеру, называется адресной частью ссылки (URL-адрес). <А HREF = "URL – адрес">Текстовый указатель ссылки </А>

Когда происходит щелчок мышью по указателю ссылки, броузер загружает документ, адрес которого задается URL-адресом. URL (Universal Resource Locator) - универсальный способ адресации ресурсов в сети может указывать на специальное место по абсолютному пути доступа, или указывать на документ в текущем пути доступа, что часто используется при организации больших структурированных Web-сайтов. В зависимости от того, какой используется URL-адрес ссылки могут быть относительными и абсолютными.

5.1 Относительные гиперссылки

В относительных гиперссылках используется в качестве URL-адреса относительный адрес ресурса. Сюда относятся файлы с того же компьютера или с той же папки, что и файл источник.

<А HREF="1.HTML">Относительная ссылка</А>

В приведенном примере при нажатии на указатель "Относительная ссылка" осуществится связь с документом 1.HTML, находящимся в одной папке с файлом-источником. Если файл находится в другой папке, необходимо указать путь до файла, начиная с диска С:\Мои документы\Webдизайн\1.html.

Внутренние гиперссылки (закладки). Для построения внутренней ссылки сначала нужно создать указатель, определяющий место назначения. Например, если необходимо сделать ссылку на текст определенной главы документа, нужно разместить там указатель и дать ему имя при помощи атрибута NAME тега $\langle A \rangle$. При этом параметр HREF не используется, и броузер не выделяет содержимое тега $\langle A \rangle$.

После того как место назначения определено, можно приступить к созданию ссылки на него. Для этого, вместо указания в параметре HREF URL-адреса документа, как это делалось ранее, поместим туда имя ссылки с префиксом #, говорящим о том, что это внутренняя ссылка.

Глава 5

Теперь, если пользователь щелкнет кнопкой мыши на указателе "Глава 5", броузер выведет соответствующую часть документа в окне просмотра.

Ссылки на электронную почту. Создание ссылки на электронную почту так же просто, как и на другую страницу. Для этого вместо URL-адреса следует указать адрес электронной почты, предварив его словом mailto:.

<А HREF="mailto:LG-77@mail.ru">Присылайте ваши отзывы Необходимо отметить, что можно использовать все типы относительных гиперссылок в одном HTML-документе, как ссылки на файл с диска, закладки, ссылки на почту. Рассмотрим пример HTML_страницы с использованием различных типов относительных гиперссылок.

<HTML>

<HEAD><TITLE> Примеры относительных гиперссылок </TITLE> </HEAD>

<BODY><H2>Примеры относительных ссылок: </H2><HR> Ссылка на файл в той же папке:

 Связаться с файлом index.html
 Ссылка на файл в другой папке:

 Связаться с файлом more.html
Ссылка на фрагмент в документе, с использованием относительного URL:

Связаться с пятой главой
<H2>Ссылка на почту: </H2>

<HR>
 Пишите ваши отклики

</BODY>

</HTML>

В броузере это будет выглядеть следующим образом:



Рисунок 57 – Примеры относительных ссылок

5.2 Абсолютные гиперссылки

Гиперссылки могут быть и более сложными. В абсолютных ссылках в качестве URL использует абсолютный адрес ресурса, начиная с протокола, по которому к нему обращаются.

<А HREF="http://www.mysite.ru">Абсолютная ссылка</А>

Можно указывать адрес сервера или непосредственно путь до файла, расположенного на удаленном сервере.

 Абсолютная ссылка

Точно также как и относительные можно использовать все типы абсолютных гиперссылок в одном HTML-документе. Различные варианты абсолютных гиперссылок приводится на следующей HTML-странице: **<HTML>**

<HEAD>

<TITLE> Примеры абсолютных гиперссылок</TITLE>

</HEAD>

<BODY>

```
<H2>Примеры абсолютных ссылок:
```

</H2>

<HR>

Ссылка с базовым URL протокола HTTP на домашнюю страницу Webузла:
 Посетите Rambler

Добавлено название каталога. URL ссылается на документ в данном каталоге, который выдается сервером по умолчанию:


```
<A HREF="http://www.microsoft.com/gallery/"> Коллекция Web-
элементов Microsoft</A> <BR>
```

Добавлено имя файла. URL полностью описывает местоположение документа:

 Данные Томаса

Добавлена ссылка на фрагмент в документе. URL ссылается на конкретное место в документе:

<**BR**>

 Первая глава

Использование URL не только протокола HTTP, но и FTP:


```
<A HREF="ftp://ftp.cdrom.com/"> Доступ к FTP-архиву </A>
<BR>
```

</BODY>

</HTML>

В броузере эта HTML-страница будет выглядеть следующим образом:



Рисунок 58 – Примеры абсолютных ссылок

Помимо рассмотренных у элемента <A> могут быть еще множество других атрибутов. Большинство из них представлены в таблице 6.

Tuominga O Da	sobble arphoyrbi 55	
Имя	Возможные	Описание
атрибута	значения	
HREF	URL	URL объект, на который делается ссылка
NAME	Текст	Присваивает якорю имя, чтобы на него
		мог сослаться другой якорь
ID	Текст	Присваивает якорю идентификатор,
		чтобы на него мог сослаться другой
		якорь, списки стилей или сценарий
TARGET	Название кадра	Определяет, в каком кадре будет
	_	открываться указанный объект
TITLE	Текст	Задает текст подписи для ссылок
ACCESSKEY	Символ	Задает клавишу на клавиатуре, по
		которой будет срабатывать ссылка
TABINDEX	Цифровое	Задает порядок, по которому будут
	значение	обходиться ссылки на странице при
		нажатии на клавишу ТАВ
REL	Текст	Задает отношение между объектом, на
		который делается ссылка и текущим
		объектом
REV	Текст	Задает отношение между текущим

Таблица 6 – Базовые атрибуты элемента <A>

	объектом	И	объектом,	на	который
	делается сс	ылк	a		

Что касается цвета гиперссылок, то их можно менять. Как известно, неактивные гиперссылки в броузере отображаются синим цветом, активные, т.е. те, на которые пользователь щелкнул мышкой отображаются карсны, а посещенные ссылки отображаются фиолетовыми.

Для того, чтобы переопределить цвета гиперссылок необходимо в теге <BODY> задать следующие атрибуты:

- LINK (цвет, который будет использоваться при выводе на экран еще не выбранных гипертекстовых связей),
- VLINK (цвет, который будет использоваться при выводе на экран уже проверенных гипертекстовых связей),
- ALINK (цвет, которым будут выделяться гипертекстовые связки в тот момент, когда пользователь щелкает по ним клавишей мыши).

Рассмотрим пример изменения цвета гиперссылок:

<HTML>

<HEAD>

<TITLE>Использование цвета в гиперссылках</TITLE> </HEAD><BODY BGCOLOR=#bluesky TEXT=SILVER LINK=BLUE VLINK=MAROON ALINK=RED >

```
<H2><A HREF= "1.html">ГИПЕРРСЫЛКА</А>
```

```
<A HREF= "2.html">ГИПЕРССЫЛКА</А>
```

```
<A HREF= "3.html">ГИПЕРССЫЛКА</А>
```

</H2>

```
</BODY>
```

</HTML>

Реализация данной HTML-страницы представлена на рисунке 59.



Рисунок 59 – Пример изменения цвета в гиперссылках

На приведенном рисунке рассмотрен пример трех гиперссылок, причем первая и третья уже посещенные, а вторая активная в данный момент. Как видно из рисунка первая и третья стали темно-красными, а вторая гиперссылка поменяла цвет и стала красной.

6 Графика и мультимедиа в НТМL

6.1 Графика

Для того чтобы вставить в Web-страницу изображение, необходимо воспользоваться тегом **** с обязательным атрибутом SRC (SouRCe, источник). Имя файла представляет собой имя выводимого графического файла. Замыкающего тега не требуется.

Рассмотрим подробнее атрибуты тега .

Атрибут SRC. Указывает файл изображения и путь к нему; изображение должно быть загружено в броузер и размещено в том месте документа, где расположен тег изображения.

Атрибут ALT. Позволяет указать текст, который будет выводиться вместо изображения броузерами, неспособными представлять графику.

Атрибут ALIGN. Определяет положение изображения относительно окружающего его текста и имеет атрибуты, рассмотренные ниже.

ALIGN= "left" определяет огибаемое текстом изображение. Изображение располагается вдоль левой границы документа, а последующие строки текста огибают его справа.

ALIGN="right" определяет огибаемое текстом изображение. Изображение располагается вдоль правой границы документа, а последующие строки текста огибают его слева.

ALIGN= "top" выравнивает верх изображения по верхнему краю самого высокого элемента в строке окружающего текста точно так же, как при использовании стандартного набора атрибутов.

ALIGN="textop" выравнивает верх изображения по верхнему краю самого высокого текстового символа в строке окружающего текста. Действие этого аргумента в большинстве случаев, но не всегда, подобно действию аргумента ALIGN= "top".

ALIGN="middle" выравнивает центр изображения по базисной линии строки окружающего текста точно так же, как при использовании стандартного набора атрибутов.

ALIGN="absmiddle" выравнивает центр изображения по центру строки окружающего текста.

ALIGN= "baseline" выравнивает нижний край изображения по базисной линии строки окружающего текста, то есть производит такое же действие, как и ALIGN= "bottom".

ALIGN= "bottom" выравнивает нижний край изображения по базисной линии строки окружающего текста точно так же, как при использовании стандартного набора атрибутов.

ALIGN="absbottom" выравнивает нижний край изображения по нижнему краю строки окружающего текста.

Атрибут USEMAP. Если присутствуют атрибут USEMAP и тег </br>
<MAP>, изображение становится чувствительной картой, или "картой изображения". Если щелкнуть кнопкой мыши на активной области изображения, для которого определен атрибут USEMAP, произойдет гипертекстовый переход к информационному ресурсу, установленному для этой области.

Атрибут BORDER. Целочисленное значение аргумента определяет толщину рамки вокруг изображения. Если значение равно нулю, рамка отсутствует.

Атрибут HSPACE и *VSPACE*. Целочисленное значение этих атрибутов задает соответственно горизонтальное и вертикальное расстояние между изображением и огибающим его текстом.

ISMAP сообщает броузеру, что данное изображение .является серверной графической картой.

Атрибуты WIDTH и HEIGHT. Оба атрибута задают целочисленные значения размеров изображения по горизонтали и по вертикали соответственно /8/.

При использования тега только атрибут SRC является обязательным. На рисунке представлен пример использования графики без использования дополнительный атрибутов.

<HTML>

<HEAD>

<TITLE>

Пример простого рисунка</TITLE>

<BODY>

</BODY>

</HTML>

Реализация этой HTML-страницы в броузере представлена на рисунке 60.



Рисунок 60 – Пример добавления простого рисунка

Теперь в данный тег можно добавлять различные атрибуты, т.е. менять размер, положение изображения, добавлять рамку.

Ниже приведен HTML-код страницы с рисунком с добавленными атрибутами, задана ширина 300, высота 200, отступ сверху и снизу 10, рамка, выравнивание по правой стороне.

<HTML> <HEAD> <TITLE> Пример рисунка с атрибутами</TITLE> <BODY>

</BODY>

</HTML>

В броузере это будет выглядеть следующим образом.



Рисунок 61 – Пример использования рисунка с атрибутами

В рисунках, также как и других элементах HTML-страниц используется выравнивание внутри окна броузера. Рассмотрим пример использования атрибута Align.

<HTML>

<HEAD>

<ТІТLЕ> Примеры выравнивания рисунков</ТІТLЕ>

</HEAD>

<BODY><P> Этот текст должен быть выровнен по верхнему краю картинки</P> <P> Этот текст должен быть выровнен по середине картинки</P>

<P> Этот текст должен быть выровнен по нижнему краю картинки</P> </BODY>

</BOD1></HTML>

Реализация этой HTML-страницы в броузере представлена на рисунке 62.



Рисунок 62 - Пример выравнивания текста относительно рисунка

Помимо этого атрибут Align может быть использован для создания обтекания текста вокруг рисунка. Рассмотрим пример двух картинок у первой значение атрибута равно LEFT, а второго RIGHT.

<HTML> <HEAD>

<TITLE> Пример обтекания рисунков</TITLE>

</HEAD>

<BODY>

У верхнего рисунка атрибут ALIGN равен LEFT, поэтому текст обтекает его справа. У верхнего рисунка атрибут ALIGN равен LEFT, поэтому текст обтекает его справа. У верхнего рисунка атрибут ALIGN равен LEFT, поэтому текст обтекает его справа. У верхнего рисунка атрибут ALIGN равен LEFT, поэтому текст обтекает его справа. У верхнего рисунка атрибут ALIGN равен LEFT, поэтому текст обтекает его справа.

<P ALIGN= "Right"> У нижнего рисунка атрибут ALIGN равен RIGHT, поэтому текст обтекает его слева. У нижнего рисунка атрибут ALIGN равен RIGHT, поэтому текст обтекает его слева. У нижнего рисунка атрибут ALIGN равен RIGHT, поэтому текст обтекает его слева. У нижнего рисунка атрибут ALIGN равен RIGHT, поэтому текст обтекает его слева. </P>

</HTML>

В броузере эта HTML-страница будет выглядеть следующим образом:



Рисунок 63 – Пример обтекания рисунков

Помимо этого, изображения на Web-странице могут использоваться в качестве гипертекстовых ссылок, как и обычный текст. Для обозначения изображения как гипертекстовой метки используется тот же тег <A>, что и для текста, но между <A> и вставляется тег изображения .

Когда якорь окружает изображение, оно превращается в ссылку. С его помощью можно организовать связь с другими документами или рисунками. Таким образом, рисунки могут превращаться в графическую кнопку.

<HTML>

<HEAD><TITLE> Примеры ссылок на рисунки</TITLE></HEAD> <BODY>

<H3>Использование рисунков в качестве гиперссылок:</H3>

1) Использование рисунка в качестве гиперссылки на htmlдокумент:

2) Использование простой гиперссылки на рисунок:

 Ссылка на рисунок

3) Использование рисунка в качестве гиперссылки на другой рисунок:

 </BODY>

</HTML>

В броузере эта HTML-страница будет выглядеть следующим образом:



Рисунок 64 – Использование рисунков в качестве гиперссылок

Как видно из рисунка 64, рисунки-гиперссылки могут быть различных типов. Так в первом случае представлен вариант использования рисунка в качестве ссылки на HTML-документ, во втором представлен вариант ссылки на графический файл, в третьем случае представлена ссылка на графический файл при нажатии рисунка-гиперссылки.

Обычно если изображение является составной частью якоря, броузер обводит его цветной рамкой, цвет рамки совпадает с цветом текстовых ссылок — либо синий либо лиловый. Таким образом создаются анимированные кнопки. На рисунке 64 представлено, как якорь связывается с элементом , при этом, что атрибуту BORDER присвоено значение "0", для того чтобы отключить вывод рамки.

```
<HTML>
<HEAD>
<TITLE>Pисунки и рамки
</TITLE>
</HEAD><BODY bgcolor="DDAADD"><BR>
<B>KHOПКа с рамкой</B>
<A HREF= "most.html">
<IMG SRC= "most.jpg" ALT= "Кнопка" BORDER= "12" HEIGHT= "70"
WIDTH= "100">
</A><BR> <BR> <BR>
<B>KHOПКа без рамки</B>
```

```
<A HREF= "most-2.html">
<IMG SRC= "most-2.jpg" ALT= "Кнопка" BORDER= "0" HEIGHT= "80"
WIDTH= "110">
</A>
</BODY>
</HTML>
```

В броузере это будет выглядеть следующим образом:



Рисунок 65 – Пример использования рисунка в качестве кнопки

6.2 Мультимедиа

Кроме графических изображений, на Web-страницу можно интегрировать музыкальные файлы различных форматов, видео, фоновый звук, Flash-ролики. Для этого в HTML используются различные теги, такие как <OBJECT>, <EMBED>, <BGSOUND> и др. Рассмотрим подробнее каждый из этих элементов.

6.2.1 Элемент ОВЈЕСТ

Элемент <OBJECT> позволяет встроить в HTML-документ любой объект (Flash-ролик, видео, Java-аплет) заданием URL встраиваемого объекта и инициализировать его (задать начальные значения необходимым параметрам). В таблице 7 представлены основные атрибуты тега <OBJECT>.

Атрибут	Описание		Значение	Пример
Data	Задает	URL	URL	Data="movie.swf"
	встраиваемого			

Таблица 7 – Атрибуты элемента < OBJECT>

	объекта		
Codebase	Содержит URL того приложения, которое необходимо для просмотра встраиваемого объекта	URL	Codebase="http:=//download.ma cromedia.com/pub/shockwave/ca bs/flash/swflash.cab#version=5,0 ,0,0"
Codetype	Содержит МІМЕ-	MIME-	Codetype="application/x-
	тип объекта, т.е.	ТИП	shokwave-flash"
	определяет, что это		
	за объект:		
	видеоролик, flash-		
	ролик, аплет и др.		
Classid	Содержит URL	URI	Classid="clsid:D27CDB6E-
	описание типа		AE6D-11ct-96B8-
	встраиваемого		444555540000
	ооъекта или может		
	служить заменои		
Standby	Содержит текст	текст	Standby="Object_225pyw2etcg
Standby	который может	ICKCI	пожацуйста положлите"
	отображаться		nomanynera, nodomdnie
	броузером пока		
	загружается объект		
Width,	Высота, ширина,	значени	Width="250" height="200"
height,	толщина рамки	e	border="0"
border	вокруг объекта		

Рассмотрим работу тега <OBJECT> на конкретном примере. Необходимо вставить объект movie.swf. Если в броузере пользователя установлен необходимый плейер, то броузер определит его наличие по специальному идентификатору, который указывает в качестве значения атрибута classid /9/.

Для flash-плейера этот атрибут задается следующим образом: Classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000". Если такой плейер не установлен, то броузер может его автоматически скачать с сайта производителя, для этого надо указать значение атрибута codebase, в него помещается URL, которому его можно скачать: Codebase= по "http:=//download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#versio n=5,0,0,0".

Атрибут classid может использоваться вместе с атрибутом data или как альтернатива ему, в зависимости от типа объекта.

В самом элементе <OBJECT> требуется также задать MIME-тип встраиваемого объекта: Codetype="application/x-shokwave-flash". Атрибут standby задает альтернативный текст, который будет загружать в момент

загрузки приложения. Также можно задать стандартные параметры ширины, высоты и объекта.

<OBJECT Classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" Codebase= "http:=//download.macromedia.com/ pub/shockwave/cabs/flash/ swflash.cab#version=5,0,0,0" width= "250" height= "200" Codetype= "application/x-shokwave-flash" Standby=="Oбъект загружается, пожалуйста, подождите">

....Задание параметров для flash-плейера </OBJECT>

6.2.2 Элемент ЕМВЕD

Элемент <EMBED> аналог элемента <OBJECT>, его можно использовать для вставки в HTML-документ музыки и видео.

Синтаксис тега:

<EMBED src= "clip.avi">

Существует и другой способ встраивания звуковых и видео файлов на HTML-страницу. Можно просто сделать ссылку на файл с помощью элемента Anchor. Например:

<А HREF= "Myvideo.avi">Мой видео файл</А>

А если файл находится не на разрабатываемой странице, то нужно прописать абсолютный URL файла.

Мой видео файл

Тег <ЕМВЕD> обладает значительным количеством параметров, задавая которые можно контролировать воспроизведение файлов. Атрибуты тега <ЕМВЕD> представлены в таблице 8.

Атрибут	Описание	Значение	Пример
1	2	3	4
Src	Задает URL объекта, который	URI	src = "clip.avi"
	должен быть встроен в		
	документ		
Alt	Задает текст, который будет	текст	alt="Свадебная
	выводиться на месте		фотография"
	изображения в том случае,		
	если отображение рисунков		
	отключено		
Width,	Задают высоту и ширину	значение в	Width="250"
height	объекта	рх или %	height="200"

Таблица 8 – Атрибуты элемента EMBED

Продолжение таблицы 8

1	2	3	4
Border	Устанавливает толщину рамки вокруг объекта	значение	Border= "0"
Vspace,	Указывают значение верхнего,	значение в	Vspace= "10"

hspace	нижнего, правого и левого полей объекта	рх или %	hspace= "5"
Align	Определяет способ выравнивания объекта на странице	top, middle, bottom, left, right	Align="right"
Pluginspage	URL Web-страницы, содержащей инструкции по инсталляции модуля	URL	Pluginspage= "URL"
Туре	МІМЕ-тип встроенного объекта, определяющий модуль, который необходимо загрузить. Используется для модулей, которым не требуются данные	MIME-тип	Type= "application/x- shopckwave- flash"
Autostart	Указывает, надо ли сразу проигрывать загружаемый файл	true (надо), false (не надо)	Autostart= "true"
Strarttime	Воспроизведение с указанного в минутах и секундах момента от начала файла. По умолчанию 00:00	"mm:ss"	Strarttime= "00:00"
Volume	Громкость воспроизведения, указанная в процентах от максимальной	%	Volume= "20%"
Controls	Задает набор органов управления консоли.	console, smallconsole, playbutton, pausebutton, stopbutton, volumelever.	Controls= "console"

Вместе с тегом <EMBED> может использоваться тег <NOEMBED> на случай, если броузер пользователя не поддерживает встраиваемые объекты. Синтаксис тега:

<NOEMBED> Ваш броузер не поддерживает встраиваемые объекты </NOEMBED></EMBED>

Рассмотрим HTML-документ, в котором используется тег <EMBED>, а также есть ссылка на звуковой файл.

