



Министерство общего и профессионального образования Ростовской области
Государственное бюджетное профессиональное образовательное учреждение
Ростовской области
«Каменский техникум строительства и автосервиса»

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ ЗАДАНИЙ**

**МДК 08.01 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНТЕРФЕЙСОВ
ПОЛЬЗОВАТЕЛЯ**

Специальность 09.02.07 Информационные системы и программирование

2022

Протокол цикловой комиссии математических,
общих естественнонаучных дисциплин и группы
«Информатика и вычислительная техника»
от «__» _____ 2022 г. № __

Протокол методического совета
от «__» _____ 2022 г. № __

Методические указания предназначены для оказания помощи обучающимся в выполнении практических заданий по МДК 08.01 Проектирование и разработка интерфейсов пользователя по специальности 09.02.07 Информационные системы и программирование.

Разработчик:

Миненкова Л.Н, преподаватель высшей квалификационной категории
ГБПОУ РО «КТСиА»

СОДЕРЖАНИЕ

Введение	4
1 Структура HTML-документа	6
2 Теги HTML	8
3 Основные типы данных HTML	11
4 Форматирование шрифта	14
5 Списки	16
6 Гиперссылки	20
7 Графика	22
8 Таблицы	23
9 Стиль в документах HTML. Внешние таблицы стилей.	26
Каскады таблиц стилей	
10 Фреймы	29
11 Формы	31
Список литературы	38

ВВЕДЕНИЕ

Выполнение обучающимся практических заданий по МДК 08.01 проводится с целью:

- систематизации и закрепления полученных теоретических знаний и практических умений по общепрофессиональным и специальным дисциплинам учебного плана;

- применения приобретенных знаний к комплексному решению конкретных производственных задач;

- углубления теоретических знаний в соответствии с заданной темой;

- формирования умений применять теоретические знания при решении поставленных вопросов;

- формирования умений использовать справочную, нормативную и правовую документацию, специальную литературу, справочники, ГОСТы, компьютерные программы и т.д.;

- развития творческой инициативы, самостоятельности, ответственности и организованности.

В ходе выполнения практических заданий проверяется сформированность общих и профессиональных компетенций, соответствующих виду деятельности Разработка дизайна веб-приложений.

Сегодня во все области человеческой деятельности вмешивается компьютерная сеть Internet. В большей или меньшей степени сеть входит в жизнь каждого: в повседневную жизнь (почта и новости, отдых и развлечения). А главное – в профессиональную деятельность. Web была создана для распространения гипертекстовых документов в привлекательном графическом формате, но теперь всеобщей заботой стало расширение ее функциональных возможностей. Пользователи стремятся персонифицировать содержимое страниц. Многие компании хотят, чтобы страницы Web могли извлекать информацию из баз данных и составлять отчеты в задаваемой пользователем форме.

Хранилище данных может иметь различную реализацию (реляционная многомерная БД, совокупность ODBC-источников данных), но предназначено для целей оперативной аналитической обработки данных. Доступ к хранилищу организуется подобно доступу к базе данных.

Базы данных не имеют прямой поддержки в Internet/Intranet. Доступ и ведение БД производится через серверные или клиентские приложения. При этом запросы на поиск и ввод данных оформляются в виде HTML-форм.

Программное обеспечение в виде клиентских и серверных приложений составляет важный ресурс. Гипертекст служит средством создания "меню", ссылающегося на разные общие и специальные прикладные программы. Наличие диалоговых свойств в HTML и интерфейса CGI позволяет строить Intranet-приложения с доступом к БД. Если используются традиционные статичные страницы гипертекста, то в ответ на запрос клиента Web-сервер передает страницу в формате HTML.

Для разработки Internet/Intranet-приложений используется широкий спектр языков и средств программирования:

- язык разметки гипертекста HTML для создания и интерпретации Web-структур;
- язык виртуальной реальности VRML для генерации и управления трехмерными сценами;
- язык Java и средства программирования мобильных приложений;
- языки сценариев для разработки интерактивных страниц.

Сущность HTML заключается в том, что после разметки документа и выделения его различных частей можно быть уверенным: документ будет красиво и правильно отображаться в любом браузере и на любом компьютере.