<HTML>

<HEAD><TITLE>Пример использования звуковых файлов </TITLE> </HEAD>

<ВОДУ>Встраивание звукового файла

с <U>полным</U> набором органов управления консоли <EMBED SRC= "example1.wav" HEIGHT=60 WIDTH=144 AUTOSTART=TRUE>

Встраивание звукового файла с <U>ограниченным</U> набором органов управления консоли <EMBED SRC= "example1.mid" HEIGHT=15 WIDTH=144 CONTROLS=SMALLCONSOLE> Это ссылка на звуковой файл </BODY>

</HTML>

Реализация этой страницы представлена на рисунке 66.



Рисунок 66 - Пример использования звуковых файлов

Приведем другой пример. Используется HTML-код, который добавляет на страницу Flash-ролик с использованием элемента EMBED.

<EMBED src= file://C:/HTML-tutorital/movie.swf quality="high"
bgcolor= "#000080" width= "250" height= "200" type="application/xshockwave-flash" plaginspage= "http: // www.macromedia.com/ shockwave/
download/ index.cgi? P1_Prod_Version=Shockwaveflash"> </EMBED>

6.2.3 Элемент BGSOUND

Элемент BGSOUND задает фоновый звук на Web-страницу. Он будет проигрываться во время просмотра пользователем страницы. В HTML поддерживаются простейшие аудиоформаты: WAV, AU, и MIDI. Звук добавляется при помощи тега BGSOUND, который помещается в тело документа, в элемент BODY.

<BGSOUND SRC= "URL" LOOP= "n" BALANCE= "m" VOLUME= "k"></BGSOUND>

Атрибут SRC по аналогии с одноименным атрибутом элемента IMG задает URL того объекта, который нужно встроить в страницу, т.е. звуковой

файл. Атрибут LOOP указывает число повторений этого музыкального файла, при задании LOOP= "-1" файл будет проигрываться бесконечно. Стереобаланс можно регулировать атрибутом BALANCE, громкость проигрывания можно регулировать атрибутом VOLUME.

Пример использования тега <Bgsound>:

<BGSOUND SRC="sound.midi" LOOP="10"></BGSOUND> - задается десятикратное повторение воспроизведения звукового файла sound.midi.

6.3 МІМЕ-типы

Когда страница передается из сети Internet, то броузер сначала анализирует ее МІМЕ-тип, чтобы определить тип приложения, которое ему потребуется. Если это простой текст или рисунок, броузер может сделать это без посторонней помощи. Если объект незнакомый, то броузер может прибегнуть к помощи внешней программы или встроенного в броузер программного модуля (плагина), которые отобразят этот объект.

MIME (Multipurpose Internet Mail Extension, многоцелевые расширения почты Internet) – это стандарт, разработанный для передачи электронной почты. В таблице представлены наиболее популярные и наиболее используемые МІМЕ-типы.

Тип файла	Расши	МІМЕ-тип
i ini quiniu	рение	
Anyur r donyate zin	× zin	application/y_zin_
Архив в формате гір	.zıp	application/x-zip-
	•	compressed, application/x-zip
Аудиозапись АU	*.au	audio/basic
Аудиозапись MIDI	*.midi	audio/mid
Аудиозапись МРЗ	*.mp3	audio/mpeg
Аудиозапись WAV	*.wav	audio/wav
Видеоклип в формате	*.avi	video/x-msvideo
AVI		
Видеоклип в формате	*.mpeg	video/mpeg
MPEG		
Документ в формате	*.html	text/html
HTML		
Документ в формате	*.xls	application/x-msexcel
MS Excel		
Документ в формате	*.doc	application/msword
MS Word		
Документ в формате	*.pdf	application/pdf
pdf	-	
Таблица стилей CSS	*.css	text/css
Flash-ролик	*.swf	application/x-
_		shockwave-flash

Таблица 9 - Наиболее используемые МІМЕ-типы

MS Jscript	*.js	text/javascript
MS VBScript	*.vbs	text/vbscript

7 Карты изображений

Очень мощным средством организации гипертекстовых ссылок с использованием графики является так называемые "карты изображения" (imagemaps). Они представляют собой изображения с чувствительными областями, при нажатии каждой из этих областей происходит связь с различными документами. Карты изображения бывают двух типов: обрабатываемые на сервере (server-side) или на клиенте (client-side).

Для создания карт на стороне клиента используется тег HTML под названием <МАР>. Синтаксис тега таков:

<MAP NAME= "имя карты">

<AREA параметры>

... <AREA параметры>

</MAP>

При этом параметры тега <AREA> могут быть следующими:

SHAPE - определяет форму активной области для карты. Shape может принимать значения RECT, CIRCLE, POLY, DEFAULT, соответствующие областям прямоугольной, круглой, многоугольной и области. Все типы областей, кроме DEFAULT, требуют указания координат в атрибуте COORDS.

COORDS - определяет координаты заданных областей. Для прямоугольной области определяются значения координат верхнего левого и правого нижнего углов ("x1,y1,x2,y2"), для круговой - координаты центра и радиус ("x, y, r"), для многоугольной - список координат всех вершин ("x1, y1, x2, y2, x3, y3,... xn, yn").

HREF - определяет URL, на который ссылается данная область.

NOREF - указывает, что область "мертва", то есть за ней не скрывается ссылка.

Обработка графических карт у клиента. Для создания графической карты, обрабатываемой на клиентской машине, необходимо добавить атрибут USEMAP в элемент , и присвоить ему ссылку на элемент <MAP>. В этом элементе задаются активные области графической карты. Например, элемент может выглядеть так: . В отличие от своего серверного собрата, данная графическая карта будет отображена в броузере как ссылка, несмотря на отсутствие элемента <A>, окружающего .

Рассмотрим пример использования клиентской графической карты. В данном примере заданы три области - прямоугольник, круг и многоугольник. При нажатии на каждую из областей происходит гипертекстовая связь с соответствующим документом с помощью атрибута HREF.

<HTML>

<HEAD>

```
<TITLE> Пример карты изображения
</TITLE>
</HEAD>
<BODY>
<H1>Графическая карта</H1>
<DIV ALIGN="CENTER">
<IMG SRC="bigapple.jpg" USEMAP= "#cat" BORDER= "1" WIDTH=
"280" HEIGHT="200">
</DIV> </MAP NAME= "cat">
<AREA SHAPE= "RECT" COORD= "6, 50, 140, 143" HREF=</p>
"rectangle.html" ALT= "Прямоугольник">
<AREA SHAPE= "CIRCLE" COORDS= "195, 100, 50" HREF=</pre>
"circle.html" ALT= "Круг">
<AREA SHAPE= "POLY" COORDS= "255, 122, 306, 53, 334, 62, 338, 0, 338,</p>
77, 374, 116, 323, 17, 255, 122" HREF= "polygon.html" ALT=
"Многоугольник">
</MAP>
</BODY>
</HTML>
```

В броузере эта HTML-страница будет выглядеть следующим образом:



Рисунок 67 – Пример графической карты, обрабатываемой у клиента

Помимо этого в картах изображения можно использовать атрибут NOHREF. Польза от атрибута NOHREF, казалось бы, небольшая, но с его помощью можно определять области, за которыми не закрепляются никакие действия. Например, можно создать глаз в виде большой графической карты. Область внутри глаза – там, где расположены радужная оболочка и зрачок, – не является активной, и щелчки по ней ни к чему не приводят, а остальная часть глаза, - веко и бровь являются активной. С помощью атрибута NOHREF сделать это очень просто. Определить большую область для всего изображения, а затем внутри его с помощью атрибута NOHREF определить небольшую область, которая не будет реагировать на щелчки мышью. Приведем пример HTML-кода с использованием атрибута NOHREF.

<HTML><HEAD>

```
<ТІТLЕ>Пример использования NOHREF
```

</TITLE>

</HEAD>

<BODY>

<MAP NAME= "map">

<h3>Пример использования атрибута NOHREF</H3> <AREA SHAPE= "circle" COORDS="100, 80, 30" ALT="Heaктивная область" NOHREF> <AREA SHAPE= "circle" COORDS= "100, 80, 100" ALT= "Активная область" HREF="about.html"></MAP> </BODY>

</HTML>

Пример реализации данного HTML-кода представлен на рисунках 68, 69. Необходимо отметить, что на первом рисунке представлен пример работы с неактивной областью, которая описана в первом теге <AREA>.

На втором же рисунке представлен пример работы с активной областью, описанной во втором теге <AREA>, по щелчку на этой области срабатывает ссылка на файл about.html.



Рисунок 68 – Использование атрибута NOHREF в графических картах (неактивная область)



Рисунок 69 – Использование атрибута NOHREF в графических картах (активная область)

Обработка графических карт на сервере. Для создания графической карты, действия с которой будут обрабатываться на сервере, необходимо включить в атрибут ISMAP, который помогает броузеру правильно разобраться с изображением.

<HTML> <HEAD><TITLE>Пример серверной графической карты</TITLE> </HEAD> <BODY> <H1 ALIGN= "CENTER">Опыт с серверной графической картой</H1> <DIV ALIGN="CENTER"> </DIV> </BODY> </HTML>

Такие графические карты не обеспечивают приемлемость скорости отклика на действия пользователя и потому имеют низкую производительность. Поэтому авторы HTML-страниц, используя по мере необходимости серверные графические карты, отдают предпочтение графическим картам, обрабатываемым у клиента.

8 Фреймы

Фреймы в HTML позволяют авторам представлять документы в нескольких окнах, которые являются частью одного фреймсета. Это обеспечивает дизайнерам способ оставлять некоторую информацию видимой, в то время как другая информация прокручивается или заменяется. Например, в одном фрейме может отображаться статический баннер, во втором навигационное меню, а в третьем - сам документ, который можно прокручивать или переходить к другому с помощью навигации во втором фрейме.

Структура разбивки окна хранится в отдельном файле, который называется документом с фреймами или фреймсетом. Структура фреймов формируется при помощи специального тега <FRAMESET>.

8.1 Задание фреймсета

Формат документа, использующего фреймы, внешне очень напоминает формат обычного документа, только вместо тега BODY используется контейнер FRAMESET, содержащий описание внутренних HTMLдокументов.

<HTML> <HEAD>...</HEAD> <FRAMESET>... </FRAMESET>

</HTML>

Общий контейнер FRAMESET описывает все фреймы, на которые делится экран. Для этого внутри тегов <FRAMESET> и </FRAMESET> находятся теги <FRAME>, описывающие соответствующие фреймы. Сколько тегов <FRAME> содержится внутри тега <FRAMESET>, на такое количество подокон будет разбит фреймсет. Любой фреймсет может включать сколь угодное число подокон, однако не рекомендуется использовать более3 фреймов, так как это существенно понизит скорость загрузки такой Web-страницы.

<FRAMESET COLS= "value" | ROWS= "value"> <FRAME SRC= "URL1"> <FRAME SRC= "URL2">

</FRAMESET>

...

Для разделения фреймсета на вертикальные или горизонтальные окна внутри тега FRAMESET используются два атрибута ROWS и COLS. ROWS задает горизонтальное разделение, а COLS задает вертикальное разделение окна броузера.

Причем размеры окна могут задаваться в различных единицах измерения: простое числовое значение, определяет фиксированный размер подокна в пикселях; значение величины подокна в процентах от 1 до 100; число со звездочкой.

Символ "*" указывает на то, что все оставшееся место будет принадлежать данному фрейму. Если указывается два или более фрейма с описанием "*", или "*,*" то оставшееся пространство делится поровну между этими фреймами. Если перед звездочкой стоит цифра, то она указывает пропорцию для данного фрейма (во сколько раз он будет больше аналогично описанного чистой звездочкой). Описание "3*,*,*", говорит, что будет создано три фрейма с размерами 3/5 свободного пространства для первого фрейма и по 1/5 для двух других.

Приведем несколько примеров:

<FRAMESET COLS= "50,*,50"> — описывает три фрейма, два по 50 точек справа и слева, и один внутри этих полосок.

<FRAMESET ROWS="20%,3*,*"> — описывает три фрейма, первый из которых занимает 20% площади сверху экрана, второй 3/4 оставшегося от первого фрейма места (60% всей площади окна), а последний 1/4 (20% всей площади окна).

<**FRAMESET ROWS="*,60%,*">** — аналогичен предыдущему примеру.

В следующем примере экран разделяется горизонтально на две части: **<FRAMESET ROWS= "50%, 50%">**.

Следующая запись формирует экран, состоящий из трех строк: высота верхней - 20 пикселов, средней - 80 пикселов, нижней - 20 пикселов:

<FRAMESET ROWS= "20, 80, 20">.

Следующий пример создает экран, на котором верхняя строка занимает 10% высоты экрана, средняя — 60%, а нижняя — оставшиеся 30%:

<FRAMESET ROWS= "10%, 60%, 30%">.

В следующем примере верхняя строка имеет высоту 20 пикселов, средняя — 80 пикселов, а нижняя занимает все оставшееся место:

<FRAMESET ROWS= "20, 80, *">.

А в следующем примере создается три столбца: второй имеет фиксированную ширину 250 пикселов. Первый фрейм получает 25% оставшегося пространства, а третий - 75%:

<FRAMESET COLS= "1*,250,3*">.

Раздел <FRAMESET> может содержать вложенные разделы <FRAMESET>, что позволяет создавать довольно сложную структуру фреймов. Атрибуты COLS и ROWS можно использовать одновременно. При этом, фреймы создаются в направлении слева направо для столбцов и сверху вниз для строк. Если указаны оба атрибута, разделы окон создаются слева направо в верхней строке, слева направо во второй строке и т.д. Например,

<FRAMESET ROWS= "50%,50%">

<FRAMESET COLS="*,*"

</FRAMESET>

</FRAMESET>

Помимо рассмотренных выше атрибутов тег <FRAMESET> может иметь также следующие:

- BORDER - задает толщину рамок в пикселях;

- FRAMEBORDER – определяет наличие рамой у содержащихся внутри элемента FRAMESET фреймов, значения 0 или 1, уеѕ или по.

- FRAMESPACING – определяет расстояние между фреймами в пикселах, используется для создания фреймов без рамок.

Тег FRAME описывает каждый фрейм внутри фреймсета и также имеет несколько атрибутов.

<FRAME SRC= "URL" NAME= "frame_name" MARGINWIDTH= "n1" MARGINHEIGHT= "n2" SCROLLING=yes|no|auto NORESIZE>

Рассмотрим более детально каждый атрибут тега <FRAME>.

SRC= "URL" - описывает URL документа, который будет отображен внутри данного фрейма. Если он отсутствует, то будет отображен пустой фрейм. В качестве URL допустимо использование не только HTML, но и GIF- или JPEG-файла.

NAME= "frame_name" - описывает имя фрейма. Имя фрейма может быть использовано для определения действия с данным фреймом из другого HTML-документа или фрейма (как правило, из соседнего фрейма этого же документа). Имя обязательно должно начинаться с символа. Содержимое поименованных фреймов может быть задействовано из других документов при помощи специального атрибута TARGET, описываемого ниже.

MARGINWIDTH задает горизонтальный отступ между содержимым кадра и его границами. Наименьшее значение этого атрибута равно 1. Значения указываются в пикселах.

MARGINHEIGHT задает вертикальный отступ между содержимым кадра и его границами. Наименьшее значение этого атрибута равно 1. Значения указываются в пикселах.

SCROLLING= "yes | no | auto" - позволяет задавать наличие полос прокрутки у фрейма. Параметр уез указывает на обязательное присутствие у фрейма полос прокрутки, параметр no — на отсутствие, auto — определяет наличие полос прокрутки только при их необходимости (значение по умолчанию).

NORESIZE - отменяет возможность изменить при помощи мыши размер фрейма так же, как и размер окна Windows. Если у одного фрейма установлен атрибут NORESIZE, то у соседних фреймов тоже не может быть изменен размер со стороны данного.

BORDERCOLOR — задает цвет рамки.

В раздел <FRAMESET> может быть добавлен также элемент NOFRAME с альтернативным содержанием для отображения в броузерах, не поддерживающих фреймы, или в броузерах с отключенной поддержкой фреймов. Такие броузеры просто отобразят помещенный внутри этого элемента текст /9/.

Рассмотрим пример использования тега <NOFRAME>.

<HTML> <HEAD> <TITLE>Использование тега NOFRAME</TITLE> </HEAD> <FRAMESET> Описание структуры фреймов <NOFRAMES> Ваш броузер не поддерживает фреймы</NOFRAMES> </FRAMESET> </HTML>

При этом, поддерживающий фреймы броузер проигнорирует все, что находится между тегами <NOFRAMES> и </NOFRAMES>. И наоборот, не поддерживающий фреймы броузер проигнорирует все, что находится между тегами <FRAMESET> и </FRAMESET>.

Рассмотрев все возможные атрибуты тегов <FRAMESET> и <FRAME>, приведем простой пример фреймсета.

<HTML>

```
<HEAD><TITLE>Пример фреймсета</TITLE>
</HEAD>
<FRAMESET COLS=*,*>
<FRAME SRC= "1.html" name= "1">
<FRAMESET ROWS=*,*>
<FRAME SRC= "2.html" name= "2">
<FRAME SRC= "3.html" name= "3">
</FRAMESET>
</FRAMESET>
```

</HTML>

В броузере этот фреймсет будет выглядеть следующим образом:

🗿 Пример фреймсета - Microsoft I	nternet Explorer 📃 🗖 🔀
Файл Правка Вид Избранное Се	рвис Справка 🥂
🜀 Назад - 🕥 - 💌 😰 🎸	🔓 🔎 Поиск 📌 Избранное 🥝 🎽
Адрес: 餐 Е ПО НТМL\фреймы\Пример фр	еймсета.html 💟 🔁 Переход 🛛 Ссылки 🎽
Фрейм 1	Фрейм 2
	Фрейм З
🕘 Готово	 З Мой компьютер

Рисунок 70 - Пример разбиения фреймсета

Здесь представлен фреймсет разбитый на три фрейма, сначала разделено окно на два одинаковых столбца, а затем второй столбец был разделен на две одинаковые строки. При этом каждый из фреймов представляет самостоятельный HTML-документ.

Рассмотрим фреймсет, когда все окна имеют одинаковый размер. В следующем примере представлен фреймсет в виде сетки фреймов.

```
<HTML>
<HEAD>
<TITLE>Пример сетки фреймов
</TITLE>
</HEAD>
<FRAMESET ROWS="*,*" COLS= "30%,30%,30%">
<FRAME SRC= "docum1.html">
<FRAME SRC= "docum2.html">
<FRAME SRC= "docum2.html">
<FRAME SRC= "docum3.html">
<FRAME SRC= "docum4.html">
<FRAME SRC= "docum5.html">
</FRAME SRC= "docum5.html">
</FRAME SRC= "docum6.html">
</FRAMESET>
</HTML>
Пример реализации данной HTML-страницы в броузере представлен ниже.
```

🗿 Пример сетки фрейма	ов - Microsoft Internet Explo	orer 📃 🗖 🔀
Файл Правка Вид Избр	анное Сервис Справка	💦 🕺
🕝 Назад 🕤 🕥 - 💽] 🗟 🏠 🔎 Поиск 👷	Избранное 🧭 🔗 · "
Адрес: 🛃 ОСОБИЕ ПО НТМЦ\(фреймы\Пример сетки фреймов.hl	tml 🔽 🛃 Переход 🛛 Ссылки 🎽
Документ 1	Документ 2	Документ 3
Документ 4	Документ 5	Документ 6
ど Готово	I 3	I 1ой компьютер

Рисунок 71 - Пример сетки фреймов

Рассмотрим другой пример фреймсета, где использованы все возможные атрибуты тега <FRAME>. Здесь фрейм 1 не должен содержать полосы прокрутки, а во фрейме 2 вокруг содержимого (изображения) останется пустое пространство и размер фрейма нельзя будет изменять. Между фреймами 3 и 4 не будет четкой границы. Границы (по умолчанию) будут присутствовать между фреймами 1, 2 и 3.

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Фреймы с атрибутами</TITLE>
</HEAD>
```

```
<FRAMESET cols= "33%, 33%, 33%">
<FRAMESET rows= "*,200">
```

```
<FRAME src= "docum1.html" scrolling= "no" >
```

```
<FRAME src= "2.jpg" marginwidth= "10" marginheight= "15"
```

noresize >

</FRAMESET>

```
<FRAME src= "docum3.html" frameborder= "0" >
```

```
<FRAME src= "docum4.html" frameborder= "0" >
```

</FRAMESET>

</HTML>

В броузере это будет выглядеть следующим образом:



Рисунок 72 – Пример использования различных атрибутов тега <FRAME>

8.2 Ссылки во фреймах

Обычно, при активизации пользователем ссылки соответствующий документ появляется в том же окне (или фрейме), что и исходный, в котором была ссылка. Добавление атрибута TARGET позволяет произвести вывод документа в другой фрейм. При этом, в качестве значения атрибута должно использоваться имя фрейма, в котором необходимо открыть файл.

Формат данного атрибута следующий: **TARGET= "frame _name"** Пример:

<А HREF="mydoc.html" TARGET= "Frame1"> Фрейм №1 </А>

Рассмотрим пример использования атрибута Target для создания HTML-страницы, состоящей из трех фреймов. При этом, трем фреймам заданы имена: menu, title и content. Содержание - это список разделов сайта и с каждого названия должна быть установлена ссылка на HTML-документ, в котором хранится его содержимое, должно находиться в файле menu.html. Содержимое разделов сайта должно отображаться в третьем фрейме с именем content. Второй фрейм title будет содержать просто названия сайта, информация в нем меняться не будет.

```
<HTML>
```

<HEAD>

<ТІТLЕ>Пример использования атрибута Target

</TITLE>

</HEAD>

<FRAMESET frameborder= "0" border= "0" framespacing= "0" COLS= "150,*">

<FRAME SRC= "menu.html" name= "menu" scrolling= "auto" noresize> <FRAMESET ROWS="50,*"> <FRAME SRC="title.html" name= "title" marginwidth= "0" scrolling= "auto" noresize> <FRAME SRC= "aboutme.html" name= "content" marginwidth= "0" scrolling="auto" noresize> </FRAMESET> <NOFRAMES> Ваш броузер не поддерживает фреймы </NOFRAMES> </FRAMESET> </HTML>

В третьем документе menu.html указаны ссылки на файлы, которые вызываются из этого фрейма, а открываться будут во фрейме content. Документ menu.html будет отображаться в первом фрейме. Код страницы menu.html представлен ниже.

<HTML>

<HEAD>

<TITLE>Meню</TITLE>

</HEAD>

<BODY bgcolor=skyblue text=blue>

<h3 align=center>Содержание</h3>

 Обо мне

Фотоальбом

Увлечения

Гостевая книга

</BODY>

</HTML>

Реализация данной домашней страницы представлена на рисунке 73.
🕙 Пример использова	ания атрибута Target - Microsoft Int 🔳 🗖 🔀
Файл Правка Вид И	1збранное Сервис Справка 🥂
🕞 Назад 🔹 🅑 👻	🖹 🛃 🏠 🔎 Поиск 🥎 Избранное 🂙
Адрес: 🙋 ример использо	жания атрибута Target.html 💌 芛 Переход 🛛 Ссылки 🎽
Содержание	Моя домашняя страница
 <u>Обо мне</u> <u>Фотоальбом</u> <u>Увлечения</u> <u>Гостевая</u> <u>книга</u> 	Добро пожаловать на мою домашнюю страницу. На этом сайте Вы узнаете много нового о моей жизни, моих увлечениях. Для того, чтобы перемещаться по сайту, воспользуйтесь главным меню. Если Вам понравилась моя домашняя страничка Вы можете прислать свои отзывы, воспользовавшись гостевой книгой.
ê 🗌	🚽 Мой компьютер 🛒

Рисунок 73 – Пример использования атрибута Target

При работе с фреймами атрибуту target можно присваивать и другие значения или ключевые слова. Такие ключевые слова начинаются обязательно с символа подчеркивания:

TARGET="_blank". Данное значение определяет, что документ, полученный по ссылке, будет загружаться в новом окне броузера.

TARGET="_self". Данное значение определяет, что документ, полученный по ссылке, будет загружаться в тот же фрейм, из которого ссылаются.

TARGET="_parent". Данное значение определяет, что документ, полученный по ссылке, будет загружаться в родительский фрейм структуры FRAMESET. Если родительского фрейма нет, то данное имя аналогично "_self".

TARGET="_top". Данное значение определяет, что документ, полученный по ссылке будет загружаться в полное окно (закрывая все остальные).

Рассмотрим еще один пример использования атрибута target, где в качестве значений этого атрибута используются зарезервированные слова.

```
<HTML>
<HEAD>
<TITLE>Пример ссылок во фреймах</TITLE>
</HEAD>
<FRAMESET COLS ="2*,*,*">
<FRAME SRC="frame_a.html" name="A">
<FRAME SRC="frame_b.html" name="B">
```

<FRAME SRC="frame_c.html" name="C"> </FRAMESET> </HTML>

Основной фреймсет представлен на рисунке 74.

🗿 Пример ссылок во фреймах - Microsoft Inte	rnet Explorer 📃 🗖 🔀
Файл Правка Вид Избранное Сервис Справк	a 🥂
🕝 Назад 👻 🐑 💌 😰 🏠 🔎 Поис	< 👷 Избранное 🧭 🗟 - 嫨 🔯 - 🎽
Адрес: 🙋 онные технологии\ПОСОБИЕ ПО НТМL\Приме	р ссылок во фреймах.html 🝸 🄁 Переход 🛛 Ссылки 🎽
<u>1. Загрузка документа во фрейм В</u>	
<u>2. Загрузка документа во фрейм С</u>	
<u>4. Загрузка документа в новое окно</u>	
<u>5. Загрузка документа в полное окно</u>	
<u>6. Загрузка документа в текущий фрейм</u>	
🙆 Готово	😏 Мой компьютер

Рисунок 74 – Основной фреймсет

Из рисунка 74 видно, что из фрейма "А" будет осуществляться ссылки на различные файлы. При этом сам документ frame_a.html будет реализован следующим образом:

```
<HTML>
<HEAD>
<ТІТLЕ>Документ для фрейма А
</TITLE>
</HEAD>
<BODY><A HREF="test.html" TARGET="B">1. Загрузка документа во
фрейм В</А>
<A HREF="test.html" TARGET="C">2. Загрузка документа во фрейм
C < A > P >
<P>
<А HREF="test.html" TARGET=" blank">4. Загрузка документа в новое
окно</А></Р>
<P>
<A HREF="test.html" TARGET=" top">5. Загрузка документа в полное
окно</А></Р>
<P>
<А HREF="test.html" TARGET=" self">6. Загрузка документа в текущий
фрейм
</A>
```

</BODY> </HTML>

При этом документ frame_a.html будет выглядеть следующим образом.



Рисунок 75 – Документ frame_a.html

8.3 Плавающие фреймы

Для вставки одного HTML-документа в тело другого средствами броузера пользователя, а не сервера, служит тег <IFRAME>:

<IFRAME SRC="адрес_документа"></IFRAME>

Внешне этот элемент выглядит как прямоугольная область, в которой отображается документ из некоторого HTML-документа. Такие окна иногда называют плавающими фреймами.

Плавающий фрейм ведет себя как обычный фрейм. Среди его свойств важно свойство *align* - выравнивание плавающего фрейма относительно окружающего содержимого документа. Применение данного атрибута аналогично при работе с изображениями при использовании тега . Возможные значения:

- absbottom выравнивает нижнюю границу фрейма по подстрочной линии символов окружающего текста;
- absmiddle выравнивает середину фрейма по центральной линии между top и absbottom окружающего текста;
- baseline выравнивает нижнюю границу фрейма по базовой линии окружающего текста;
- left выравнивает фрейм по левому краю элемента-контейнера;
- middle выравнивает воображаемую центральную линию окружающего текста по воображаемой центральной линии фрейма;
- right выравнивает фрейм по правому краю элемента-контейнера;

- texttop выравнивает верхнюю границу фрейма по надстрочной линии символов окружающего текста;
- top выравнивает верхнюю границу фрейма по верхней границе окружающего текста.

Ниже приведен пример загрузки двух HTML-документов в плавающие фреймы.

<hr/>HTML>

<HEAD>

<TITLE>Плавающие фреймы

</TITLE>

</HEAD>

<BODY>

<H2> Включение HTML-документов на стороне клиента

</H2>

<H3>Использование тега IFRAME</H3>

<Н4>Здесь в окнах показаны страницы Web-сайтов:

</H4>

<IFRAME src= "docum1.html" scrolling= "yes" align= "right" frameborder= "1">

<IFRAME src= "docum2.html" scrolling= "yes" align= "top" frameborder= "1">

</IFRAME>

```
<P>Использованы различные значения свойства align</P></BODY>
```

</HTML>

Реализация данного HTML-кода представлена на рисунке 76.



Рисунок 76 – Плавающие фреймы

9 HTML-формы

Формы являются наиболее популярным способом "обратной связи" с пользователем. С помощью HTML можно создавать как простые формы, предполагающие выбор одного из нескольких ответов, так и сложные формы для заказов или для того, чтобы получить от пользователей страницы какиелибо комментарии и пожелания.

Форма представляет собой несколько полей, где пользователь может ввести некоторую информацию, либо выбрать какую-то опцию. После того, как пользователь отправит информацию, она обрабатывается программой (скриптом), размещенной на сервере. Существует также возможность обрабатывать формы "на стороне клиента", встраивая в свои страницы скрипты, написанные на языках JavaScript и Visual Basic Script.

9. 1 Задание формы

Для реализации различных форм в HTML используется один тег <FORM>. Элемент <FORM> обозначает документ как форму и определяет границы использования других тегов, размещаемых в форме:

<FORM METHOD= "get|post" ACTION= "URL" ENCTYPE> Элементы формы и другие элементы HTML</FORM>

Обязательные атрибуты тега <FORM>:

- ACTION указывает URL скрипта, который примет и обработает данные формы. Если эта опция не указана, данные отправляются по адресу страницы, на которой размещена форма.
- МЕТНОД определяет, как должны обрабатываться выходные данные из формы, т. е. определяет метод передачи данных программе-обработчику формы. Возможные значения GET или POST. Значение GET, используемое по умолчанию, предписывает посылать информацию формы вместе с URL, а значение POST предписывает посылать информацию формы отдельно от URL. Значение POST используется обычно в случае отправки данных формы по электронной почте или при необходимости передавать значительный объем информации.

Необязательные атрибуты тега <FORM>:

- ENCTYPE определяет тип MIME декодирования информации (значение этого атрибута по умолчанию – "application/x-www-formurlencoded"),
- SCRIPT используется для передачи URI скрипта. Язык скрипта и интерфейс пользователя при этом не являются частью спецификации HTML.
- NAME указывает имя формы. Это необходимо, если требуется доступ к данным формы с помощью встроенного скрипта на JavaScript или Visual Basic Script, а также в том случае, если данные формы предназначены для отправки по электронной почте.

Рассмотрим примеры:

Заголовок формы, используемой на поисковой машине Yandex, имеет вид: <form name="web" method="get" action="/yandsearch">.

Заголовок формы, отправляющей информацию на адрес E-mail, может иметь вид: <form action = "mailto:LG-77@mil.ru?subject = Internet-Test" method = "post" enctype="text/plain" name="Q" onSubmit= "return Validate()">

Здесь форма имеет имя Q, в качестве action указана отправка сообщения по E-mail на адрес LG-77@mail.ru с темой "Internet-Test", опция епстуре предписывает отправлять данные формы как текст, а опция onSubmit связывает отправку формы с функцией Validate(), написанной на Javascript. Этот прием широко используется для проверки корректности заполнения формы.

Внутри тега <FORM>...</FORM> заключаются основные элементы управления формами. К ним, в частности, относятся однострочное текстовое поле, ниспадающий список, флажок, прокручивающееся поле, переключатели, многострочное текстовое поле. Все они задаются с помощью соответствующих тегов. Рассмотрим основные элементы управления формами.

9.2 Задание элементов форм

9.2.1 Тег INPUT

Контейнер INPUT является самым распространенным контейнером HTML-формы. Существует целых 10 типов этого контейнера (text, image, submit, reset, hidden, password, file, checkbox, radio, button), причем каждый из них отображается по-разному. В общем виде контейнер <INPUT> имеет вид:

<INPUT NAME="Имя" ТҮРЕ= "Тип" [вариации параметров, зависящие от типа] >

Чаще всего контейнер INPUT применяется для организации текстового поля ввода: например, для ввода списка ключевых слов или для заполнения регистрационных форм.

Однострочное текстовое поле - INPUT (text). Тип text контейнера INPUT определяет текстовое поле ввода, в котором пользователь Webстраницы может ввести свою информацию. Поле типа text имеет в общем случае следующий вид:

<INPUT NAME= "Имя" TYPE= "text" SIZE= "number" MAXLENGTH= "number">

Атрибут NAME используется для именования поля как элемента формы. Имя поля попадает в запрос при отправке на CGI-модуль (левая часть пары "имя_поля=значение"), а также применяется в JavaScript для чтения и изменения значений текстовых полей формы /9/.

Атрибут SIZE задает размер видимой на экране части текстового поля. Атрибут MAXLENGTH задает максимальный размер поля. Он полезен в тех случаях, когда требуется ограничить вводимые данные по длине. Рассмотрим пример использования тега <INPUT> с атрибутом Туре= "text". <**HTML>**

```
<HIML>
<HEAD><TITLE> Формы в HTML </TITLE>
</HEAD></BODY><FORM>
Введите имя <INPUT TYPE=TEXT SIZE=10 MAXLENGTH=15>
Введите фамилию <INPUT TYPE=TEXT SIZE=20 MAXLENGTH=15>
</FORM>
</HTML>
```

В броузере это будет выглядеть следующим образом:

🕙 Формы в HTML - Microsoft Internet Explorer	
Файл Правка Вид Избранное Сервис Справка	1
🕞 Назад 👻 🕥 т 💽 👔 🎾 Поиск	*
Адрес: 🙋 СОБИЕ ПО НТМL\формы\Формы.html 💙 🎅 Переход	Ссылки »
Введите имя Введите фамилию	
🕙 Готово 🛛 🚽 🧕 Мой компьютер	

Рисунок 77 – Пример однострочного тестового поля

В приведенном примере для каждого поля определена максимальная длина поля равная 15 символам. В первом случае строчка будет "прокручиваться" в горизонтальном направлении справа налево при превышении размера видимой области, равной 10 символам. Во втором случае ввод остановится во второй трети поля.

При разработке различных форм часто требуется выравнивание элементов. Например, при реализации формы-анкеты нужно выравнивать графы. В этом случае формы с текстовыми полями помещаются в таблицу. Ниже приведен пример использования таблиц для выравнивания элементов форм.

```
<HTML>
<HEAD>
<TITLE> Использование таблиц в формах </TITLE>
</HEAD>
<BODY>
<FORM>
<TABLE>
<TR><TD>Имя: </TD>
```

```
<TD><INPUT SIZE=5 MAXLENGTH=15></TD>
</TR>
<TR><TD>Фамилия: </TD>
<TD><INPUT SIZE=10 MAXLENGTH=15></TD>
</TR></TABLE>
</FORM>
</BODY>
```

Пример реализации данной HTML-страницы в броузере представлен на рисунке 78. В данном примере хорошо видно выравнивание полей по столбцам.

🕘 Ист	юльзова	ние та	аблиц в фо	рмах - Л	lic [
Файл	Правка	Вид	Избранное	Сервис	Справ	, »	
GH	азад т	0	💌 🛃	6	🔎 Пои	ск	»
Адрес:	🙋 е таб.	лиц в ф	ормаx.html 💊	🖌 🔁 Пе	реход	Ссыл	пки »
Имя Фак	: иилия:						
							~
۲ 🕑			🔙 🛃 Мой ка	омпьютер			

Рисунок 78 – Использование таблиц в формах

Пароль - INPUT (password). Тип password определяет текстовое поле, которое позволяет скрыть набираемый текст от посторонних глаз. По своим атрибутам поле типа password не отличается от поля типа text. При использовании этого поля следует помнить, что пароль или любая другая информация, которая вводится в поле типа password, будет передаваться по сети в виде ASCII-символов и. будет доступна для просмотра при условии ее захвата посторонним лицом. Поле типа password имеет в общем случае следующий вид: <INPUT NAME= "Имя" TYPE= "password" SIZE= "number" MAXLENGTH= "number">

Рассмотрим пример использования тега <INPUT> с атрибутом Type= "password".

```
<HTML>
<HEAD><TITLE> Использование пароля </TITLE></HEAD>
<BODY>
<FORM> Пароль: <INPUT TYPE=password SIZE=10 > </FORM>
</BODY>
</HTML>
```

В броузере эта HTML-страница будет выглядеть следующим образом:

🗿 Ист	тользова	ние па	ароля - Мі	crosoft	Interne	
Файл	Правка	Вид	Избранное	Серви	ис Справка	i 🥂
G	Іазад т	0	× 2	1 🏠	🔎 Поиск	**
Адрес:	🖉 Испол	њзован	ие пароля.h	tml 🔽	🔿 Переход	Ссылки »
Паро	оль: •••	••••				~
🙆 Γοτα	D		1 😴	1ой комп	ьютер	2

Рисунок 79 – Использование пароля

Невидимое поле - INPUT (hidden). Тип hidden был введен в формы по причине отсутствия поддержки сеансов в протоколе HTTP. Впоследствии были разработаны более эффективные способы эмуляции сеансов, например, cookie. Тем не менее, поля hidden до сих пор могут использоваться для поддержки сеансов. В ряде случаев, когда запросы вынуждены проходить через сито систем защиты, это единственный способ реализации поддержки сеансов. Поле типа hidden имеет в общем случае следующий вид:

<INPUT NAME= "Имя" TYPE= "hidden">

Рассмотрим пример использования тега <INPUT> с атрибутом Type= "hidden".

<FORM> <INPUT MAXLENGTH=250 TYPE=hidden> </FORM>

В данном случае задано невидимое поле шириной в 250 символов.

Флажок - INPUT (checkbox). Тип checkbox применяется в качестве селектора. Поле типа checkbox имеет в общем случае следующий вид:

<INPUT NAME= "Имя" TYPE= "checkbox">

Поле типа checkbox можно применить, например, в заявке на комплектующие, где нужно выбирать из нескольких заранее определенных позиций. Рассмотрим пример использования тега <INPUT> с атрибутом Туре= "checkbox".

<HTML>

<HEAD>

<TITLE> Использование checkbox </TITLE>

</HEAD>

<BODY>Что желаете купить?

<FORM>

1. Принтер - <INPUT NAME=print TYPE=checkbox>

2. Сканер - <INPUT NAME=scan TYPE=checkbox>


```
3. Монитор - <INPUT NAME=monitor TYPE=checkbox>
```

</FORM>

</BODY> </HTML>

В броузере это будет выглядеть следующим образом:



Рисунок 80 – Использование checkbox

При использовании полей checkbox следует учитывать, что они хороши там, где не нужно выбирать. Отображение поля типа checkbox в запросе зависит от параметров, которые указаны в контейнере INPUT. Для checkbox, как и для любого другого поля, необходимо применение атрибута NAME, иначе значение поля не будет учтено в запросе. Кроме того, в checkbox используются атрибуты CHECKED и VALUE.

По умолчанию, если поле отмечено как выбранное пользователем, в запрос попадает пара "name=on", где name - имя поля. Атрибут VALUE позволяет изменить значение выбранного поля. Например:

<FORM> Принтер - <INPUT NAME=print VALUE=print
TYPE=checkbox> </FORM>

В этом случае вместо "on" в правой части равенства появится "print".

Атрибут CHECKED определяет состояние поля по умолчанию, т.е. в момент первоначальной загрузки страницы или выбора кнопки Reset. Если он указан, то поле считается по умолчанию выбранным (отмеченный прямоугольник). Рассмотрим пример использования атрибута checked.

<hr/>

<HEAD>

<TITLE> Использование checkbox-checked </TITLE>

</HEAD>

<BODY>Что желаете купить?

<FORM>

1. Принтер - <INPUT NAME=print TYPE=checkbox CHECKED>

2. Сканер - <INPUT NAME=scan TYPE=checkbox>

3. Монитор - <INPUT NAME=monitor TYPE=checkbox>

</FORM> </BODY> </HTML>

В броузере эта HTML-страница будет выглядеть следующим образом:



Рисунок 81 – Использование атрибута checked

Важным отличием поля типа checkbox от поля типа radio является обработка полей с одинаковыми именами. В случае поля типа checkbox - это разные поля, и их значения никак не связаны между собой. При одновременном выборе полей с одинаковыми именами в запрос попадут все выбранные поля. При этом пары "имя=значение" будут просто повторяться. Другой вариант можно реализовать только через поля типа radio.

Переключатель - INPUT (radio). Тип radio контейнера INPUT определяет поле "переключатель". Данный тип применяется там, где необходимо обеспечить выбор из нескольких заданных взаимоисключающих вариантов. Поле типа radio имеет в общем случае следующий вид:

<INPUT NAME= "Имя" TYPE= "radio">

Например, в анкете может быть графа "Пол" и нужно выбрать один из двух вариантов, либо мужской, либо женский. Именно здесь и применяются атрибут radio. Рассмотрим пример использования переключателей. **<HTML>**

```
<HTML>
<HEAD><TITLE> Использование radio </TITLE>
</HEAD>
<BODY>
<FORM>
Пол:
<INPUT NAME=pol TYPE=radio>Мужской
<INPUT NAME=pol TYPE=radio>Женский
</FORM>
</BODY>
</HTML>
```

В броузере это будет выглядеть следующим образом:



Рисунок 82 – Использование radio

В данном случае при выборе одного из вариантов со второго автоматически снимается отметка. Это главное отличие типа поля radio от типа checkbox, имена полей одинаковые.

Если этот пример оставить так как он есть, то скрипт, который будет принимать данные, не получит сведений о том, какой из вариантов был выбран. В любом случае будет выдаваться запрос типа: "?pol=on". Если вариантов не выбирать, то соответствующая пара "имя_поля-значение" вообще не появится в запросе.

Для того, чтобы указать выбранный вариант в контейнере INPUT, нужно ввести атрибут VALUE:

<FORM>

Пол:

<INPUT NAME=pol TYPE=radio VALUE=m>Мужской <INPUT NAME=pol TYPE=radio VALUE=f>Женский </FORM>

В данном случае вместо "on" передается соответствующее значение. Скрипт теперь в состоянии различить выбранный вариант.

Если в контейнерах INPUT типа radio задать разные имена, то разницы между этим типом и типом checkbox почти не будет:

<FORM>

Пол:

<INPUT NAME=pol TYPE=radio VALUE=m>Мужской <INPUT NAME=pol2 TYPE=radio VALUE=f>Женский </FORM>

В данном контексте следует рассматривать и применение атрибута CHECKED в полях этого типа. Как и в checkbox, атрибут CHECKED позволяет определить значение поля по умолчанию.