Не следует считать HTML языком компьютерной верстки или языком описания печати. Единственное, что умеет HTML. — классифицировать части документа и обеспечивать его правильное отображение в браузере. Это позволяет отображать документы на самых разных платформах.

1 СТРУКТУРА HTML-ДОКУМЕНТА

Структура HTML-документа позволяет использовать вложенные друг в друга контейнеры. Собственно, сам документ - это один большой контейнер который начинается с тага `<HTML>` и заканчивается тагом `</HTML>`:

`<HTML>` Содержание документа `</HTML>`

Контейнер HTML или гипертекстовый документ состоит из двух других вложенных контейнеров: заголовка документа (HEAD) и тела документа (BODY):

Любой HTML-документ содержит три основных обязательных раздела:

- HTML,
- HEAD,
- BODY.

Структура HTML-документа имеет вид:

```
<HTML>
  <HEAD>
    <TITLE>Простейший документ
  </TITLE>
  </HEAD>
  <BODY >
    Тело документа, отображаемое браузером
  </BODY>
</HTML>
```

Рассмотрим подробнее каждый из них.

Раздел HTML определяет специфику документа, содержание которого будет интерпретироваться браузером. Раздел описывается тегом-контейнером `<HTML></HTML>` и дает браузеру информацию о том, что документ разработан с помощью языка разметки HTML.

Раздел HEAD выполняет функцию рабочего заголовка HTML-документа. Он описывается парным тегом `<HEAD></HEAD>`. Внутри этого раздела располагается служебная информация, не отображаемая в главном окне HTML-документа. Рассмотрим теги, которые содержит раздел HEAD.

- ◆ Название документа `<TITLE>`. Парный тег `<TITLE></TITLE>` предназначен для указания имени созданному электронному документу.

Следует помнить, что под именем документа в данном случае имеется ввиду не файловое наименование, а визуальный заголовок HTML-страницы.

- ◆ Связь между документами <LINK>. Часто бывает так, что несколько документов структурно или функционально взаимосвязаны между собой. В этом случае используется тег <LINK>, который не является парным, а значит, не требует указания закрывающегося тега.
- ◆ Мета-определения <META>. Мета определения электронного документа описываются целым рядом параметров, входящих в состав непарного тега <META> и предназначенных для описания внутренних свойств HTML-файла. Все мета-определения, в сущности, имеют два основных типа данных: HTTP-EQUIV и NAME. Первый можно отождествить с заголовками протокола передачи гипертекстовых данных HTTP.
- ◆ Элементы <STYLE> и <SCRIPT>. В структуру раздела документа HEAD также могут входить теги-контейнеры <STYLE></STYLE> и <SCRIPT></SCRIPT>. Первый парный тег описывает стилевые шаблоны документа, второй – содержит код исполняемых сценариев (скриптов).

Раздел BODY является одним из самых важных компонентов любого HTML-документа, т.к. в нем располагается содержательная часть, которая выводится браузером на экран монитора пользователя. Раздел описывается парным тегом <BODY></BODY>, внутри которого размещается большинство существующих тегов HTML. Тег <BODY> имеет ряд параметров, которые условно можно разделить на четыре основные группы: параметры фона, границ документа, текста и гиперссылок.

- background - фоновый узор, указывает на адрес фонового изображения
- bgcolor - Указание фонового цвета
- bottommargin - устанавливает границу нижнего поля документа(в пикселях)
- leftmargin - отступ слева (в пикселях)

rightmargin -отступ справа (в пикселях)

topmargin - отступ сверху (в пикселях)

link - цвет ссылок

alink - цвет активных ссылок

vlink - цвет посещенных ссылок

text - цвет текста

bgproperties = «fixed» - фон не прокручивать

scroll - устанавливает наличие или отсутствие полосы прокрутки

(распознаётся только браузером Internet Explorer)

Таблица 1 Основные цвета, задаваемые в HTML приведены в таблице:

Название	Значение	Цвет
black	#000000	Черный
silver	#808080	Темно- серый
gray	#C0C0C0	Серый
white	#FFFFFF	Белый
maroon	#800000	Бордовый
red	#FF0000	Красный
purple	#800080	Фиолетовый
fuchsia	#FF00FF	Лиловый
green	#00F000	Зеленый
lime	#00FF00	Ярко-зеленый
olive	#808000	Оливковый
yellow	#FFFF00	Желтый
navy	#000080	Темно-синий
blue	#0000FF	Синий
teal	#008080	Сине-зеленый
aqua	#00FFFF	Бирюзовый

2 ТЕГИ HTML

Форматирование документа на языке HTML задается специальными дескрипторами (тегами). *Тегом* называется единица разметки — особый набор символов, имеющий в HTML особое значение. Их называют так, просто воспроизводя английское слово tag. Заключена эта команда в угловые скобки «<>» и «>». Например, теги могут выглядеть так: , и т.д.