```
<HTML>
<HEAD>
<TITLE> Использование radio </TITLE></HEAD>
<BODY>
<FORM>
Пол: <br>
<INPUT NAME=pol TYPE=radio VALUE=m CHECKED> Мужской<br>
<INPUT NAME=pol TYPE=radio VALUE=f >Женский <br>
</FORM>
</BODY>
</HTML>
```

В броузере это будет выглядеть следующим образом:



Рисунок 83 – Использование radio

Применяться атрибут Checked может, например, на HTML-странице, где речь идет о приеме на вредную работу. На эту работу нанимаются преимущественно мужчины. Альтернативой по умолчанию станет значение "m" для поля с именем pol. Значение по умолчанию устанавливается либо при первичной загрузке страницы, либо при выборе кнопки типа reset.

Рисунок - INPUT (image). Кроме текста в полях формы можно вводить и координаты местоположения манипулятора "мышь". Для этой цели служит тип image контейнера INPUT. В данном типе множество атрибутов контейнера INPUT дополняется атрибутами контейнера IMAGE.

Рассмотрим пример использования тега <INPUT> с атрибутом Туре= "IMAGE".

<FORM>

<INPUT TYPE=image SRC=image.gif NAME=i ALIGN=left BORDER=0> </FORM>

При размещении поля этого типа атрибут NAME, кроме стандартных функций именования, выполняет еще и функции атрибута ALT из контейнера IMG. При наведении манипулятора "мышь" на картинку

появляется значение поля NAME. Выбор данного поля манипулятором "мышь" приводит к немедленной передаче данных из формы серверу, а затем скрипту. При этом координаты манипулятора "мышь" передаются в виде "имя поля.х=DD&имя поля.y=DD".

Кнопка - INPUT (button). Во всех современных интерфейсах есть объекты, имитирующие кнопки управления. Контейнер INPUT позволяет создать кнопку при помощи типа button. Изначально в формах было только две кнопки: Submit и Reset.

Submit позволяет инициировать отправку данных серверу из формы и перезагрузку текущей страницы. Reset позволяет выставить значения полей формы по умолчанию. Это бывает необходимо в случае неправильного ввода данных.

Тип button контейнера INPUT является обобщением и расширением случаев submit и reset на более широкий класс объектов, которые принято называть кнопками. Форма с кнопками может выглядеть, например, следующим образом:

<FORM>

<INPUT TYPE=button VALUE= "Кнопка"> </FORM>

Кнопки, в отличие от типа image, не вызывают события submit в случае их нажатия. Кнопка также гораздо удобнее гипертекстовой ссылки, если не требуется перезагрузка страницы. При выборе гипертекстовой ссылки перезагрузка произойдет обязательно, если только не позаботиться об этом при указании атрибута HREF контейнера A (Anchor). Можно использовать гиперссылки и кнопки вместе. Например, HTML-страница обработки заказов доставки пиццы. При нажатии кнопки ссылка связывается с файлом, в котором представлены расценки и ингредиенты различных сортов. Рассмотрим пример такой HTML-страницы.

<HTML>

```
<HEAD>
<TITLE>Кнопка-гиперссылка
</TITLE>
</HEAD>
<BODY bgcolor=pink>
<H2>Заказ пиццы</H2>
<H3>Введите свои данные для приема заказа</H3>
ФИО: <INPUT TYPE=text name=t1 size=20><br>
Адрес доставки: <INPUT TYPE=text name=t2 size=20><br>
Выберите пиццу:
<SELECT>
<OPTION> с мясом
<OPTION> с колбасой
<OPTION> с грибами
<OPTION> с сыром
<OPTION> с оливками
```

```
</SELECT><BR>
<A HREF= "1.HTML"><INPUT TYPE=button VALUE="Обработать
заказ">
</A>
</BODY>
</HTML>
В броузере эта HTML-страница будет выглядеть следующим образом:
В броузере эта HTML-страница будет выглядеть следующим образом:
```



Рисунок 84 – Использование кнопки в качестве гиперссылки

Кнопка отправки данных - INPUT (Submit). Кнопка Submit - едва ли не самый важный элемент формы. Она инициирует отправку данных формы на сервер. При указании поля типа Submit следует учитывать, что если полю дать имя, то оно появится в запросе. Рассмотрим пример использования тега <INPUT> с атрибутом Type= "Submit".

```
<HTML>
<HEAD>
<TITLE>Использование submit
</TITLE>
</HEAD>
<FORM>
<INPUT TYPE=submit VALUE="Отправить?">
<INPUT TYPE=submit NAME="Подача запроса">
</FORM>
</BODY>
</HTML>
```

Пример реализации данной HTML-страницы представлен на рисунке 85.

🕘 Ист	10.ЛЬЗОВа	ние с	ubmit - M	icrosoft	Inter	net Exp		
Файл	Правка	Вид	Избранно	е Серв	ис С	правка		-
G	Іазад 🔻	•	× (2 🏠	ρ	Поиск		»
Адрес:	🥙 ормы	і\Исполь	зование su	ıbmit.html	~	🗲 Переход	Ссы	лки »
0	тправит	гь?	Под	ача заг	роса			~
🕘 Γοτα	во			🛛 😼 Moi	й комп	ьютер		

Рисунок 85 - Использование кнопки Submit

Пример демонстрирует сразу две особенности полей типа submit. Вопервых, если задать значение атрибута VALUE, то оно отобразится в качестве текста на кнопке (первая кнопка). Во-вторых, если не указывать атрибута VALUE, то будет подставлено значение кнопки по умолчанию. Втретьих, если есть имя, то при нажатии на кнопку произойдет событие onSubmit, и в запросе, который будет послан на сервер, появится пара "имя поля submit-значение".

В одной форме может быть столько кнопок типа Submit, сколько необходимо для работы. При этом их желательно именовать. В противном случае, скрипт не сможет определить, какая из кнопок вызвала передачу данных.

Кнопка сброса данных - INPUT (Reset). Поле типа Reset представляет собой кнопку, при нажатии на которую все исправления и назначения в форме принимают свои значения по умолчанию. Кнопка восстанавливает статус умолчания для всех полей формы /10/. Рассмотрим пример использования тега <INPUT> с атрибутом Туре= "Reset".

```
<HTML>
<HEAD>
<TITLE>Использование reset
</TITLE>
</HEAD>
<BODY>
<FORM>
<INPUT TYPE=text >
<INPUT TYPE=reset NAME=s value="Очистить">
</FORM>
</BODY>
</HTML>
Пример реализации данной HTML-страницы представлен на рисунке
```

86.

🕘 Ист	юльзова	ание ге	eset - Micro	soft Inte	ernet (
Файл	Правка	Вид	Избранное	Сервис	Справк	а	1
Эн	азад т	0	💌 🛃	6	🔎 Поиск		**
Адрес:	🕘 Испо	льзован	ие reset.html	💌 🔁 I	Переход	Ссыл	тки »
				Очисти	ть		<
570			10				
🕑 Го			🔡 Мой	компьют	ер		

Рисунок 86 - Использование кнопки Reset

Выбор кнопки Reset "Очистить" вернет значение текстового поля в исходное состояние. На практике элементы Submit и Reset используются вместе. При создании сложных форм всегда должна присутствовать кнопка позволяющая отправить данные из полей формы, и всегда должна быть кнопка очистки содержимого полей.

Приведем простой пример проверки пароля пользователя. Кнопка "Найти" проверяет пароль пользователя, а кнопка "Очистить" сбрасывает содержимое этого поля, и позволяет пользователю снова ввести пароль. **<HTML>**

```
<HEAD> <TITLE> Использование submit и reset </TITLE></HEAD></BODY></BODY></FORM> Введите пароль:
<input type= "password" name= "paswd" size=20>
<input type= "submit" value= "Hайти">
<input type= "reset" value= "Oчистить">

</FORM> </BODY>

B броузере данная HTML -страница будет выглядеть следующим обг
```

В броузере данная HTML-страница будет выглядеть следующим образом:

🕙 Исполы	ование s	ubmit и res	et - Micr	osoft Inte	ernet	Explorer	_	
Файл Пра	вка Вид	Избранное	Сервис	Справка				- 1
🕞 Назад	• 🔊	- 💌 💈	6	🔎 Поиск	*	Избранное	Ø	**
Адрес: 🙋 () HTML\фор	мы\Использов	ание subm	it и reset.ht	tml 💌	🔁 Пере»	⊙д ⊂	сылки »
Введите	пароль:			Hai	йти	Очисти	1ТЬ	~
								×
é				۹ 😼	Чой ком	пьютер		

Рисунок 87 – Пример формы авторизации пользователя

Файл - INPUT (file). Элемент <INPUT type=file> используется для передачи больших объемов данных в World Wide Web. Рассмотрим пример использования тега <INPUT> с атрибутом Type= "file".

<FORM ENCTYPE=multipart/form-data METHOD=post ACTION=mailto:Lg@mail.ru> <INPUT TYPE=file NAME=file> <INPUT TYPE=submit VALUE= "Опубликовать"> </FORM>

В данном примере отправляется пользователю LG-77 на хосте mail.ru файл по протоколу SMTP. Для того чтобы обеспечить составное тело сообщения, применяется тип кодирования содержания multipart/form-data.

По умолчанию используется другой метод кодирования — application/x-www-form-urlencoded, поэтому его надо переопределить через атрибут ENCTYPE. Но кардинальное решение заключается в вызове скрипта, который примет файл и выполнит его преобразование, разместит файл на сервере, или сделает что-либо еще.

<FORM ENCTYPE=multipart/form-data METHOD=post ACTION=file.cgi><INPUT TYPE=file NAME=file>

<INPUT TYPE=submit VALUE= "Опубликовать"> </FORM>

Одно из преимуществ данного подхода заключается в том, что файл присоединяется к сообщению как двоичные данные. При вставке содержимого файла в тело HTTP-сообщения никакого преобразования не происходит.

9.2.2 Тег SELECT

Контейнер SELECT применяется в формах для создания раскрывающегося списка (ниспадающего меню) или так называемых списков прокрутки. При этом можно организовать выбор из множества вариантов только одного или отметить сразу несколько вариантов отчетов. В общем случае контейнер <SELECT> имеет следующий вид.

<SELECT> <OPTION> <OPTION> <OPTION> <OPTION>

</SELECT>

Рассмотрим простой пример HTML-страницы с использованием раскрывающегося списка.

<HTML> <HEAD> <TITLE> Использование select </TITLE> </HEAD> <BODY>

```
<FORM>
<SELECT>
<OPTION>Понедельник
<OPTION>Вторник
<OPTION>Среда
<OPTION>Четверг
<OPTION>Пятница
<OPTION>Пятница
<OPTION>Воскресенье
</SELECT>
</FORM>
</BODY>
</HTML>
```

В результате появится следующее ниспадающее меню (рисунок 88).



Рисунок 88 – Пример ниспадающего меню

При нажатии на стрелочку справа появится весь список (рисунок 89).



Рисунок 89 - Ниспадающий список

По умолчанию в качестве текущей опции устанавливается первая опция списка. Значение по умолчанию устанавливается либо кнопкой RESET, либо при наборе в поле LOCATION броузера, либо при переходе по гипертекстовой Для принудительного указания ссылке. выбора по умолчанию следует воспользоваться атрибутом SELECTED контейнера OPTION. Выбранное значение альтернативы становится значением поля select. И запросе передается "имя поля selectв пара значение альтернативы".

Для того, чтобы задать опцию, выбранную по умолчанию, необходимо использовать атрибут Selected. При этом предыдущий HTML-код изменится следующим образом.

<HTML> <HEAD> <TITLE> Использование select </TITLE> </HEAD> <BODY> <FORM> <SELECT> <ОРТІОN>Понедельник <OPTION>Вторник <OPTION selected>Среда <ОРТІОN>Четверг <ОРТІОN>Пятница <ОРТІОN>Суббота <OPTION>Воскресенье </SELECT> </FORM> </BODY> </HTML>



Рисунок 90 – Использование атрибута selected

Для того, чтобы сформировать список прокрутки в контейнере SELECT, необходимо указать в атрибуте SIZE число видимых элементов.

```
<HTML>
<HEAD> <TITLE> Использование select</TITLE>
</HEAD>
<BODY>
<FORM>
<SELECT SIZE=5>
<ОРТІОN>Понелельник
<OPTION>Вторник
<OPTION>Среда
<ОРТІОN>Четверг
<ОРТІОN>Пятница
<ОРТІОN>Суббота
<OPTION>Воскресенье
</SELECT>
</FORM>
</BODY>
</HTML>
```

В броузере это будет выглядеть следующим образом:



Рисунок 91 – Использование атрибута Size

В данном случае реализовано окно прокрутки для пяти вариантов. Атрибут SIZE не изменяет самого характера работы с полем. Оно продолжает оставаться списком вариантов, из которого можно выбрать только один. Для организации множественного выбора в контейнере SELECT необходимо указать атрибут MULTIPLE /11/.

```
<HTML>
<HEAD> <TITLE> Использование select </TITLE>
</HEAD>
<BODY>
<FORM>
<SELECT SIZE=5 MULTIPLE>
```

<ОРТІОN SELECTED>Понедельник
<ОРТІОN>Вторник
<option selected="">Среда</option>
<ОРТІОN>Четверг
<ОРТІОN>Пятница
<ОРТІОN>Суббота
<option>Воскресенье</option>
В броузере это будет выглядеть следующим образом:



Рисунок 92 – Использование Multiple

В данном примере по умолчанию выбрано два дня недели: понедельник и среда. По идее, пользователь должен иметь возможность либо устанавливать отметку о выборе опции, либо снимать ее. При этом таких отметок в одном списке может быть несколько.

Необходимо отметить, что контейнер OPTION никогда не используется сам по себе. Его применение имеет смысл только в контексте контейнера SELECT. В контейнере OPTION можно указывать два атрибута: VALUE и SELECTED.

Атрибут VALUE позволяет задать элементу <SELECT> значение, отличное от альтернативы, которая определена в OPTION. Это позволяет существенно упростить обработку запроса. Например, при выборе дней недели вместо их полных имен можно использовать сокращения.

Помимо этого, можно использовать <SELECT> вместе с кнопкой Submit. Тогда все выбранные элементы сразу отправятся по кнопке Submit по адресу, указанному в action.

Рассмотрим пример совместного использования тега <SELECT> с кнопкой Submit.

```
<HTML>
<HEAD> <TITLE> Использование select
</TITLE>
</HEAD>
<BODY>
<FORM><SELECT NAME=s>
<OPTION VALUE=1>Понелельник
<OPTION VALUE=2>Вторник
<OPTION VALUE=3 selected >Среда
<OPTION VALUE=4>Четверг
<OPTION VALUE=5>Пятница
<OPTION VALUE=6>Суббота
<OPTION VALUE=7>Воскресенье
</SELECT>
<INPUT TYPE=submit value="Подача запроса">
</FORM> </BODY>
</HTML>
```

Реализация данной HTML-страницы представлена на рисунке 93.



Рисунок 93 – Использование Select и Submit

При этом контейнер OPTION не именуется. Более того, при программировании в JavaScript все действия над значениями опций выполняются в рамках объекта SELECT.

9.2.3 Ter TEXTAREA

Данный контейнер это первая попытка разрешить пользователю вводить большие текстовые фрагменты. Контейнер удобен там, где не требуется форматное представление данных, либо поле должно содержать произвольный фрагмент текста, например, комментарий. Если контейнер употребляется в документе, то нужно указывать как тег начала контейнера, так и тег его конца.

Содержание поля TEXTAREA можно передать как методом GET, т.е. в URL-скрипта, так и методом POST.

Атрибуты контейнера TEXTAREA позволяют сформировать на экране окно, где будет отображаться значение поля, и правила этого отображения. Атрибут COLS определяет число столбцов в окне, ROWS число строк.

Рассмотрим пример использования тега <TEXTAREA>.

<HTML>

<HEAD>

<TITLE>Пример textarea </TITLE> </HEAD> <BODY>

<FORM>

Присылайте свои отзывы и пожелания:
<TEXTAREA NAME=address COLS=50 ROWS=5>Teкct...

</TEXTAREA>

```
<INPUT TYPE=submit NAME=button value="Отправить">
```

<INPUT TYPE=reset NAME=button value="Cброс">

</FORM>

</BODY>

</HTML>

В броузере эта страница буде выглядеть следующим образом:

🖀 Пример textarea - Microsoft Internet Explorer 📃 🗖	
Файл Правка Вид Избранное Сервис Справка	
🕝 Назад 🕘 🕤 💽 🛃 🏠 🔎 Поиск 👷 Избранное 🧔	»
Адрес: 🕘 О НТМL\формы\Использование textarea.html 💟 🎅 Переход 🛛 Ссыл	ки »
Присылайте свои отзывы и пожелания: Текст	X
Отправить Сброс	1
餐 Готово 🛛 🚽 Мой компьютер	

Рисунок 94 – Использование Textarea

9.3 Атрибуты форм

Как уже было сказано ранее, конструкция формы состоит из основного тега <FORM> и атрибутов ACTION, МЕТНОД и ENCTYPE.

<FORM METHOD=get|post ACTION="URL" ENCTYPE>Элементы управления формой </FORM>

9.3.1 Атрибут АСТІОЛ

Атрибут ACTION описывает URL, который будет вызываться для обработки формы. Данный URL почти всегда указывает на CGI-программу, обрабатывающую данную форму.

Пример:

<FORM METHOD="POST" ACTION="chkpaswd.exe" >

При обработке информации пользователя из форм можно также использовать другие известные методы: копирование в электронный почтовый ящик, перенос на Web-страницу, сохранение в базу данных, сохранение в файл и т.д.

9.3.2 Атрибут МЕТНОД

Атрибут Method определяет метод передачи данных пользователя из форм. На сегодняшний день наиболее популярны два метода – GET и POST.

Метод GET - информация из формы добавляется в конец URL, который был указан в описании заголовка формы. CGI-программа (CGI-скрипт) получает данные из формы в виде параметра переменной среды QUERY STRING.

Метод POST - данный метод передает всю информацию о форме немедленно после обращения к указанному URL. CGI- программа получает данные из формы в стандартный поток ввода. Сервер не будет пересылать сообщение об окончании пересылки данных в стандартный поток ввода: вместо этого используется переменная среды CONTENT_LENGTH для определения, какое количество данных необходимо считать из стандартного потока ввода. Рассмотрим на примерах отличие работы методов GET и POST.

Метод GET. Основная задача формы, как известно, это предоставление шаблона ввода данных, которые будут переданы скрипту. Сам скрипт при этом указывается через URL, который задается в атрибуте ACTION. Синтаксис:

<FORM ACTION=script.cgi> <INPUT NAME=n1 VALUE= "Поле1"> <INPUT NAME=n2 VALUE= "Поле2"> <INPUT VALUE= "Отправить"> </FORM>

При использовании данного метода скрипт просто распечатывает пару "имя поля формы - значение поля формы" в виде HTML-таблицы. Данный запрос из формы определяют как запрос типа URLENCODED, переданный по методу GET. При передаче значений по методу GET формируется только заголовок HTTP-сообщения и не формируется его тело. Поэтому все содержание полей формы помещается в URL и таким образом передается скрипту. Из текста скрипта видно, что данные извлекаются из переменной окружения QUERY_STRING, в которую сервер помещает запрос. Запросы, которые передаются в методе GET, можно условно разделить на два типа: ISINDEX и FORM-URLENCODED.

Метод POST. Очевидно, что в строку URL нельзя поместить бесконечное число символов. И броузер, и среда, в которой функционирует сервер, имеют ограничения либо, как в случае броузера, по длине поля location, либо по длине области переменных окружения. Поэтому для передачи относительно больших объемов предпочтительнее использовать метод доступа POST. Синтаксис:

```
<FORM METHOD=post ACTION=script.cgi>
<INPUT NAME=n1 VALUE= "Поле1">
<INPUT NAME=n2 VALUE= "Поле2">
<INPUT VALUE= "Отправить">
</FORM>
```

При использовании данного метода в контейнере FORM добавляется атрибут METHOD, который принял значение POST. Результат работы скрипта не изменился, но сам скрипт претерпел существенные изменения. Теперь запрос принимается со стандартного ввода, а не из переменной окружения QUERY_STRING.

При использовании метода POST данные передаются как тело HTTPсообщения, и скрипт читает их со стандартного ввода.

9.3.3 Атрибут ЕNCTYPE

Атрибут ENCTYPE задает тип кодировки данных пользователя из форм. Существует два типа кодирования содержания (тела) HTTPсообщения, которые можно определить в форме:

- application/x-www-form-urlencoded

- multipart/form-data

Первый тип кодирования выбирается по умолчанию и является основным способом. В URL документа можно использовать только символы набора Latin1. Это первая половина таблицы ASCII за вычетом первых 20 символов. Все остальные символы заменяются своими шестнадцатеричными эквивалентами /12/.

Кроме того, такие символы, как "+" или "&", играют роль разделителей. Если они встречаются в значении поля, то тоже заменяются на шестнадцатеричный эквивалент.

Второй тип применяется для передачи двоичной информации в теле HTTP-сообщения. Если проводить аналогию с электронной почтой, то multipart/form-data обеспечивает присоединение файла данных к HTTP-запросу. Наиболее типичным примером является передача файла с машины пользователя на сервер. Синтаксис:

<HTML> <HEAD> <BODY>

```
<FORM ACTION=script.cgi METHOD=post ENCTYPE=multipart/form-
data>
<INPUT NAME=n1 VALUE= "Поле1">
<INPUT NAME=n2 TYPE=file>
<INPUT VALUE="Отправить">
</FORM>
</BODY>
```

В данном случае HTTP-сообщение будет очень похоже на почтовое сообщение в стандарте MIME. Естественно, что для приема такого сообщения нужен скрипт, который бы смог разобрать его на части, а потом декодировать необходимую информацию.

9.4 Пример формы

Рассмотрим пример формы, с использованием всех рассмотренных выше элементов управления формами и применением всех атрибутов. Так здесь присутствует однострочное текстовое поле, переключатели, раскрывающийся список, флажки, многострочное текстовое поле, кнопки Submit и Reset.

```
<HTML>
<HEAD>
<TITLE>Пример использования форм
</TITLE>
</HEAD>
<BODY>
<FORM method = "post" action = "mailto: kis@ngasu.nsk.su ? subject =
Information" >
<PRE>
Ваше имя: <INPUT name="name" type= "text" maxlength="40" size="40"
value= "">
Ваш пол:
<INPUT type="radio" name= "pol" value= "male" checked>Мужской
<INPUT type= "radio" name= "pol" value="female">Женский
На каком курсе Вы учитесь?
<SELECT name= "curs" size= "1">
<option value= "1" selected>Первый</option>
<option value="2">Второй</option>
<option value= "3">Третий</option>
<option value= "4">Четвертый</option>
<option value= "5">Пятый</option>
</SELECT>
Какие языки программирования Вы знаете?
<INPUT type= "checkbox" name= "pascal" value= "yes" checked>Паскаль
<INPUT type= "checkbox" name= "c" value= "ves">Си
```

```
<INPUT type= "checkbox" name= "asm" value= "yes">Ассемблер
Напишите несколько слов о себе:
</PRE>
<TEXTAREA name= "about" rows= "6" cols= "60" wrap= "off">
</TEXTAREA>
<P>
<INPUT type= "submit" value= "Отправить данные">
<INPUT type= "reset" value= "Отправить данные">
</FORM>
</FORM>
</BODY>
</HTML>
```

Пример реализации данной HTML-страницы представлен на рисунке 95.

省 Пример использования форм - Microsoft Internet Explorer	
Файл Правка Вид Избранное Сервис Справка	1
🕞 Назад 🔹 🐑 🔹 🛃 🏠 🔎 Поиск 👷 Избранное 🤣 🔗 - 🌺	• •
Адрес: 🙋 ые технологии\ПОСОБИЕ ПО НТМL\Пример использования форм.html 💌 🎅 Переход	Ссылки »
Ваше имя: Ваш пол: • Мужской • Женский На каком курсе Вы учитесь? Первый • Какие языки программирования Вы знаете? • Паскаль • Си • Ассемблер Напишите несколько слов о себе:	
Отправить данныеОчистить форму	
🕙 Готово 🛛 😼 Мой компьютер	

Рисунок 95 - Пример реализации формы

Эта форма пытается отправить данные по указанному в заголовке адресу электронной почты mailto:kis@ngasu.nsk.su, используя тему письма "Information". Дальнейшие события зависят от того, установлена ли на машине пользователя почтовая программа-клиент. Если да, то информация будет отправлена, причем данные из полей формы придут в виде строк текста name=value, например, информация о курсе в виде curs=1. Если на машине нет программы электронной почты, Windows предупредит об этом.

10 Использование CGI при создании интерактивных интерфейсов

Формы были созданы и используются в WWW для получения отклика пользователя на предоставленную информацию и сбора данных о пользователе. После заполнения пользователем формы и запуска процесса ее обработки информация из нее попадает к программе, работающей на сервере. Для обработки большого количества откликов используются программы, поддерживающие Common Gateway Interface (CGI) и расположенные на сервере, в адрес которого поступают оклики. Механизм инициирования такой прикладной программы определен в спецификации CGI. Там же задан и порядок обмена данными между HTTP-сервером и программой, которая в спецификации CGI именуется скриптом. Таким образом, пользователь может интерактивно взаимодействовать с Web-сервером через Internet.

CGI (Common Gateway Interface, общий шлюзовой интерфейс) - спецификация, позволяющая Web-серверу выполнять произвольные прикладные программы. В результате работы таких программ (сценариев, или "скриптов") обрабатываются HTML-документы. С помощью CGI-сценариев могут приниматься данные от пользователя, они позволяют организовать диалог на Web-страницах, запросы к базам данных и т.д /13/.

В общем случае, интерактивный интерфейс пользователя представляет собой систему, обеспечивающую взаимодействие пользователя и программы. Для WWW, интерактивный интерфейс определить можно как **HTML-документов**, последовательность реализующих интерфейс классифицировать пользователя. Можно также условно принципы построения интерфейса по типу формирования HTML-документа:

- статический;

- динамический.

В первом случае источником интерфейса является HTML-документ, созданный в каком-либо текстовом или HTML-ориентированном редакторе. Следовательно, данный документ остается неизменным в течение использования. Во втором случае источником интерфейса является HTML-документ, сгенерированный сgi-модулем. Следовательно, появляется некоторая гибкость в видоизменении интерфейса во время использования.

Таким образом, можно ввести понятие интерактивного интерфейса для WWW. Интерактивный интерфейс для WWW представляет собой последовательность статических или динамически формируемых HTMLдокументов, реализующих интерфейс пользователя.

Практически любая задача, решающая проблему получения данных от клиента, связана с построением интерфейса. Наиболее интересным является построение интерфейсов к различным базам данных, доступ к SQL-серверу, получение информации от периферийных устройств, создание клиентских рабочих мест. Все это возможно посредством CGI.

Common Gateway Interface (CGI) является стандартом интерфейса внешней прикладной программы с WWW сервером.

Задача построения вышеназванных интерактивных интерфейсов делится на две части:

- клиентская часть;

- серверная часть.

Для создания клиентской части необходимо создать HTML-документ, в котором реализован интерфейс с пользователем. В языке HTML это возможно посредством форм.

Серверная часть состоит из исполняемого модуля, решающего основные задачи обработки данных поступающих от клиентской части, формирования ответа в формате HTML, и т.д. Такой модуль называется *cgi*-*модулем*.



Рисунок 96 - Две части интерактивного интерфейса

Для реализации взаимодействия "клиент-сервер" важно, какой метод HTTP запроса использует клиентская часть при обращении к WWW серверу. В общем случае, запрос - это сообщение, посылаемое клиентом серверу. Первая строка НТТР запроса включает в себя метод, который должен быть применен к запрашиваемому ресурсу, идентификатор ресурса (URI-Uniform Identifier). Resource используемую версию НТТР-протокола. И В рассматриваемом случае, клиентская часть применяет методы запроса POST и GET. Метод POST используется для запроса серверу, чтобы тот принял информацию, включенную в запрос, как относящуюся к ресурсу, указанному идентификатором ресурса. Метод GET используется для получения любой информации, идентифицированной идентификатором ресурса в НТТР запросе.

Для WWW-сервера прикладные программы или CGI-модули, обрабатывающие поток данных от клиента или (и) формирующие обратный поток данных могут быть написаны на таких языках программирования как:

– C/C++;

- Любой UNIX shell;
- Fortran;
- Perl;
- Visual Basic.

10.1 Спецификация CGI

CGI определяет четыре информационных потока:

- 1. Переменные окружения.
- 2. Стандартный входной поток.
- 3. Стандартный выходной поток.
- 4. Командная строка.



Рисунок 97 - CGI-интерфейс

10.1.1 Переменные окружения

Переменные окружения условно делятся на два типа:

- общие для всех типов запросов (устанавливаются для всех типов);
- зависящие от метода запроса.
- К переменным первого типа относятся следующие переменные:
- SERVER_SOFTWARE содержит информацию о WWW сервере (название/версия);
- SERVER_NAME содержит информацию об имени машины, на которой запущен Web-сервер, символическое имя или IP-адрес соответствующего URL;
- GATEWAY_INTERFACE содержит информацию о версии CGI (CGI/версия).

Следующие переменные являются специфичными для разных типов запросов и значения этим переменным присваиваются перед вызовом cgi-модуля:

- CONTENT_LENGTH значение этой переменной соответствует длине стандартного входного потока в символах;
- СОNTENT_TYPE эта переменная специфицирована для запросов содержащих дополнительную информацию, таких как HTTP POST и PUT, и содержит тип данных этой информации;
- SERVER_PROTOCOL эта переменная содержит информацию об имени и версии информационного протокола (протокол/версия);
- SERVER_PORT значение переменной содержит номер порта, на который был послан запрос;
- REQUEST_METHOD метод запроса, который был использован "POST", "GET", "HEAD" и т.д.;
- РАТН_INFO значение переменной содержит полученный от клиента виртуальный путь до сgi-модуля;
- РАТН_TRANSLATED значение переменной содержит физический путь до сді-модуля, преобразованный из значения РАТН_INFO;
- SCRIPT_NAME виртуальный путь к исполняемому модулю, используемый для получения URL;
- QUERY_STRING значение этой переменной соответствует строке символов следующей за знаком "?" в URL соответствующему данному запросу. Эта информация не декодируется сервером;
- REMOTE_HOST содержит символическое имя удаленной машины, с которой был произведен запрос. В случае отсутствия данной информации сервер присваивает пустое значение и устанавливает переменную REMOTE_ADDRESS;
- REMOTE_ADDRESS содержит IP-адрес клиента;
- AUTH_TYPE если WWW-сервер поддерживает аутентификацию пользователей и сgi-модуль является защищенным от постороннего доступа то, значение переменной специфицирует метод аутотентификации;
- REMOTE_USER содержит имя пользователя в случае аутотентификации;
- REMOTE_IDENT содержит имя пользователя, полученное от сервера;
- НТТР_АССЕРТ список типов МІМЕ известных клиенту. Каждый тип в списке должен быть отделен запятой согласно спецификации НТТР (тип/подтип, тип/подтип и т.д.);
- HTTP_USER_AGENT название программы просмотра которую использует клиент при посылке запроса.

10.1.2 Стандартный вывод

CGI-модуль выводит информацию в стандартный выходной поток. Этот вывод может представлять собой или документ, сгенерированный cgiмодулем, или инструкцию серверу, где получить необходимый документ. Обычно cgi-модуль производит свой вывод. Преимущество такого подхода в том, что cgi-модуль не должен формировать полный HTTP заголовок на каждый запрос.

В некоторых случаях необходимо избегать обработки сервером вывода cgi-модуля, и посылать клиенту данные без изменений. Для отличия таких cgi-модулей, CGI требует, чтобы их имена начинались на nph-. В этом случае формирование синтаксически правильного ответа клиенту cgi-модуль берет на себя.

Вывод cgi-модуля должен начинаться с заголовка содержащего определенные строки и завершаться двумя символами CR(0x10).

Любые строки не являющиеся директивами сервера, посылаются непосредственно клиенту. На данный момент, CGI спецификация определяет три директивы сервера:

1 Content-type. Используется Content-type MIME или тип возвращаемого документа.

Например: **Content-type**: text/html <CR><CR> сообщает серверу, что следующие за этим сообщением данные - есть документ в формате HTML.

2 Location. Указывает серверу, что возвращается не сам документ, а ссылка на него. Если аргументом является URL, то сервер передаст указание клиенту на перенаправление запроса. Если аргумент представляет собой виртуальный путь, сервер вернет клиенту заданный этим путем документ, как если бы клиент запрашивал этот документ непосредственно.

Например: Location: http://host/file.txt приведет к тому, что WWW сервер выдаст file.txt, как если бы он был затребован клиентом. Если сдімодуль возвращает ссылки на gopher сервер, например на gopher://gopher.ncsa.uiuc.edu/. Вывод будет следующий:

Location: gopher://gopher.ncsa.uiuc.edu/

3 Status. Задает серверу HTTP/1.0 строку-статус, которая будет послана клиенту в формате: nnn xxxxx

где: nnn - 3-х цифровой код статуса

ххххх - строка причины

Например: HTTP/1.0 200 OK

В данном случае, клиенту будет сообщено об успешном выполнении запроса.

10.1.3 Стандартный входной поток

В случае метода запроса POST данные передаются как содержимое HTTP –запроса и будут посланы в стандартный входной поток.

Данные передаются cgi-модулю в следующей форме:

name=value&name1=value1&...&nameN=valueN

где name - имя переменной, value - значение переменной, N - количество переменных.

На файловый дескриптор стандартного потока ввода посылается CONTENT LENGTH байт. Так же передает сервер cgi-модулю CONTENT TYPE (тип данных). Сервер не посылает символ конца файла после передачи CONTENT LENGTH байт данных или после того, как cgiмодуль их прочитает. Переменные окружения CONTENT LENGTH и СОNTENT ТҮРЕ устанавливаются в тот момент, когда сервер выполняет сgi-модуль. Таким образом, если в результате исполнения формы с аргументом тега FORM - METHOD="POST" сформирована строка данных firm=MMM&price=100023, установит то сервер значение CONTENT LENGTH равным 21 и CONTENT TYPE в application/x-wwwform-urlencoded, а в стандартный поток ввода посылается блок данных.

В случае метода GET, строка данных передается как часть URL. T.e. например http://host/cgi-bin/script?name1=value1&name2=value2.

В этом случае переменная окружения QUERY_STRING принимает значение name1=value1&name2=value2.

10.1.4 Аргументы командной строки

CGI-модуль в командной строке от сервера получает:

- остаток URL после имени cgi-модуля в качестве первого параметра (первый параметр будет пуст, если присутствовало только имя cgiмодуля),
- список ключевых слов в качестве остатка командной строки для скрипта поиска,
- чередующиеся имена полей формы с добавленным знаком равенства и соответствующих значений переменных.

Ключевые слова, имена и значения полей формы передаются декодированными и перекодированными так, что сgi-модуль в командной строке получит информацию без необходимости осуществлять дополнительные преобразования /14/.

10.2 Последовательность действий для обработки входных данных cgi-модуля для разных методов запроса GET и POST

Исходя из разницы методов запросов GET и POST, можно определить последовательность действий для обработки входных данных cgi-модуля для разных типов запросов.

Для метода GET:

- 1 Получить значение переменной QUERY_STRING.
- 2 Декодировать имена и их значения (учитывая, что все пробелы при декодировании сервером были заменены символом "+" и все символы с десятичным кодом больше 128 преобразованы в символ "%" и следующим за ним шестнадцатеричным кодом символа.).
- 3 Сформировать структуру соответствия "имя значение" для дальнейшего использования в сді-модуле.

Для метода POST:

- 1 Получить из стандартного входного потока CONTENT_LENGTH символов.
- 2 Декодировать имена и их значения (учитывая, что все пробелы при декодировании сервером были заменены символом "+" и все символы с десятичным кодом больше 128 преобразованы в символ "%" и следующим за ним шестнадцатеричным кодом символа.).
- 3 Сформировать структуру соответствия "имя значение" для дальнейшего использования в сgi-модуле.

Очевидно, что отличие только в источнике данных. Поэтому, в принципе, возможно создание единого модуля для методов POST и GET. Необходимо только добавить в начало проверку значения переменной REQUEST_METHOD для определения метода запроса. После формирования структуры "имя-значение" можно приступить к решению задач, ради которых, собственно, создавался сgi-модуль. Понятно, что задачи, решаемые cgi-модулем, могут быть очень разнообразными (получение и обработка почты, доступ к базам данных, гостевая книга и т.д.).

Следующим важным моментом является динамическое формирование cgi-модулем HTML-документа (оформление результата работы модуля). Например, таблицы выборки из базы данных.

Для этого cgi-модуль должен выдать в стандартный выходной поток заголовок состоящий из строки: Content-type: text/html. После этого заголовка можно давать любой текст в формате HTML.

10.3 Примеры сді-модулей

Рассмотрим простой пример формы на языке HTML использующую программу query.

```
<HTML>
<HEAD>
<TITLE>Пример использования CGI
</TITLE>
</HEAD>
<BODY>
```

```
<FORM ACTION="http://main.server.ru/cgi-bin/post-query"
METHOD="POST">
<В>Ввелите свое имя
<I>(Фамилию Имя Отчество)
</I>:
</B>
<br><INPUT name=RealName type=text size=40 maxlength=60</pre>
value="Петров Иван Сидорович">
<P>
Пол: <INPUT name=Sex type=Radio value= "Мужской" CHECKED>-
мужской <INPUT name=Sex type=Radio value= "Женский">-женский<P>
<INPUT name=Submit type=submit value= "Послать запрос">
<INPUT name=Reset type=reset value= "Сброс">
</FORM>
</BODY>
</HTML>
```

В броузере эта HTML-страница будет выглядеть следующим образом:

🕙 Пример использования CGI - Microsoft Internet Explo 🛛	_ 🗆 🔀
Файл Правка Вид Избранное Сервис Справка	1
🕞 Назад 👻 💿 – 💽 😰 🏠 🔎 Поиск 🤺 Избранн	** +0e
Адрес: 🖉 СОБИЕ ПО НТМL\формы\Пример CGI.html 💌 🄁 Переход	Ссылки »
Введите свое имя (Фамилию Имя Отчество): Петров Иван Сидорович Пол: • мужской • женский Послать запрос Сброс	
	~
🕙 Готово 🤤 Мой компьютер	

Рисунок 98 - Пример формы

После инициации формы путем нажатия кнопки "Послать запрос" Web-сервер обрабатывает поток данных от формы (заменяет все пробелы в именах и значениях на символ "+", заменяет все символы с десятичным кодом большим 128 на символ "%" и следующим за ним шестнадцатеричным кодом символа (например "И" в %С8)). Выходной поток примет следующий вид:

RealName=%CF%E5%F2%F0%EE%E2+%C8%E2%E0%ED+%D1%E8% E4%EE%F0 %EE%E2%E8%F7&Sex= %CC%F3%E6% F1%EA%EE%E9 &Submit= %CF%EE%F1%EB%E0%F2%FC +%E7%E0%EF%F0%EE%F1.
В момент передачи управления модулю post-query сервер присваивает значения переменным окружения и аргументам командной строки:

```
argc = 0. argv =
SERVER SOFTWARE = NCSA/1.5.1
SERVER NAME = iceman.cnit.nsu.ru
GATEWAY INTERFACE = CGI/1.1
SERVER PROTOCOL = HTTP/1.0
SERVER PORT = 80
REQUEST METHOD = POST
HTTP ACCEPT = image/gif, image/x-xbitmap,
image/jpeg, image/pjpeg,*/*
PATH INFO =
PATH TRANSLATED =
SCRIPT NAME = /cgi-bin/test-cgi
QUERY STRING =
REMOTE HOST = fwa.cnit.nsu.ru
REMOTE ADDR = 193.124.209.74
REMOTE USER =
AUTH TYPE =
CONTENT TYPE = application/x-www-form-urlencoded
CONTENT LENGTH = 142
     И на экране броузера это будет выглядеть следующим образом:
```

Query Results

You submitted the following name/value pairs:

RealName = Петров Иван Сидорович

Sex = Мужской

Submit = Послать запрос

Для демонстрации реализации формы с методом запроса GET воспользуемся той же самой формой, что и для метода POST и программой query. Для этого изменим значение атрибутов ACTION и METHOD в теге FORM.

<FORM action= "http://main.server.ru/cgi-bin/query" METHOD=GET>

После инициации формы сервер установит следующие значения для переменных окружения и аргументов командной строки:

```
argc = 0. argv is =
SERVER SOFTWARE = NCSA/1.5.1
SERVER NAME = iceman.cnit.nsu.ru
GATEWAY INTERFACE = CGI/1.1
SERVER PROTOCOL = HTTP/1.0
SERVER PORT = 80
REQUEST METHOD = GET
HTTP ACCEPT = image/gif, image/x-xbitmap,
image/jpeg, image/pjpeg, */*
PATH INFO =
PATH TRANSLATED =
```