Теги делятся на две категории. *Начальный тег* открывает действие некоторого эффекта, а *конечный тег* — отменяет его. *Конечный тег* всегда

выглядит как ключевое слово, перед которым стоит символ / ("косая черта", или «слэш»).

Существует две основных разновидности тегов: парные и непарные. Парный тег требует наличия закрывающего тега. Тэг называется закрывающим, так как он отменяет команду. При этом закрывающий тег отличается от открывающего тега наличием косой черты, например, /. Открывающий и закрывающий теги образуют контейнер, внутри которого находится информация. Непарный тег не имеет закрывающего тега.

Все теги HTML по их назначению и области действия можно разделить на следующие основные группы

- определяющие структуру документа;
- оформление блоков гипертекста (параграфы, списки, таблицы, картинки);
- гипертекстовые ссылки и закладки;
- формы для организации диалога;
- вызов программ.

Структура гипертекстовой сети задается гипертекстовыми ссылками. Гипертекстовая ссылка - это адрес другого HTML документа или информационного ресурса Internet, который тематически, логически или каким-либо другим способом связан с документом, в котором ссылка определена.

Для записи гипертекстовых ссылок в системе WWW была разработана специальная форма, которая называется Universe Resource Locator. Типичным примером использования этой записи можно считать следующий пример:

Этот текст содержит

```
<A HREF="http://polyn.net.kiae.su/altai/index.html">
```

```
гипертекстовую ссылку</A>
```

В приведенном выше примере тэг "A", который в HTML называют якорем (anchor), использует атрибут "HREF", который обозначает гипертекстовую ссылку (Hypertext Reference), для записи этой ссылки в форме URL. Данная ссылка указывает на документ с именем "index.html" в директории

"altai" на сервере "polyn.net.kiae.su", доступ к которому осуществляется по протоколу "http".

Гипертекстовые ссылки в HTML делятся на два класса: контекстные гипертекстовые ссылки и общие. Контекстные ссылки вмонтированы в тело документа, как это было продемонстрировано в предыдущем примере, в то время как общие ссылки связаны со всем документом в целом и могут быть использованы при просмотре любого фрагмента документа. Оба класса ссылок присутствуют в стандарте языка с самого его рождения, однако, первоначально наибольшей популярностью пользовались контекстные ссылки. Эта популярность привела к тому, что механизм использования общих ссылок практически полностью "атрофировался". Однако по мере стандартизации интерфейса пользователя и стилей представления информации разработчики языка снова вернулись к общим ссылкам и стремятся приспособить их к задачам управления этим интерфейсом. Справедливости ради, следует отметить, что общие гипертекстовые ссылки в большинстве браузеров не используются и не отображаются.

Таблица 2 Теги структурного (логического) форматирования:

<P>...</P>	Абзац
<CENTER>...</CENTER>	Центрирование
<BLOCKQUOTE>...</BLOCKQUOTE>	Цитата
<H1>...</H1>	Заголовки разных уровней. H1 – самый большой, H6 – самый маленький
<H2>...</H2>	
<H3>...</H3>	
<H4>...</H4>	
<H5>...</H5>	
<H6>...</H6>	
 	Переход на новую строку
<DIV>...</DIV>	Выравнивание
<PRE>...</PRE>	Переформированный текст
<HR>	Горизонтальная линейка
<ADDRESS>...</ADDRESS>	Подпись автора или адрес организации
<!--...-->	Комментарий

Таблица 3 Теги физического форматирования:

...	Жирный
<I>...</I>	Курсив
<U>...</U>	Подчеркнутый
<STRIKE>...</STRIKE> <S>...</S>	Перечеркнутый
_{...}	Нижний индекс
^{...}	Верхний индекс
<TT>...</TT>	Печатная машинка (моноширинный текст)
<BIG>...</BIG>	Увеличенный шрифт (на один пункт больше)
<SMALL>...</SMALL>	Уменьшенный шрифт (на один пункт меньше)
...	Тег, задающий размер, цвет и начертание шрифта

3 ОСНОВНЫЕ ТИПЫ ДАННЫХ HTML

Значения атрибутов в языке HTML могут иметь различные типы данных. Основными типами данных являются следующие:

◆ CDATA

Значения атрибутов типа CDATA являются цепочками символов, включая, возможно, специальные символы. При отображении переносы строк игнорируются, а символы возврата каретки и табуляции заменяются на пробелы. Также игнорируются начальные и конечные пробелы в значении атрибута.

◆ ID и NAME

Значения атрибутов типа ID и NAME должны начинаться с латинской буквы (A-Z, a-z) и состоять только из латинских букв (A-Z, a-z), цифр (0-9), дефисов, подчеркиваний, двоеточий и точек. Они отличаются тем, что NAME может содержать специальные символы, а ID не может.

◆ IDREF и IDREFS

Значения типа IDREF и IDREFS указывают на идентификаторы, т. е. на значения атрибута id других элементов. Значение типа IDREF — это единственный идентификатор, а значение типа IDREFS — это список идентификаторов, разделенных пробелами.

- ◆ **NUMBER**

Числовые значения атрибутов — это десятичные числа, состоящие хотя бы из одной цифры (0-9).

- ◆ **Текст**

Текстовые значения атрибутов — это значения типа CDATA, представляющие собой осмысленный текст.

- ◆ **URI**

Значения атрибутов типа URI — это унифицированные идентификаторы ресурсов (полные или относительные).

- ◆ **Цвет**

Цветовые значения атрибутов могут задаваться либо шестнадцатеричным числом с префиксом «#» вида «#rrggbb», задающим RGB-код цвета, либо одним из 16-ти символических имен.

- ◆ **Пиксели**

Значения этого типа — целые числа, задающие количество пикселей.

- ◆ **Длина**

Значения этого типа — либо целые числа, задающие количество пикселей, либо доли в процентах от размера по горизонтали или вертикали: например, 50% означает половину всего размера, а 50 означает 50 пикселей.

- ◆ **Кратный размер (мульти-длина)**

Кратный размер — это либо целое число, задающее количество пикселей, либо доля в процентах от размера по горизонтали или вертикали, либо относительный размер вида i^* , где i — целое число. При распределении пространства обозреватель сначала выделяет место для размеров, заданных в числах и процентах, а затем разделяет оставшееся пространство между элементами с относительными размерами. Элементу размером 3^* будет выделено пространство в три раза большее, чем элементу размером 1^* .

- ◆ **Типы содержимого (типы MIME)**

Значения атрибутов этого типа задают типы файлов MIME для связанных

или вложенных ресурсов. Примеры типов содержимого включают «text/html», «image/png», «image/gif», «video/mpeg», «audio/basic», «text/tcl», «text/javascript» и «text/vbscript».

- ◆ **Код языка**

Значения атрибутов этого типа задают код языка. Например, ru – Русский.

- ◆ **Кодировки символов.**

Значения этого типа задают имя таблицы кодировки символов.

- ◆ **Отдельные символы**

Значение символьного атрибута — это единственный символ Unicode, который может быть также специальным символом.

- ◆ **Дата и время.**

Значения атрибутов этого типа задают дату и время в формате ГГГГ-ММ-ДДТЧ:ММ:ССУВЗ, где ГГГГ — четырехзначный номер года, ММ — двузначный номер месяца (01 — 12), ДД — двузначный номер дня (01 — 31), ЧЧ — двузначный номер часа (00 — 23), ММ — двузначный номер минуты (00 — 59), СС — двузначный номер секунды (00 — 59) и УВЗ — указатель временной зоны. Отметим, что буква Т, отделяющая дату от времени — это именно прописная латинская буква Т. Если какой-либо из компонентов времени неизвестен, то используется 00.

- ◆ **Типы ссылок**

Значения атрибутов этого типа представляют собой список типов ссылок, разделенных пробелами.

- ◆ **Дескрипторы носителей**

Значения атрибутов этого типа представляют собой список имен устройств для отображения документа, разделенных пробелами.