```
SCRIPT_NAME = /cgi-bin/test-cgi
QUERY_STRING =
RealName=%CF%E5%F2%F0%EE%E2+%C8%E2%E0%ED+%D1%E8
%E4%EE%F0%EE%E2%E8%F7&Sex=%CC%F3%E6%F1%EA%EE%E9&Sub
mit=%CF%EE %F1%EB%E0%F2%FC+%E7%E0%EF%F0%EE%F1
REMOTE_HOST = fwa.cnit.nsu.ru
REMOTE_ADDR = 193.124.209.74
REMOTE_USER =
AUTH_TYPE =
CONTENT_TYPE =
CONTENT_LENGTH =
```

При этом, выходной поток от формы появился в значении переменной QUERY_STRING. Результат работы query полностью совпадает с результатом работы post-query.

11 Загрузка страницы на Web-сервер

Следующим шагом после создания своего Web-сайта является опубликование его на каком–либо Web-сервере в сети Internet. Наиболее популярным на сегодняшний день является Web-сервер <u>www.narod.ru</u>.

Для того, чтобы зарегистрировать и выложить свой Web-сайт в сети Internet, необходимо проделать следующие шаги.

В строке адрес броузера набрать <u>www.narod.ru</u>. Здесь необходимо зарезервировать имя для сайта. Для этого в строке необходимо набрать любое слово (окончание narod.ru добавится автоматически) и для регистрации имени нажать "Занять" (рисунок 99).

Мастерская	n <u>Novra</u>	Чаты	<u><u> </u></u>
	Постройте свой сайт за 60 с	екунд!	
<u>2ë</u>	Займите это имя для своего сайта .narod.1	и Занять	
	И получите в придачу неогранич дизайна, счетчики, статистику пос форум и чат.	е нное пространство , готовые вар зещений, гостевую книгу, персо	онанты ональный

Рисунок 99 – Ввод имени Web-сайта для регистрации

После этого появится диалоговое окно регистрации, (шаг 1 из 2), в котором необходимо ввести информацию о владельце Web-сайта (рисунок 98).

Регистрация: шаг 1 из 2
должен состоять из символов А-z, О-9, -, начинаться с буквы, заканчиваться буквой или цифрой и содержать не более 20 символов
Существует мнение, что все хорошие логины на Яндексе уже заняты. Но это не так. Мы можем посоветовать вам интересный незанятый логин. Для этого достаточно указать имя и фамилию (они понадобятся и для дальнейшей регистрации).
Яндекс охраняет персональные сведения пользователей в соответствии с <u>Соглашение</u>
Дальше

Рисунок 100 – Заполнение полей регистрации (шаг 1 из 2)

Здесь нужно пройти все пункты регистрации, которые предлагаются автоматически. При этом, в строку "Логин" ничего не надо вводить, потому что в предыдущей форме уже введено имя будущего сайта, а в форме 2 в строке "Логин" он уже будет введён. После этого ввести Имя, Фамилию и нажать кнопку "Далее". Появится диалоговое окно регистрации, шаг 2 из 2 (рисунок 101).

	Регистрация: шаг 2 из 2	
	Ваш логин - aparatka	
Пароль:		<u>Как правил</u>
	Пароль должен содержать не менее 4 симеолов из списка: А-z, 0-9, ! @#\$%^&*()_ может совпадать с логином.	-+ине
Подтвердите пароль:		
Контрольный вопрос:	не выбран	Если вы за получить д
Ответ:		90 MC 1963 1964
Электронная почта:	(если есть)	Вы сможет: работе со (адрес, на н подтвержде
Контрольные цифры:	введите, похалуйста, число, которое вы видите оправа	Если вы не контрольны вашем брау графики. Включите е заполните
	Нажимая кнопку <ok>, вы принимаете условия <mark>Пользовательского соглаш</mark></ok>	<u>Iehina</u> l OK

Рисунок 101 – Заполнение полей регистрации (шаг 2 из 2)

В этом окне необходимо ввести пароль, под которым будет храниться доступ к сайту, подтвердить пароль, т.е. ввести его ещё раз.

Бывают случаи, когда пользователи забывают пароль, под которым зарегистрирован сайт. Поэтому используется контрольный вопрос, который нужно ввести и ответ на него, например дата дня рождения, или имя любимого домашнего питомца.

Затем в строке "Электронная почта" ввести адрес электронного почтового ящика, на этот адрес будет выслано письмо с просьбой подтвердить регистрацию.

Затем появятся цифры, которые указаны на рисунке 101, они постоянно меняются. Их необходимо ввести и нажать ОК.

Последний этап регистрации имени под сайт. В последнем окне регистрации требуется ввести дополнительную информацию о себе. Нужно указать обращение, псевдоним, пол, дату рождения, страну, округ, город, сферу занятий.

После этого указать кодировку и формат писем, которые будут направляться службой поддержки. Затем необходимо нажать "Сохранить" (рисунок 102).

глобычыная воменнотерная сеть нактариеть	Регистрация успешно завершена.			
R	Теперь у вас на Яндексе есть <u>свой паспорт</u> . В нем хранятся ваши личнь предпочтения.			
	Вы можете начать пользоваться ј	<u>сервисами Яндекса</u> прямо сейчас.		
hactop	Просим вас найти минуту и рассказать немного о себе, чтобы Яндексу было легче в обслуживать, а вам — удобней пользоваться Яндексом.			
	Дополнительная инфо	рмация		
Обращение:	Не указано			
Псевдоним:				
Пол:	 предпочитаю не указывать мужской женский 			
Ваш день рождения:	января 💌			
Страна:	не указано			
Округ, город:				
Сфера моих занятий:	другое	•		
	Кодировка и формат писем, напр	авляемых вам службой поддержки:		
Кодировка:	KOIB-R			
Формат писем:	🔵 Текст 🖲 HTML			

Рисунок 102 – Последний этап регистрации

Затем на сайте narod.ru выбрать пункт "Загрузка файлов на сайт" (рисунок 103). Здесь необходимо указать путь к файлам с помощью кнопки "Обзор" и нажать кнопку "Загрузить файлы" (рисунок 104).



Рисунок 103 – Загрузка файлов на сервер

Для просмотра всех файлов, которые уже загружены, нужно выбрать "Управление файлами и HTML-редактор" (рисунок 105). Здесь отображены все файлы и папки, которые были выложены в сети Internet. Можно как создавать папки, так и перемещая файлы из папки в папку. Не стоит забывать о том, что если меняются расположение хотя бы одного файла, необходимо изменить в html-коде сайта путь к перемещаемому файлу (рисунок 106).

рузка файлог	в		
Загрузка в	папку «Корневая п	апка» <u>Выбра</u>	ать папку
	Обзор		Обзор





Рисунок 105 – Выбор просмотра выложенных на Web-сайт страниц

Яn	dex®		ППЦЦХ полуфинал Кубка Кремля — 17:21 НОВОСТИ 10:00 мск — автоматически обработано 993 источника.		
	W	<u>Novra</u>	Чаты	Помощь	Выход
Μ	астерская / Управление файлами				Проект: <u>Fla</u>
			<u>Создать папку</u> <u>Создать с</u>	траницу по шаблону Создать	<u>html-файп Загрузить файпы</u>
	Отмеченное: Копировать Переместить	Удалить			
	Имя	Описание	<u>Размер</u>	Дата	
	🗀 Папки				
	Download			27.07.2005	Свойства
	image			27.07.2005	Свойства
	🔋 Страницы				
	🖉 aboutus.htm		4316 B	27.07.2005	Свойства
	L home.htm		2554 Б	27.07.2005	Свойства
	🗷 <u>index.htm</u> (<u>0</u>)		5077 B	27.07.2005	Свойства
	Отмеченное: Копировать Переместить	Удалить			

Рисунок 106 – Просмотр выложенных на Web-сайт страниц

Заключение

В данном учебном пособии рассматриваются основы создания Webстраниц с помощью наиболее популярного на сегодняшний день языка разметки гипертекста – Hypertext Markup Language (HTML).

Материал изложен с нарастанием сложности. Сначала студенты изучают особенности правильной организации информации при планировании Web-сайта. После этого изучают историю развития и особенности языка HTML. Также рассматривается структура языка HTML, конструкция HTML-файла, основные теги, комментарии.

Затем студенты знакомятся со средствами форматирования текста в HTML, заголовками, абзацами. Рассматривают списки (нумерованные, маркированные, списки определений), таблицы, в частности, подробно рассматриваются атрибуты таблиц. Помимо этого учатся работать с горизонтальными линейками, бегущей строкой.

Также студенты знакомятся со способами организации гиперссылок, изучают относительные и абсолютные ссылки, закладки, также знакомятся с тем, как осуществляется вставка в страницу рисунков, как организуются ссылки из частей рисунка с помощью карт изображений, как добавляются интерактивные средства и эффекты мультимедиа. Помимо этого изучаются возможности работы с цветом в HTML. Рассматриваются способы задания цвета: символьно, в шестнадцатеричном или десятеричном формате.

Затем в пособии описаны способы создания фреймов, разбиение фреймсета на различное количество фреймов, рассмотрены плавающие фреймы. Изучаются ссылки между фреймами, закладки во фреймах.

Также в пособии рассматривается организация форм в HTML. Студенты изучают различные элементы управления формами, такие как однострочное текстовое поле, раскрывающийся список, флажок, прокручивающееся поле, переключатели, многострочное текстовое поле. Рассматриваются различные атрибуты форм, кодировка данных при передаче данных пользователя, способы сохранения данных из форм, методы, используемые при передаче данных CGI-модулю.

Помимо этого рассматривается возможность обработки данных пользователя из форм с помощью CGI-Common Gateway Interface. Заканчивается изложение материала в пособии описанием загрузки готового Web-сайта на сервер.

В приложениях студентам представлены контрольные вопросы для закрепления изученного материала. Также в приложениях присутствуют тесты. Здесь каждый вопрос теста содержит пять вариантов ответа, что наиболее усложняет прохождение тестирования студентами.

Таким образом, предлагаемое учебное пособие поможет студентам научиться разрабатывать общую структуру Web-сайтов на языке HTML, добавлять на HTML-страницы различные элементы дизайна и размещать готовые Web-сайты в сети Internet путем загрузки их на Web-сервер.

Список использованных источников

1 Основы Web-технологий / П.Б. Храмцов, С.А. Брик, А.М. Русак, А.И. Сурин; под редакцией П.Б. Храмцова. [Текст] – М.: ИНТУИТ.РУ "Интернет-Университет Информационных Технологий", 2003. – 512 с. – ISBN 5-9556-0001-9.

- 2 **Артемьев, В.И.** Разработка Intranet-приложений [Электронный ресурс], 1998. Режим доступа: WWW.URL: http://www.fizmat.vspu.ru./citforum/ internet/ intranet app/ interintr_01.shtml.
- 3 Вин, Д. Искусство Web-дизайна [Текст]/Д. Вин. -СПб. : Питер, 2003. 224 с. : ил.- (Самоучитель). ISBN 5-318-00402-4.
- 4 Шафран, Э. Создание Web-страниц [Текст]/ Э.Шафран. -СПб . : Питер, 2000. - 320с. : ил + CD-ROM.- (Самоучитель). - ISBN 5-314-00072-5.
- 5 **Коротеева, Е.** Введение в HTML [Электронный ресурс], 1995. Режим доступа: WWW.URL:http://books.kulichki.net/index.php.
- 6 Поданева, Ю. Спецификация HTML 4.0 [Электронный ресурс]. Режим доступа: http://bookslauliahlsi.net/dots/html/html2/acuer.html#too

http://bookskulichki.net/data/html/html2/cover.html#toc.

- 7 Введение в HTML. Учебник для вузов [Электронный ресурс]/П.Б. Храмцов, С.А. Брик, А.М. Русак, А.И. Сурин; под редакцией П.Б. Храмцова, 2003. Режим доступа: WWW.URL: http://www.intuit.ru.
- 8 **Пауэлл, Т.А.** Полное руководство по HTML [Текст]/Пер. с англ. А.В. Качанов. Мн.: ООО "Попурри", 2002. 912 с. ISBN 5-9556-0001-9.
- 9 Матросов, А.В. НТМL 4.0. Новый уровень создания НТМL-документов: наиболее полное руководство в подлиннике [Текст]/А.В. Матросов, А.О. Сергеев, М.П. Чаунин. – СПб.: БХВ - Петербург, 2001. – 672 с.:ил. – ISBN 5-8206-0072-Х.
- 10 **Гончаров, А.** Самоучитель HTML [Текст]/А. Гончаров. -СПб .: Питер, 2001. - 240 с. : ил.- (Самоучитель). - ISBN 5-272-00072-2.
- 11 **Кисленко,** Н. П. Основы HTML [Электронный ресурс]. Режим доступа: WWW.URL: http://books.kulichki.net/data/html/html7.
- 12 Журко, Ю.В. Учебник по HTML [Электронный ресурс]. Режим доступа: WWW.URL: http://books.kulichki.net/data/html/html3.
- 13 Введение в CGI. Учебник для вузов [Электронный ресурс]/ П.Б. Храмцов, С.А. Брик, А.М. Русак, А.И. Сурин; под редакцией П.Б. Храмцова, 2003. Режим доступа: WWW.URL: http://www.intuit.ru.
- 14 Пэтчетт, К. CGI/Perl: создание программ для Web/ К.Пэтчетт, М. Райт, BHV. Киев, 2000.
- 15 Штайнер, Г. HTML/XML/CSS: справочник/ Г. Штайнер [Текст]. М.: Лаборатория Базовых знаний, 2001. 512 с.: ил. ISBN5-93208-091-4.
- 16 Мальчук, Е.В. HTML и CSS. Самоучитель [Текст]. М.: Издательский дом "Вильямс", 2006. 416 с.- 5-8459-0907-4.

17 Гаевский, А.Ю. 100 % самоучитель по созданию Web-страниц и Webсайтов. HTML и JavaScript [Текст]. – М.: Технолоджи-3000, 2005. – 464 с. – ISBN 5-94472-027-1.

Приложение А *(справочное)* Контрольные вопросы для закрепления материала

- 1 Язык гипертекстовой разметки HTML. Особенности языка. HTML?
- 2 Что понимается под терминами "закрывающий" и "открывающий" теги? Всегда ли они оба применяются?
- 3 Каким тегом можно внести комментарий в HTML?
- 4 Каково назначение метаданных?
- 5 Какие теги используются для описания метаданных в HTML?
- 6 Какие существуют специальные теги в HTML?
- 7 Какие существуют элементы форматирования на уровне блоков в HTML?
- 8 Какие существуют элементы логического форматирования текста в HTML?
- 9 Каково назначение тега <CITE>?
- 10 Каково назначение тега <ABBR>?
- 11 Какие существуют элементы физического форматирования текста в HTML?
- 12 Каково назначение тега ?
- 13 Каково назначение тега <TT>?
- 14 Какие основные типы списков используются в HTML?
- 15 С помощью какого тега задаются маркированные списки в HTML?
- 16 С помощью какого тега задаются нумерованные списки в HTML?
- 17 С помощью какого тега задаются списки определений в HTML?
- 18 С помощью какого тега задается элемент в нумерованных и маркированных списках?
- 19 Назначение тега <HR>?
- 20 С помощью какого тега задается бегущая строка в HTML?
- 21 Какие теги позволяют управлять цветами в HTML-документе?
- 22 Какой тег используется при работе с таблицами в HTML?
- 23 Каким образом запретить разбиение данных в ячейке таблицы, т.е. представить единой строкой?
- 24 Как сделать, чтобы в таблице отсутствовали рамки?
- 25 С помощью какого атрибута можно задать цвет рамке в таблице?
- 26 С помощью какого атрибута в теге <TABLE> задается расстояние от содержимого ячейки до рамки?
- 27 Основное назначение атрибутов ROWSPAN и COLSPAN в таблицах?
- 28 С помощью какого атрибута задается фон таблицы, строк или ячеек в HTML?
- 29 Основное назначение атрибута CELLSPACING?

- 30 Можно ли использовать в качестве фона таблицы графические файлы?
- 31 Какой тег описывает гиперссылки?
- 32 Чем отличаются абсолютные и относительные ссылки?
- 33 Как указать ссылку на часть документа?
- 34 Может ли рисунок быть использован в качестве гиперссылки?
- 35 Может ли в качестве гиперссылки быть указана ссылка на электронный почтовый ящик?
- 36 Каким тегом можно вставить графику в Web-документ?
- 37 Как указать текст, который будет отображен броузером, не поддерживающим графику?
- 38 С помощью какого атрибута тега можно задать URL-адрес графического файла?
- 39 С помощью какого атрибута тега можно задать высоту изображения?
- 40 С помощью какого атрибута тега можно задать ширину изображения?
- 41 С помощью какого атрибута тега можно задать отступ от изображения сверху и снизу?
- 42 С помощью какого атрибута тега можно задать отступ от изображения слева и справа?
- 43 Основное назначение атрибута USEMAP тега ?
- 44 Основное назначение атрибута ISMAP тега ?
- 45 Каким тегом можно вставить анимацию в Web-документ?
- 46 Каким тегом можно вставить элемент мультимедиа в Webдокумент?
- 47 В чем отличие использования тегов <EMBED> и <OBJECT>?
- 48 Каким образом можно указать конкретное число повторений файлов мультимедиа?
- 49 С помощью какого тега можно вставить фоновый звук на HTMLстраницу?
- 50 Как указать программе просмотра, проигрывать файл автоматически?
- 51 Что такое стандарт МІМЕ?
- 52 Какие наиболее популярные и используемые MIME-типы в сети Internet?
- 53 Каково основное назначение карт изображения?
- 54 Какой тег определяет изображение как графическую карту, т.е. разбивает заданный рисунок на несколько активных областей?
- 55 С помощью какого атрибута тега <AREA> определяется форма активной области для карты изображения?
- 56 Какие три формы активных областей для карт изображения существуют?
- 57 С помощью какого атрибута тега <AREA> определяются координаты заданных активных областей?

- 58 С помощью какого атрибута тега <AREA> задается URL, с которой связывается активная область?
- 59 С помощью какого атрибута можно задать неактивные области карты изображений?
- 60 С помощью какого тега задается главный фреймсет в HTML?
- 61 С помощью какого тега задается конкретный фрейм?
- 62 Для чего используются атрибуты COLS и ROWS в теге <FRAMESET>?
- 63 В каких единицах задается разбиение окна броузера на несколько фреймов?
- 64 Каковы возможные значения параметра Target?
- 65 Какое значение атрибута Target позволяет открыть документ в новом окне?
- 66 Какое значение атрибута Target позволяет открыть документ в том же окне?
- 67 С помощью какого тега задаются плавающие фреймы?
- 68 Для чего нужны формы в HTML?
- 69 Какие основные элементы управления формой существуют?
- 70 Какие значения может принимать атрибут "METHOD" в теге <FORM>?
- 71 В чем основное отличие работы методов GET и POST при обработке данных пользователя из форм?
- 72 Для чего используется атрибут "ACTION" в теге <FORM>?
- 73 Для чего используется атрибут "ENCTYPE" в теге <FORM>?
- 74 Для чего используется тег INPUT?
- 75 Как можно задать однострочное текстовое поле с помощью элементов управления формами?
- 76 Как можно задать поле ввода пароля с помощью элементов управления формами?
- 77 Как можно задать переключатели с помощью элементов управления формами?
- 78 Что позволяет сделать атрибут SUBMIT тега INPUT?
- 79 Что позволяет сделать атрибут RESET тега INPUT?
- 80 Как можно задать флажки (checkbox) с помощью элементов управления формами?
- 81 Для чего используется тег TEXTAREA?
- 82 Как создать раскрывающийся список в HTML, используя элементы форм?
- 83 Каким образом можно сформировать список прокрутки?
- 84 Назначение атрибута "SELECT MULTIPLE"??
- 85 Каким образом в раскрывающимся списке задать поле выбранным по умолчанию?
- 86 Что такое CGI?
- 87 На каком языке могут создаваться CGI-программы?
- 88 Что такое переменные окружения?

89 В каком формате передаются данные из форм CGI-программе?

90 Что такое стандартный выходной поток?

Приложение Б (справочное) Тесты

1) Кто является родоначальником языка HTML?

а) Джон Десембер;

б) Ларри Робертс;

в) Дж. С. Ликлайдер;

г) Тим Бернс-Ли;

д) Боб Тейлор.

2) В каком году разработан язык HTML?

a) 1989;

б) 1983;

в) 1991;

г) 1995;

д) 1999.

3) Комментарии в НТМL обозначаются:

a) <!--

б) //

B) {

г) /* --*/

д) нет верного ответа.

4) Заголовок какого уровня в HTML наиболее важен?

a) <H1>;

б) <H2>;

- в) <H3>;
- г) <H4>;
- д) <H6>.

5) Заголовок какого уровня в HTML наименее важен?

- a) <H1>;
- б) <H2>;
- в) <H3>;
- г) <H4>;
- д) <H6>.

6) Какой тег в HTML задает элемент абзаца?

- a) <P>;
- б)
;
- в) <HR>;

г) <DIV>; д) <DIR>.

7) Какой тег в HTML используется для перевода строки?

a) <P>;

б)
;

в) <HR>;

г) <DIV>;

д) <DIR>.

8) Какой тег используется для структуризации HTML-документов в виде разделов?

a) <P>;

б)
;

в) <HR>;

г) <DIV>;

д) <DIR>.

9) Какой тег в HTML используется для задания горизонтальной линии? а) <P>;

δ)
;

в) <HR>;

г) <DIV>;

д) <DIR>.

10) Какой элемент логического форматирования текста предназначен для обозначения в документе цитаты из другого источника?

a) <CITE>...</CITE>; б) <CODE>...</CODE>; в) <DFN>...</DFN>; г) <ADDRESS>...</ADDRESS>; д) <PRE>...</PRE>.

11) Какой элемент логического форматирования текста предназначен для обозначения в документе определений?

a) <CITE>...</CITE>; б) <CODE>...</CODE>; в) <DFN>...</DFN>; г) <ADDRESS>...</ADDRESS>; д) <PRE>...</PRE>.

12) Какой элемент логического форматирования текста предназначен для обозначения в документе исходного кода программы?

a) <CITE>...</CITE>;

б) <CODE>...</CODE>;

B) <DFN>...</DFN>;
r) <ADDRESS>...</ADDRESS>;
д) <PRE>...</PRE>.

13) Какой элемент логического форматирования текста позволяет отобразить текст "как есть" (без форматирования), теми же символами и с тем же разбиением на строки?

a) <CITE>...</CITE>; б) <CODE>...</CODE>; в) <DFN>...</DFN>; г) <ADDRESS>...</ADDRESS>;

- д) <PRE>...</PRE>.
- 14) Какой тег отображает текст моноширинным шрифтом?
- a) <STRIKE>...</ STRIKE>;
- **δ**) <U>...</U>;
- в) ...;
- Γ) <TT>...</TT>;
- д) _{...}.

15) Какой тег сдвигает текст ниже уровня строки и выводит его шрифтом меньшего размера (нижний индекс)?

- a) <STRIKE>...</ STRIKE>;
- δ) <U>...</U>;
- в) ...;
- $\Gamma) < TT > \dots < /TT >;$
- д) _{...}.

16) Какой тег отображает начертание текста полужирным ?

- a) <STRIKE>...</ STRIKE>;
- б) <U>...</U>;
- в) ...;
- Γ) <TT>...</TT>;
- д) _{...}.

17) Какой тег отображает начертание текста подчеркнутым?

- a) <STRIKE>...</ STRIKE>;
- **δ**) <U>...</U>;
- в) ...;
- Γ) <TT>...</TT>;
- д) _{...}.

18) Какой тег отображает начертание текста перечеркнутым?

- a) <STRIKE>...</ STRIKE>;
- δ) <U>...</U>;

в) ...; г) <TT>...</TT>; д) _{...}.

19) Какой тег сдвигает текст выше уровня строки и выводит его шрифтом меньшего размера (верхний индекс)?

a) <<u>SUP</u>>...</<u>SUP</u>>;

- б) <HR>; в) ...; г) <TT>...</TT>;
- д) _{...}.

20) Какой тег задает начертание текста курсив?

- a) ^{...};
- δ) <I>...</I>;
- в) ...;
- Γ) <TT>...</TT>;
- д) _{...}.

21) Какой тег отвечает за форматирование нумерованных списков?

- a) UL;
- б) OL;
- в) DL;
- г) LI;
- д) DT.

22) Какой тег отвечает за форматирование ненумерованных (маркированных) списков?

- a) UL;
- б) OL;
- в) DL;
- г) LI;
- д) DT.

23) Какой тег отвечает за форматирование списков определений? а) UL;

- б) OL;
- в) DL:
- г) LI;
- д) DT.

24) Каким тегом обозначаются пункты в нумерованных и маркированных списках?

- a) UL;
- б) OL;

в) DL; г) LI; д) DT.

25) В списках определений каждый из терминов помещается в тег:

- a) DD;
- б) OL;
- в) DT.
- г) DL;
- д) LI;

26) С помощью какого атрибута в нумерованных списках задается схема нумерации для всего списка?

a) DISC;

- б) START;
- в) TYPE;
- г) CIRCLE;
- д) SQUARE.

27) С помощью какого атрибута в нумерованных списках задается, с какого значения начинать нумерацию списка?

- a) DISC;
- б) START;
- в) TYPE;
- г) CIRCLE;
- д) SQUARE.

28) С помощью какого атрибута в маркированных списках задается тип маркера для всего списка?

- a) DISC;
- б) START;
- в) TYPE;
- г) CIRCLE;
- д) SQUARE.

29) Какое значение должен принять тип маркера, чтобы отображаться в виде квадрата?

- a) DISC;
 δ) START;
 b) TYPE;
 r) CIRCLE;
- д) SQUARE.

30) Какое значение принимает тип маркера, чтобы отображаться в виде круга?

a) DISC; б) START; в) TYPE; г) CIRCLE; д) SQUARE.

31) Какое значение принимает тип маркера, чтобы отображаться в виде окружности?

a) DISC; б) START; в) TYPE; г) CIRCLE; д) SQUARE.

32) Можно ли использовать в качестве маркера в списках графический файл?

а) можно, только в броузере Internet Explorer 5,5 и выше;

б) можно только броузере Opera;

в) можно если это поддерживается обработчиками событий JavaScript;

г) можно всегда;

д) нельзя.

33) При использовании Ecaspe-последовательностей, каким образом можно задать левую угловую скобку?

a) <

б)

в) »

г) >

д) «.

34) При использовании Ecaspe-последовательностей, каким образом можно задать неразрывный пробел?

a) <

б)

в) »

г) >

д) «.

35) При использовании Ecaspe-последовательностей, каким образом можно задать правую угловую кавычку?

a) <

б)

в) »

- г) >
- д) «.

- 36) Какое задание цвета в HTML является неверным?
- a) BGCOLOR=rgb(238,130,238);
- б) BGCOLOR=Violet;
- в) BGCOLOR= "Blue";
- г) BGCOLOR= "#0000111100001111";
- д) BGCOLOR="#EE82EE".

37) Атрибут BGCOLOR используется:

- а) в теге ВОДУ для задания цвета фона Web-странице;
- б) в теге ВОДУ для задания цвета текста Web-страницы;
- в) в теге FONT для задания размера шрифта;
- г) в теге FONT для задания цвета текста;
- д) в теге FONT для задания гарнитуры шрифта.

38) Атрибут SIZE используется:

- а) в теге ВОДУ для задания цвета фона Web-странице;
- б) в теге ВОДУ для задания цвета текста Web-страницы;
- в) в теге FONT для задания размера шрифта;
- г) в теге FONT для задания цвета текста;
- д) в теге FONT для задания гарнитуры шрифта.
- 39) Атрибут ТЕХТ используется:
- а) в теге ВОДУ для задания цвета фона Web-странице;
- б) в теге ВОДУ для задания цвета текста Web-страницы;
- в) в теге FONT для задания размера шрифта;
- г) в теге FONT для задания цвета текста;
- д) в теге FONT для задания гарнитуры шрифта.

40) Атрибут COLOR используется:

- а) в теге ВОДУ для задания цвета фона Web-странице;
- б) в теге ВОДУ для задания цвета текста Web-страницы;
- в) в теге FONT для задания размера шрифта;
- г) в теге FONT для задания цвета текста;
- д) в теге FONT для задания гарнитуры шрифта.

41) Атрибут FACE используется:

- а) в теге ВОДУ для задания цвета фона Web-странице;
- б) в теге ВОДУ для задания цвета текста Web-страницы;
- в) в теге FONT для задания размера шрифта;
- г) в теге FONT для задания цвета текста;
- д) в теге FONT для задания гарнитуры шрифта.

42) Для чего нужны динамические кнопки?

а) динамические кнопки позволяют применить ряд эффектов, которые будут активизироваться при наведении указателя мыши посетителями сайта. Например, внешний вид кнопки может изменяться;

б) динамические кнопки позволяют использовать динамическое отображение страниц.

в) выполнять некоторые функции, например, запрашивать файл по FTP-протоколу;

г) служат для выделения текста;

д) нет верного ответа.

43) Бегущая строка поддерживается броузерами:

a) Internet Explorer;

б) Opera;

в) Mozilla;

г) Netscape;

д) Не поддерживается ни одним из перечисленных броузеров.

44) С помощью какого тега задается бегущая строка в HTML?

- a) <MARQUEE>;
- б) <HITCOUNTER>;
- B) <HOVERBUTTON>;
- Γ) < OBJECT>;
- д) <BANNER>.

45) При задании бегущей строки в HTML, какой атрибут определяет направление движения?

a) Direction;

б) Behavior;

в) Loop;

г) Scrollamount;

д) Scrolldelay.

46) При задании бегущей строки в HTML, какой атрибут задает характер движения строки?

a) Direction;

б) Behavior;

в) Loop;

г) Scrollamount;

д) Scrolldelay.

47) При задании бегущей строки в HTML, какой атрибут задает число повторения текста?

a) Direction;б) Behavior;в) Loop;

г) Scrollamount;

д) Scrolldelay.

48) При задании бегущей строки в HTML, какой атрибут задает определяет величину паузы между тактами перемещения текста в милисекундах?

a) Direction;

- б) Behavior;
- в) Loop;
- г) Scrollamount;
- д) Scrolldelay.

49) При задании бегущей строки в HTML, какой атрибут устанавливает длину в пикселях "прыжка текста" за один такт, при большом значении этого параметра текст движется рывками?

a) Direction;

- б) Behavior;
- в) Loop;
- г) Scrollamount;
- д) Scrolldelay.

50) С помощью какого тега задаются таблицы в HTML?

- a) <TABLE>....</TABLE>;
- б) <TR>...</TR>;
- в) <TD>...</TD>;
- Γ) <TH>...</TH>;
- д) <CAPTION>...</CAPTION>.

51) С помощью какого тега задаются строки таблиц HTML?

- a) <TABLE>....</TABLE>;
- б) <TR>...</TR>;
- в) <TD>...</TD>;
- Γ) <TH>...</TH>;
- д) <CAPTION>...</CAPTION>.

52) С помощью какого тега задаются столбцы таблиц в HTML?

- a) <TABLE>....</TABLE>;
- б) <TR>...</TR>;
- в) <TD>...</TD>;
- г) <TH>...</TH>;
- д) <CAPTION>...</CAPTION>.

53) С помощью какого тега задаются заголовки таблиц в HTML?

- a) <TABLE>....</TABLE>;
- б) <TR>...</TR>;

в) <TD>...</TD>; г) <TH>...</TH>; д) <CAPTION>...</CAPTION>.

54) С помощью какого тега задаются подписи таблиц в HTML? a) <TABLE>....</TABLE>; б) <TR>...</TR>; в) <TD>...</TD>; г) <TH>...</TH>; д) <CAPTION>...</CAPTION>.

55) Если в таблице будет задано два тега <TR> и три тега <TD>, то в броузере отобразится

а) две стоки, два столбца;

б) три строки, два столбца;

в) две строки, три столбца;

г) две стоки, пять столбцов;

д) пять сток, два столбца.

56) С помощью какого атрибута в таблицах задается объединение нескольких строк в одну?

a) ROWSPAN;
б) COLSPAN; **B) CELLPADDING;**г) CELLSPACING;
д) BGCOLOR.

57) С помощью какого атрибута в таблицах задается объединение нескольких столбцов в один?

a) ROWSPAN;
б) COLSPAN; **B) CELLPADDING;**г) CELLSPACING;
д) BGCOLOR.

58) Какой атрибут определяет ширину пустого пространства между содержимым ячейки и ее границами в таблицах?

a) ROWSPAN; б) COLSPAN; **в) CELLPADDING;** г) CELLSPACING; д) BGCOLOR.

59) Какой атрибут определяет ширину промежутков между ячейками таблицы?

a) ROWSPAN;
б) COLSPAN; **B) CELLPADDING;**г) CELLSPACING;
д) BGCOLOR.

60) В каком из приведенных ниже примерах, задается выравнивание содержимого ячейки таблицы по ее нижней границе?

a) ALIGN=bleedleft;

б) ALIGN=left;

в) VALIGN=bottom;

г) VALIGN=top;

д) VALIGN=middle.

61) В каком из приведенных ниже примерах, задается прижатие содержимого ячейки таблицы вплотную к левому краю?

a) ALIGN=bleedleft;

б) ALIGN=left;

в) VALIGN=bottom;

г) VALIGN=top;

д) VALIGN=middle.

62) В каком из приведенных ниже примерах, задается выравнивание содержимого ячейки таблицы по ее верхней границе?

a) ALIGN=bleedleft;

б) ALIGN=left;

в) VALIGN=bottom;

г) VALIGN=top;

д) VALIGN=middle.

63) В каком из приведенных ниже примерах, задается центровка содержимого ячейки таблицы по вертикали?

a) ALIGN=bleedleft;

- б) ALIGN=left;
- в) VALIGN=bottom;
- г) VALIGN=top;
- д) VALIGN=middle.

64) Что определяет атрибут BORDER у элемента разметки TABLE?

а) расстояние между ячейками;

б) расстояние от содержания до границы;

- в) ширину границы;
- г) вид границы;

д) нет верного ответа.

65) Какой вариант задания гиперссылки верный?

a) ;

б) ;

в) ссылка;

г) 1.html;

д) ссылка

66) Какой из примеров отображает абсолютную гиперссылку?

a) ссылка</А>;

б) ссылка;

в) ссылка</А>;

г) ссылка;

д) ссылка</А>.

67) Какой из приведенных примеров отображает ссылку на почтовый ящик?

a) ссылка</А>;

б) ссылка;

в) ссылка</А>;

г) ссылка;

д) ссылка</А>.

68) Какой из приведенных примеров не является относительной гиперссылкой?

a) ссылка;

б) ссылка</А>;

в) ссылка</А>;

г) ссылка</А>;

д) ссылка.

69) Какой из приведенных пример гиперссылок является ссылкой на место в документе (закладкой)?

a) ссылка;

б) ссылка</А>;

в) ссылка</А>;

г) ссылка</А>;

д) ссылка.

70) Какой пример отображает правильную ссылку на графический файл, активизируя рисунок?

a) ;

б) ;

в) ссылка<A/>

г)

д) ссылка

71) Какой вариант задания гиперссылки является неверным?

- a) ссылка</А>;
- б) ссылка</А>;
- в) ссылка;
- г) </А>;
- д) .

72) Какой из вариантов создания гиперссылки правильный?

a) Присылайте ваши отзывы;

6) ;

B) ;

г) </А>;

д) ссылка</А>.

73) Какая из приведенных ниже гиперссылок не является абсолютной?

a) ссылка;

б) ссылка;

B) ссылка ;

г) ссылка;

д) ссылка</А>.

74) Цвет, который будет использоваться при выводе на экран текста из еще не выбранных гипертекстовых ссылок?

a) ALINK;

- б) VLINK;
- в) LINK;
- г) BLINK;
- д) DLINK.

75) Цвет, который будет использоваться при выводе на экран активных гипертекстовых ссылок?

a) ALINK;δ) VLINK;

- в) LINK;
- г) BLINK;
-) DLINK,
- д) DLINK.

76) Цвет, который будет использоваться при выводе на экран уже посещенных гипертекстовых ссылок?

- a) ALINK;
- б) VLINK;
- в) LINK;
- г) BLINK;
- д) DLINK.

77) Какой атрибут тега <A> задает URL объекта, на который делается ссылка?

a) HREF;

б) NAME;

в) TARGET;

г) ID;

д) TITLE.

78) Какой атрибут тега <A> присваивает якорю имя, чтобы на него мог сослаться другой якорь?

a) HREF;

б) NAME;

в) TARGET;

г) ID;

д) TITLE.

79) Какой атрибут тега <A> определяет, в каком фрейме будет открываться указанный объект?

a) HREF;

б) NAME;

в) TARGET;

г) ID;

д) TITLE.

80) Какой атрибут тега <A> задает текст подписи для ссылок?

a) HREF;

б) NAME;

в) TARGET;

г) ID;

д) TITLE.

81) Какой атрибут тега <A> присваивает якорю идентификатор, чтобы на него мог сослаться другой якорь, списки стилей или сценарий?

a) HREF;

б) NAME;

- в) TARGET;
- г) ID;
- д) TITLE.

82) Какой атрибут тега <A> задает клавишу на клавиатуре, по которой будет активизироваться ссылка?

a) HREF;б) ACCESEKEY;в) TABINDEX;

- г) REL;
- д) REV.

83) Какой атрибут тега <A> задает порядок, по которому будут обходиться ссылки на странице при нажатии на клавишу ТАВ?

a) HREF;
б) ACCESEKEY;
в) TABINDEX;
г) REL;
д) REV.

84) Какой атрибут тега <A> задает отношение между объектом, на который делается ссылка и текущим объектом?

a) HREF;

б) ACCESEKEY;

в) TABINDEX;

г) REL;

д) REV.

85) Какой атрибут тега <A> задает отношение между текущим объектом и объектом, на который делается ссылка?

a) HREF;б) ACCESEKEY;

B) TABINDEX;

г) REL;

д) REV.

86) Что произойдет, если при использовании тега <A> во фреймах атрибут TARGET будет опущен?

а) файл по ссылке загрузится в новое окно;

б) файл по ссылке загрузится в том же фрейме, из которого на него ссылались;

в) файл по ссылке загрузится в родительское окно;

г) файл по ссылке загрузится в родительское окно;

д) файл по ссылке не загрузится.

87) Какой вариант задания рисунка в HTML является верным?

a) ;

б) ;

в) <PIC SRC="1.jpg">;

г) <PIC SCR="1.jpg">;

д) .

88) Какой из атрибутов тега позволяет указать текст, который будет выводиться вместо изображения броузерами, неспособными представлять графику?

a) ALT;

б) ALIGN;

в) BORDER; г) SRC; д) USEMAP.

89) Какой из атрибутов тега определяет толщину рамки вокруг изображения?

a) ALT; б) ALIGN; в) BORDER; г) SRC; д) USEMAP.

90) Какой из атрибутов тега указывает файл изображения и путь к нему?

a) ALT; б) ALIGN; в) BORDER; г) SRC; д) USEMAP.

91) Какой из атрибутов тега определяет изображение как графическую карту или карту изображения?

a) ALT; б) ALIGN; в) BORDER; г) SRC; д) USEMAP.

92) Какой из атрибутов тега определяет положение изображения относительно окружающего его текста?

a) ALT; б) ALIGN; в) BORDER; г) SRC; д) USEMAP.

93) Какой из атрибутов тега задает горизонтальное расстояние между вертикальной границей страницы и изображением, а также между изображением и огибающим его текстом?

a) ISMAP;δ) WIDTH;

- B) HEIGHT;
- г) VSPACE;
- д) HSPACE.

94) Какой из атрибутов тега задает вертикальное расстояние между строками текста и изображением?

- a) ISMAP;
- б) WIDTH;
- в) HEIGHT;
- г) VSPACE;
- д) HSPACE.

95) Какой из атрибутов тега задает целочисленное значение размера изображения по вертикали?

- a) ISMAP;
- б) WIDTH;
- в) HEIGHT;
- г) VSPACE;
- д) HSPACE.

96) Какое значение атрибута ALIGN выравнивает верх изображения по верхнему краю самого высокого элемента в строке окружающего текста?

a) ALIGN="left";δ) ALIGN="right";B) ALIGN="top";

 Γ) ALIGN="textop";

д) ALIGN="middle".

97) Какое значение атрибута ALIGN определяет огибаемое текстом изображение. Изображение располагается вдоль левой границы документа, а последующие строки текста огибают его справа?

a) ALIGN="left";

- б) ALIGN="right";
- в) ALIGN="top";
- г) ALIGN="textop";
- д) ALIGN="middle".

98) Какое значение атрибута ALIGN выравнивает нижний край изображения по базисной линии строки окружающего текста?

- a) ALIGN="absbottom";
- б) ALIGN="bottom";
- B) ALIGN="absmiddle";
- г) ALIGN="middle";
- д) ALIGN="top".

99) Что будет отображать броузер не поддерживающий изображения, или у которого выключено отображение графики?

a) диалоговое окно с запросом указания местоположения загружаемого изображения;

б) выведет содержимое атрибута ALT;

в) сообщение об ошибке;

г) выведет содержимое атрибута IMG;

д) на экране ничего не отобразится.

100) Что произойдет, если у изображения не указаны атрибуты WIDTH, HEIGHT?

а) изображение отобразится на всю ширину и высоту окна броузера;

б) отобразится в исходном размере;

в) отобразится в виде иконки;

г) отобразится надпись, указанная в атрибуте ALT;

д) изображение не отобразится.

101) В каком примере графическое изображение не вставится в HTMLдокумент?

a) <INPUT TYPE=image>;

б) ;

в) ;

г) <BODY BACKGROUND="1.gif">;

д) <TD BACKGROUND="1.bmp">.

102) Возможно ли использовать изображение в качестве гипертекстовой ссылки?

а) да, но только в броузере Internet Explorer;

б) нет, не возможно, это не поддерживается ни одним броузером;

в) возможно;

г) возможно, только файлы JPEG;

д) только небольшие файлы.

103) Что отобразится в окне броузера при использовании следующего HTML-кода ?

а) изображение, при активизации которого загружается графический файл;

б) изображение, при активизации которого загружается HTMLстраница;

в) указатель ссылки, при активизации которого загружается HTMLстраница;

г) указатель ссылки, при активизации которого загружается графический файл;

д) статическое изображение, не являющееся гиперссылкой.

104) Что отобразится в окне броузера при использовании следующего HTML-кода Текст?

а) изображение располагается вдоль левой границы документа, а последующие строки текста огибают его справа;

б) изображение располагается вдоль правой границы документа, а последующие строки текста огибают его слева;

в) изображение располагается вдоль правой границы документа, а последующие строки текста огибают его снизу;

г) изображение располагается вдоль правой границы документа, а последующие строки текста огибают его сверху;

д) изображение располагается вдоль правой границы документа, а последующие строки текста огибают его.

105) Какой вариант задания видео файла верен:

a) Мой видео файл

б) Мой видео файл

B) <A HREF= "My.avi"

г) <EMBED src="VideoFail.avi" width= "10" height= "20"></EMBED>

д) все варианты верны.

106) При использовании тега <OBJECT> какой атрибут содержит URL описание типа встраиваемого объекта?

- a) Classid;
- б) Codebase;
- в) Standby;
- г) Data;
- д) Align.

107) При использовании тега <OBJECT> какой атрибут задает URL встраиваемого объекта?

a) Classid;

б) Codebase;

- в) Standby;
- г) Data;

д) Align.

108) С помощью какого тега встраивается Flash-ролик в тело HTMLдокумента?

a) <EMBED>; б) <OBJECT>; в) <BGSOUND>; г) ; д) <BACKGROUND>.

109) При работе с тегом EMBED какой атрибут указывает, надо ли сразу проигрывать загружаемый файл?

- a) Pluginspage;
- б) Autostart;

в) Strarttime;г) Туре;д) Controls.

110) При работе с тегом EMBED какой атрибут указывает воспроизведение с указанного в минутах и секундах момента от начала файла?

a) Pluginspage;

б) Autostart;

в) Strarttime;

г) Туре;

д) Controls.

111) При работе с тегом EMBED какой атрибут URL Web-страницы, содержащей инструкции по инсталляции модуля?

a) Pluginspage;

б) Autostart;

в) Strarttime;

г) Туре;

д) Controls.

112) Что произойдет, если при использовании тега <BGSOUND> атрибуту LOOP задать значение LOOP= "-1"?

а) звуковой файл не будет проигрываться;

б) звуковой файл будет проигрываться бесконечно;

в) появится диалоговое окно с запросом числа повторений файла;

г) звуковой файл будет проигрываться без звука;

д) задаєтся десятикратное повторение воспроизведения звукового файла.

113) С помощью какого МІМЕ-типа задается видеоклип в формате AVI?

a) video/x-msvideo;

б) application/x-shockwave-flash;

в) audio/mpeg;

г) audio/basic;

д) application/x-zip-compressed.

114) Чтобы включить поддержку клиентской карты изображения, необходимо ввести тег:

a) ;

б) <BODY BACKGROUND= "picture.gif">;

B) .

г) .;

д) .

115) С помощью какого элемента определяются области карты изображения в рамках тега <MAP>?

a) <AREA>; б) <MAP>; в) ; г) <USEMAP>; д) <SHAPE>.

116) Какой атрибут тега <AREA> определяет URL, по которому будет совершаться переход, если будет выбрана данная область на графической карте?

a) COORDS; б) NOHREF; в) TARGET; г) HREF; д) SHAPE.

117) С помощью какого атрибута тега <AREA> можно определять области, за которыми не закрепляются никакие действия?

a) COORDS; б) NOHREF; в) TARGET; г) HREF; д) SHAPE.

118) Какой атрибут тега <AREA> задает конкретную фигуру (форму) на карте изображения?

a) COORDS; б) NOHREF; в) TARGET; г) HREF; д) SHAPE.

119) Какой атрибут тега <AREA> задает координаты формы на карте изображения?

a) HREF; б) SHAPE; в) COORDS; г) ID; д) TARGET.

120) Какой атрибут тега <AREA> задает метку, на которую могут ссылаться другие якоря, списки стилей и сценарии?

a) HREF;

б) SHAPE;

в) COORDS; г) ID; д) TARGET.

121) Какой атрибут тега <AREA> задает кадр, в котором будет открыта ссылка?

a) HREF; б) SHAPE; в) COORDS; г) ID; д) TARGET.

122) Какое значение атрибута SHAPE отображает область на карте изображения в виде неправильного многоугольника?

a) SHAPE="RECT";
б) SHAPE="CIRCLE";
в) SHAPE="POLY";
г) SHAPE="SQUARE";
д) SHAPE="DISC".

123) Какое значение атрибута SHAPE отображает область на карте изображения в виде окружности?

a) SHAPE="RECT";
б) SHAPE="CIRCLE";
в) SHAPE="POLY";
г) SHAPE="SQUARE";
д) SHAPE="DISC".

124) Какое значение атрибута SHAPE отображает область на карте изображения в виде прямоугольника?

- a) SHAPE="RECT";
- б) SHAPE="CIRCLE";
- B) SHAPE="POLY";
- г) SHAPE="SQUARE";
- д) SHAPE="DISC".

125) При задании фреймовой страницы можно ли использовать тег ВОДУ для описания тела фреймсета?

а) можно;

б) нельзя;

в) можно, если это поддерживается броузером, например Internet Explorer;

г) можно, если указать в теге BODY имена всех вызываемых фреймовых страниц;

д) можно, если указать в теге BODY имя основного фреймсета.

126) С помощью какого атрибута можно задать разделение фреймсета на горизонтальные подокна?

а) с помощью атрибута ROWS тега FRAMESET;

б) с помощью атрибута COLS тега FRAMESET;

в) с помощью атрибута SRC тега FRAME;

г) с помощью атрибута MARGINWIDTH тега FRAME;

д) с помощью атрибута SCROLLING тега FRAME.

127) С помощью какого атрибута можно задать разделение фреймсета на вертикальные подокна?

а) с помощью атрибута ROWS тега FRAMESET;

б) с помощью атрибута COLS тега FRAMESET;

в) с помощью атрибута SRC тега FRAME;

г) с помощью атрибута MARGINWIDTH тега FRAME;

д) с помощью атрибута SCROLLING тега FRAME.

128) В каком из приведенных ниже примеров представлен фреймсет в виде сетки страниц?

a) <FRAMESET ROWS= "50%,50%" COLS="20,30,50">;

б) <FRAMESET ROWS= "20,*" COLS="*,*,*">;

в) <FRAMESET ROWS="*,*" COLS="30%,30%,30%">;

г) <FRAMESET ROWS="*,*" COLS="30%,2*,50">;

д) <FRAMESET ROWS="50,* " COLS="*,50%,*">.

129) Какие из приведенных ниже примеров задания фреймсета не являются аналогичными?

a) <FRAMESET ROWS="300,300" COLS="200,200">;

б) <FRAMESET ROWS="50%,50%" COLS="50%,50%">;

B) <FRAMESET ROWS="*,*" COLS="*,*">;

г) <FRAMESET ROWS="50%,50%" COLS="50%,50%">;

д) <FRAMESET ROWS="*,50%" COLS="*,50">.

130) Как задается URL фреймовой страницы во фреймсете?

а) с помощью атрибута HREF тега FRAMESET;

б) с помощью атрибута LOCATION тега BODY;

в) с помощью атрибута SRC тега FRAME;

г) с помощью атрибута ROWS тега FRAMESET;

д) с помощью атрибута COLS тега FRAMESET.

131) Что будет отображать броузер не поддерживающий фреймовую структуру документов?

а) пустую страницу;

б) текст, заключенный в элементе NOFRAMES, если он указан;

в) первый, встреченный фрейм;

г) названия всех фреймов;

д) текст, заключенный в элементе IFRAMES, если он указан.

132) Какой атрибут тега <FRAME> задает горизонтальный отступ между содержимым фрейма и его границами?

a) SRC; б) NAME; в) MARGINWIDTH; г) MARGINHEIGHT; д) SCROLLING.

133) Какой атрибут тега FRAMESET определяет расстояние между фреймами в пикселах?

a) FRAMESPACING;

б) FRAMEBORDER;

в) BORDER;

г) NORESIZE;

д) BORDERCOLOR.

134) Как задаются плавающие фреймы?

a) с помощью тега <IFRAME>;

б) с помощью тега <FRAMEI>;

в) с помощью тега <FRAMEIN>;

г) с помощью тега <FRAMEOUT>;

д) с помощью тега <FRAMEALL>.

135) В плавающих фреймах с помощью какого значения атрибута align задается выравнивание нижней границы фрейма по подстрочной линии символов окружающего текста?

a) texttop;

б) top;

в) absbottom;

г) baseline;

д) middle.

136) В плавающих фреймах с помощью какого значения атрибута align задается выравнивание нижней границы фрейма по базовой линии окружающего текста?

a) texttop;

б) top;

в) absbottom;

г) baseline;

д) middle.

137) В плавающих фреймах с помощью какого значения атрибута align задается выравнивание воображаемой центральной линию окружающего текста по воображаемой центральной линии фрейма?

a) texttop;

б) top;

в) absbottom;

г) baseline;

д) middle.

138) В плавающих фреймах с помощью какого значения атрибута align задается выравнивание верхней границы фрейма по верхней границе окружающего текста?

a) texttop;

б) top;

в) absbottom;

г) baseline;

д) middle.

139) В плавающих фреймах с помощью какого значения атрибута align задается выравнивание верхней границы фрейма по надстрочной линии символов окружающего текста?

a) texttop;

б) top;

в) absbottom;

г) baseline;

д) middle.

140) Какое значение атрибута target определяет, что документ, полученный по ссылке, будет загружаться в тот же фрейм, из которого ссылаются?

a) "_blank";
б) "_self";
в) "_parent";
г) "_top";
д) "_name".

141) Какое значение атрибута target определяет, что документ, полученный по ссылке, будет загружаться в новом окне броузера?

a) "_blank";
δ) "_self";
B) "_parent";
Γ) "_top";

д) "_name".
142) Какое значение атрибута target определяет, что документ, полученный по ссылке, будет загружаться в родительский фрейм?

a) "_blank"; б) "_self"; в) "_parent"; г) "_top"; д) "_name".

143) Какое значение атрибута target определяет, что документ, полученный по ссылке будет загружаться в полное окно (закрывая все остальные)?

a) "_blank";
б) "_self";
в) "_parent";
г) "_top";
д) " name".

144) Что такое форма?

a) это инструмент, с помощью которого HTML-документ может послать некоторую информацию в некоторую заранее определенную точку внешнего мира;

б) вид форматирования текста;

в) способ отображения таблиц;

г) инструмент для работы с картами изображений;

д) нет верного ответа.

145) С помощью какого контейнера задается форма?

- a) FORM;
- б) INPUT;
- в) SELECT.
- г) TEXTAREA;

д) SUBMIT.

146) С помощью какого общего тега задаются многие элементы управления форма, такие как однострочное текстовое поле, флажок, прокручивающееся поле, переключатели?

a) TEXTAREA; б) SUBMIT

B) INPUT;

=) CELECT

г) SELECT.

д) FORM.

147) Каким образом можно создать однострочное текстовое поле?
a) <INPUT TYPE=TEXT>;
б) <TEXTAREA>;

B) <INPUT TYPE= TEXTAREA>;
г) <OPTION> .
д) <INPUT TYPE=CHECKBOX>.

148) С помощью какого атрибута задается размер видимой на экране части текстового поля?

a) NAME;
б) MAXLENGTH;
в) SIZE;
г) ТҮРЕ;
д) ТЕХТ.

149) Производится ли шифрование передаваемой информации элемента ввода <INPUT TYPE= PASSWORD>?

а) да;

б) нет;

в) данные шифруются, если это поддерживается броузером.

г) в зависимости от содержимого атрибута action;

д) в зависимости от используемого метода передачи данных.

150) Какой атрибут тега INPUT используется для организации невидимого поля?

a) < INPUT TYPE=hidden>;

б) <INPUT TYPE=PASSWORD>;

в) <INPUT TYPE=CHECKBOX>;

г) <INPUT TYPE=RADIO>;

д) <INPUT TYPE=SUBMIT>.

151) Какой атрибут тега INPUT используется для организации селектора?

a) < INPUT TYPE=hidden>;

б) <INPUT TYPE=PASSWORD>;

B) <INPUT TYPE=CHECKBOX>;

г) <INPUT TYPE=RADIO>;

д) <INPUT TYPE=SUBMIT>.

152) Если при использовании <СНЕСКВОХ> использовать одинаковые значения атрибута NAME, то какому тегу будет аналогична такая запись?

a) < INPUT TYPE=RESET>;

б) <INPUT TYPE=PASSWORD>;

B) <INPUT TYPE=CHECKBOX>;

г) <INPUT TYPE=RADIO>;

д) <INPUT TYPE=SUBMIT>.

153) Какой тег используется в НТМL для задания переключателей?

a) <INPUT TYPE= CHECKBOX>; б) <INPUT TYPE= RADIO>; B) <INPUT TYPE= BUTTON>; r) < INPUT TYPE=RESET>; д) <INPUT TYPE= IMAGE>.

154) Какой атрибут при использовании переключателей позволяет определить значение поля по умолчанию?

a) VALUE;

б) CHECKED;

в) NAME;

г) TYPE;

д) CHECK.

155) Какой атрибут тега <INPUT> позволяет создать кнопку?
a) <INPUT TYPE= SUBMIT>;
б) <INPUT TYPE=RADIO>;
в) <INPUT TYPE=BUTTON>;
г) < INPUT TYPE=RESET>;
д) <INPUT TYPE= IMAGE>.

156) Какой атрибут тега <INPUT> позволяет инициировать отправку данных серверу из формы и перезагрузку текущей страницы?

a) <INPUT TYPE=SUBMIT>; б) <INPUT TYPE=RADIO>; в) <INPUT TYPE= BUTTON>; г) < INPUT TYPE=RESET>; д) <INPUT TYPE=IMAGE>.

157) Какой атрибут тега <INPUT> позволяет выставить значения полей формы по умолчанию?

a) <INPUT TYPE=SUBMIT>; б) <INPUT TYPE=RADIO>; в) <INPUT TYPE=BUTTON>; г) < INPUT TYPE=RESET>; д) <INPUT TYPE=IMAGE>.

158) Какой тег используются для создания многострочного текстового поля?

a) <INPUT TYPE=TEXT>;

б) <TEXTAREA>;

B) <INPUT TYPE=TEXTAREA>;

 Γ) <OPTION>;

д) <INPUT TYPE=CHECKBOX>.

159) При создании многострочного текстового поля, с помощью какого атрибута задается ширина области ввода?

a) ROWS;

б) COLS;

в) WIDTH;

г) HEIGHT;

д) SIZE.

160) Какой тег используется для организации списка с прокруткой?
а) INPUT;
б) SELECT;
в) TEXTAREA;
г) OPTION;
д) SUBMIT.

161) С помощью какого атрибута элемента формы SELECT можно задать выбор нескольких значений из раскрывающегося списка в формах?

a) MULTIPLE;
б) SIZE;
в) SELECTED;
г) VALUE;
д) OPTION.

162) Для того, чтобы сформировать список прокрутки с числом видимых элементов в контейнере SELECT, какой нужно указать атрибут? a) MULTIPLE;

б) SIZE;

B) SELECTED;

г) VALUE;

д) OPTION.

163) С помощью какого атрибута элемента формы SELECT можно задать принудительный выбор элемента списка по умолчанию?

a) MULTIPLE;
б) SIZE;
в) SELECTED;
г) VALUE;
д) OPTION.

164) Какой атрибут элемента FORM описывает URL, который будет вызываться для обработки формы?

a) HREF;
δ) LOCATION;
b) ACTION;
г) ENCTYPE;

д) METHOD.

165) С помощью какого атрибута тега <FORM> указывается метод передачи данных пользователя из форм?

a) HREF;
б) LOCATION;
в) ACTION;
г) ENCTYPE;
д) METHOD.

166) При использовании какого метода информация из формы добавляется в конец URL, который был указан в описании заголовка формы. CGI-скрипт получает данные из формы в виде параметра переменной среды QUERY STRING?

a) GET; δ) HEAD;

- в) POST;
- г) PUT;
- д) TUCH.

167) Какой метод в HTML передает всю информацию о форме немедленно после обращения к указанному URL. CGI- программа получает данные из формы в стандартный поток ввода?

a) GET;

- б) HEAD;
- в) POST;
- г) PUT;
- д) ТАСН.

168) Какой атрибут ENCTYPE задает тип кодировки данных пользователя из форм?

a) HREF;
б) LOCATION;
в) ACTION;
г) ENCTYPE;
д) METHOD.

169) Что использует сервер для передачи данных об информационном запросе от сервера к шлюзу?

а) Поток стандартного вывода;

б) Командную строку и переменные окружения;

в) Командную строку;

г) Протокол НТТР;

д) Переменные окружения.

170) Куда помещаются данные из HTML-форм при вызове CGIпрограммы методом GET?

а) в тело НТТР-запроса;

б) в аргументы командной строки CGI-программы;

в) в переменную окружения QUERY_STRING;

г) в переменную окружения REMOTE_USER;

д) в стандартный поток ввода CGI-программы.

171) CGI-модули, обрабатывающие поток данных от клиента и формирующие обратный поток данных не могут быть написаны на таких языках программирования как:

a) C/C++;

б) Delphi;

в) Assembler;

г) Perl;

д) Visual Basic.

172) Значение какой переменной окружения соответствует длине стандартного входного потока в символах?

a) REMOTE_IDENT;
δ) QUERY_STRING;
b) HTTP_ACCEPT;
r) CONTENT TYPE;

д) CONTENT LENGTH.

173) Из какой переменной окружения можно узнать IP-адрес хоста, производящего запрос?

a) REMOTE_USER;
б) PATH_INFO;
в) REMOTE_ADDRESS.
г) QUERY_STRING;
д) HTTP_ACCEPT.

174) Из какой переменной окружения можно узнать броузер, который использует клиент для посылки запроса?

a) HTTP_USER_AGENT;

б) HTTP_ACCEPT;

B) REMOTE_ADDRESS.

г) REMOTE_USER;

д) QUERY_STRING.

175) Какими символами отделяются пары переменная=значение друг от друга в передаваемой строке?

а) \$; б) @;

в) &;
г) #;
д) %.