- ◆ **Сценарий**

Значения атрибутов этого типа — это сценарии клиента, которые обычно представляют собой вызов функции или несколько операторов на интерпретируемом языке. Синтаксис сценария определяется синтаксическими правилами соответствующего языка программирования.

◆ **Таблица стилей**

Значения атрибутов этого типа — таблицы стилей на языке CSS.

◆ **Целевые имена кадров**

За исключением приведенных ниже зарезервированных имен (таблица 3.4), целевые имена кадров должны начинаться с алфавитных символов (a-z, A-Z). Агенты пользователей должны игнорировать все остальные имена.

4 ФОРМАТИРОВАНИЕ ШРИФТА

HTML допускает два подхода к шрифтовому выделению фрагментов текста. С одной стороны, можно прямо указать, что шрифт на некотором участке текста должен быть **жирным** или *наклонным*, то есть изменить **физический стиль** текста. С другой стороны, можно пометить некоторый фрагмент текста как имеющий некоторый отличный от нормального **логический стиль**, оставив интерпретацию этого стиля браузеру. Поясним это на примерах.

Физические стили

Под физическом стилем принято понимать прямое указание браузеру на модификацию текущего шрифта. Например, все, что находится между метками `` и ``, будет написано **жирным шрифтом**. Текст между метками `<I>` и `</I>` будет написан *наклонным шрифтом*.

Несколько особняком стоит пара меток `<TT>` и `</TT>`. Текст, размещенный между этими метками, будет написан шрифтом, имитирующим пишущую машинку, то есть имеющим фиксированную ширину символа.

Логические стили

При использовании логических стилей автор документа не может знать заранее, что увидит на экране читатель. Разные браузеры толкуют одни и те же метки логических стилей по-разному. Некоторые браузеры игнорируют некоторые метки вообще и показывают нормальный текст вместо выделенного логическим стилем. Вот самые распространенные логические стили.

` ... `

От английского emphasis — акцент.

**... **

От английского strong emphasis — сильный акцент.

<CODE> ... </CODE>

Рекомендуется использовать для фрагментов исходных текстов.

<SAMP> ... </SAMP>

От английского sample — образец. Рекомендуется использовать для демонстрации образцов сообщений, выводимых на экран программами.

<KBD> ... </KBD>

От английского keyboard — клавиатура. Рекомендуется использовать для указания того, что нужно ввести с клавиатуры.

<VAR> ... </VAR>

От английского variable — переменная. Рекомендуется использовать для написания имен переменных.

Пример:

Подытожим наши знания о логических и физических стилях с помощью примера:

```
<html>
<head>
<title>Пример 5</title>
</head>
<body>
<H1>Шрифтовое выделение фрагментов текста</H1>
<P>Теперь мы знаем, что фрагменты текста можно выделять
<B>жирным</B> или <I>наклонным</I> шрифтом. Кроме того,
можно
включать в текст фрагменты с фиксированной шириной символа
<TT>(имитация пишущей машинки)</TT></P>
<P>Кроме того, существует ряд логических стилей:</P>
<P><EM>EM - от английского emphasis - акцент </EM><BR>
<STRONG>STRONG - от английского strong emphasis - сильный
акцент </STRONG><BR>
<CODE>CODE - для фрагментов исходных текстов</CODE><BR>
<SAMP>SAMP - от английского sample - образец </SAMP><BR>
<KBD>KBD - от английского keyboard - клавиатура</KBD><BR>
<VAR>VAR - от английского variable - переменная </VAR></P>
```

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

5 СПИСКИ

HTML позволяет определять внешний вид целых абзацев текста. Абзацы можно организовывать в списки, выводить их на экран в отформатированном виде, или увеличивать левое поле.

С помощью HTML можно создать 3 вида списков:

- ◆ Нумерованные
- ◆ Маркированные
- ◆ Списки определений

Маркированные списки: ...

Текст, расположенный между метками и , воспринимается как маркированный (нумерованный) список. Каждый новый элемент списка следует начинать с метки . Например, чтобы создать вот такой список:

- Иван;
- Данила;
- белая кобыла

необходим вот такой HTML-текст:

```
<UL>  
<LI>Иван;  
<LI>Данила;  
<LI>белая кобыла  
</UL>
```

Обратите внимание: у метки нет парной закрывающей метки.

В качестве маркеров могут быть использованы различные символы.

Таблица 4 Маркеры:

Атрибут	Значение	Описание
TYPE	circle	Задаёт тип маркера
	square	
	disc	

Пример:


```

HTML- код
<UL TYPE=" circle">
<LI TYPE="
square">текст
<LI TYPE="
disc">текст
<LI>текст
.....
</UL>

```

Вид в окне браузера

- Текст
- Текст
- текст

Нумерованные списки: ...

Нумерованные списки устроены точно так же, как ненумерованные, только вместо символов, выделяющих новый элемент, используются цифры. Если слегка модифицировать наш предыдущий пример:

```

<OL>
<LI>Иван;
<LI>Данила;
<LI>белая кобыла
</OL>

```

получится вот такой список:

1. Иван;
2. Данила;
3. белая кобыла

Таблица 5 Основные атрибуты тега приведены в таблице:

атрибут	значение	описание
COMPACT	-	Ввод списка в более компактном виде (часто игнорируется браузерами)
START	3	Значение (число), с которого начинается нумерация
TYPE	A,a,I,i,1	Задаёт тип номера
VALUE	Всегда число	Переопределяет номер одного элемента списка

Пример:

```

HTML-код
<OL > START="2"
TYPE="i"
<LI >текст
<LI>текст
.....
</OL>

```

Вид в окне браузера

b)текст
c)текст

Списки определений: <DL> ... </DL>

Список определений несколько отличается от других видов списков. Вместо меток в списках определений используются метки <DT> (от английского definition term — определяемый термин) и <DD> (от английского definition definition — определение определения). Разберем это на примере. Допустим, у нас имеется следующий фрагмент HTML-текста:

```
<DL>
<DT>HTML
<DD> Термин HTML (HyperText Markup Language) означает язык
маркировки гипертекстов. Первую версию HTML разработал сотрудник
Европейской лаборатории физики элементарных частиц Тим Бернерс-Ли.
<DT>HTML-документ
<DD>Текстовый файл с расширением *.htm (Unix-системы могут
содержать файлы с расширением *.html).
</DL>
```

Этот фрагмент будет выведен на экран следующим образом:

HTML

Термин HTML (HyperText Markup Language) означает 'язык маркировки гипертекстов'. Первую версию HTML разработал сотрудник Европейской лаборатории физики элементарных частиц Тим Бернерс-Ли.

HTML-документ

Текстовый файл с расширением *.htm (Unix-системы могут содержать файлы с расширением *.html).

Обратите внимание: точно так же, как метки , метки <DT> и <DD> не имеют парных закрывающих меток.

Если определяемые термины достаточно коротки, можно использовать модифицированную открывающую метку <DL COMPACT>. Например, вот такой фрагмент HTML-текста:

```
<DL COMPACT>
<DT>А
<DD>Первая буква алфавита
<DT>Б
<DD>Вторая буква алфавита
<DT>В
<DD>Третья буква алфавита
</DL>
```

будет выведен на экран примерно так:

- А
 Первая буква алфавита
- Б
 Вторая буква алфавита
- В
 Третья буква алфавита

Вложенные списки

Элемент любого списка может содержать в себе целый список любого вида. Число уровней вложенности в принципе не ограничено, однако злоупотреблять вложенными списками все же не следует.

Вложенные списки очень удобны при подготовке разного рода планов и оглавлений.

Наши знания о списках можно вкратце свести в следующем примере:

```
<html>
<head>
<title>Пример </title>
</head>
<body>
<H1>HTML поддерживает несколько видов списков </H1>
<DL>
<DT>Ненумерованные списки
<DD>Элементы ненумерованного списка выделяются специальным
символом и отступом слева:
<UL>
<LI>Элемент 1
<LI>Элемент 2
<LI>Элемент 3
</UL>
<DT>Нумерованные списки
<DD>Элементы нумерованного списка выделяются отступом слева, а
также нумерацией:
<OL>
<LI>Элемент 1
<LI>Элемент 2
<LI>Элемент 3
</OL>
<DT>Списки определений
<DD>Этот вид списков чуть сложнее, чем два предыдущих, но и
выглядит более эффектно.
```

<P>Помните, что списки можно встраивать один в другой, но не следует закладывать слишком много уровней вложенности. </P>
<P>Обратите внимание, что внутри элемента списка может

находиться

несколько абзацев. Все абзацы при этом будут иметь одинаковое левое поле. </P>

</DL>

</body>

</html>

Текст с отступом: <BLOCKQUOTE> ... </BLOCKQUOTE>

Текст, заключенный между метками <BLOCKQUOTE> и </BLOCKQUOTE>, выводится браузером на экран с увеличенным левым полем.

6 ГИПЕРССЫЛКИ

Гиперссылка (или просто ссылка) создается с помощью тега <A>...., так называемого якоря. HTML позволяет создать ссылку на веб-страницу, расположенную в любой точке мира. Таким образом все веб-страницы объединяется в одно глобальное пространство под названием WEB. Использование ссылок возможно потому что любая веб- страница в WEB имеет свой уникальный адрес, который называется унифицированный указатель ресурса. URL- адрес веб-страницы в якоря задает атрибут HREF. Текст находящийся внутри контейнера <A>...., превращается в ссылку.

Пример: Создание ссылки на сайт ИМПЭ им. А.С Грибоедова

ИМПЭ

Если веб-страница, на которую нужно сделать ссылку, расположена на том же сервере, можно использовать абсолютный или относительный адрес.

Абсолютный

Первая страница

Относительный

Если файл, на который создаем ссылку, расположен в той же папке (на том же уровне):

Первая страница

Если файл, на который создаем ссылку, расположен в другой папке. Например, на один уровень ниже в папке Images:Я в детстве

Если файл, на который создаем ссылку, расположен в другой папке, на один уровень выше в папке 3kurs:

Пробная страница

Веб-страница, на которую переходим по ссылке, может открываться в том же окне, может создавать новое окно. При использовании фреймов, когда в окне браузера открываются одновременно несколько веб-страниц можно перейти в любой из фреймов или загрузить страницу поверх всех фреймов. За направление перехода по ссылке отвечает атрибут TARGET.

Если нужно сделать ссылку внутри одной веб-страницы, то необходимо сначала создать закладку в том месте, куда будет осуществляться переход.

Пример: Создание ссылки внутри одной веб-страницы.

Первая страница

...

а потом саму ссылку в любом месте веб-страницы:.

...

Переход к первой закладке

Протокол mailto

Протокол mailto открывает окно почтового клиента, в котором можно создать и отправить почтовое сообщение по указанному адресу, если почтовый клиент настроен:

<A HREF=<mailto:wasya@mail.ru> > Пишите мне письма

Таблица 6 Атрибуты тега <A>...:

Атрибуты	Значение	Описание
HREF	www.iile index.html #metka	Задаёт адрес перехода по ссылке
NAME	metka	Создаёт на странице закладку-место, к которому можно перейти по ссылке.
TARGET	_self	Открыть в том же окне или фрейме
	_top	Открыть в том же окне поверх всех фреймов

	_blank	Создать новое окно и открыть в нем
	_parent	Открыть в родительском фрейме
	Имя фрейма	Открыть во фрейме "имя фрейма"
TITLE	Текст	Текст- подсказка, виден при наезде мышкой

7 ГРАФИКА

На веб-странице можно разместить фотографии, простые и анимированные рисунки. В Интернете в основном используют два формата для вывода на экран графической информации: JPG и GIF. Формат JPG применяют для вставки фотографий и веб-страницу. При малом объёме файла сохраняется качество фотоснимка с минимальными потерями. Формат Gif – для рисунков, поддерживает анимацию и прозрачность.

Вставить изображение в текст веб-странице можно с помощью тега , в котором атрибуту SRC присваивается URL – адрес картинка (аналогична HREF в якорях) , - одиночный, у него нет закрывающего тега.

Изображение лучше хранить в отдельной папке, чтобы отличать графическую информацию от текстовой.

Пример:

Таблица 6 Атрибуты тега :

Атрибут	Значение	Описание
SRC	Photo. jpg	Задаёт URL картинка
ALT	Фото	Всплывающие подсказка, появляющийся при наезде мышкой. Текст виден, когда отключен показ рисунков в браузере.
ALIGEN	Left.right.cent	Выравнивание
WIDTH	100,50%	Ширина изображения. Без её указания картинка выводиться в натуральную величину.
HEIGHT	100,50%	Высота изображения. Без её указания картинка выводиться в натуральную величину.
BORDER	5	Рамка вокруг рисунка
HSPACE	0	Свободное пространство слева и справа от изображения
VSPACE	10	Свободное пространство сверху и снизу от изображения.

Рисунки можно использовать в роли кнопок. Для этого тег изображения нужно поместить внутри контейнера – ссылки.

Пример:

< A HREF= photo.html>< IMG SRC= images/button.gif><A>

Рисунок может быть фоном веб – страницы для этого тегу <BODY> Добавляется атрибут BACKGROUND="images/fon.jpg."

Пример:

```
<BODY BACKGROUND="images/fon.jpg.">
```

8 ТАБЛИЦЫ

Таблица начинается с метки <TABLE> и заканчивается меткой </TABLE>. Метка <TABLE> может включать несколько атрибутов:

ALIGN

Устанавливает расположение таблицы по отношению к полям документа. Допустимые значения: ALIGN=LEFT (выравнивание влево), ALIGN=CENTER (выравнивание по центру), ALIGN=RIGHT (выравнивание вправо).

WIDTH

Ширина таблицы. Ее можно задать в пикселах (например, WIDTH=400) или в процентах от ширины страницы (например, WIDTH=80%).

BORDER

Устанавливает ширину внешней рамки таблицы и ячеек в пикселах (например, BORDER=4). Если атрибут не установлен, таблица показывается без рамки.

CELLSPACING

Устанавливает расстояние между рамками ячеек таблицы в пикселах (например, CELLSPACING=2).

CELLPADDING

Устанавливает расстояние между рамкой ячейки и текстом в пикселах (например, CELLPADDING=10).

ROWS

Устанавливает количество строк таблицы

COLS

Устанавливает количество столбцов таблицы

BACKGROUND

Устанавливает цвет фона таблицы (например, Red ,=ff0000)

BACKGROUND

Устанавливает фоновую картинку (например, Images/fon. Gif)

BORDERCOLOR

Устанавливает цвет рамки таблицы (например, Red ,=ff0000)

FRAME

Устанавливает рамки вокруг таблицы

RULES

Устанавливает линии в сетке между ячейками

GELLPADDING

Устанавливает расстояние между границей ячейки и данными в ней

CELLSPACING

Устанавливает свободное пространство внутри таблицы между ячейками

Таблица может иметь заголовок (<CAPTION> ... </CAPTION>), хотя заголовок не является обязательным. Метка <CAPTION> может включать атрибут ALIGN. Допустимые значения: <CAPTION ALIGN=TOP> (заголовок помещается над таблицей) и <CAPTION ALIGN=BOTTOM> (заголовок помещается под таблицей).

Каждая строка таблицы начинается с метки <TR> и заканчивается меткой </TR>. Метка <TR> может включать следующие атрибуты:

ALIGN

Устанавливает выравнивание текста в ячейках строки. Допустимые значения: ALIGN=LEFT (выравнивание влево), ALIGN=CENTER (выравнивание по центру), ALIGN=RIGHT (выравнивание вправо).

VALIGN

Устанавливает вертикальное выравнивание текста в ячейках строки. Допустимые значения: VALIGN=TOP (выравнивание по верхнему краю), VALIGN=MIDDLE (выравнивание по центру), VALIGN=BOTTOM (выравнивание по нижнему краю).

Каждая ячейка таблицы начинается с метки `<TD>` и заканчивается меткой

`</TD>`. Метка `<TD>` может включать следующие атрибуты:

NOWRAP

Присутствие этого атрибута означает, что содержимое ячейки должно быть показано в одну строку.

COLSPAN

Устанавливает "размах" ячейки по горизонтали. Например, `COLSPAN=3` означает, что ячейка простирается на три колонки.

ROWSPAN

Устанавливает "размах" ячейки по вертикали. Например, `ROWSPAN=2` означает, что ячейка занимает две строки.

ALIGN

Устанавливает выравнивание текста в ячейке. Допустимые значения: `ALIGN=LEFT` (выравнивание влево), `ALIGN=CENTER` (выравнивание по центру), `ALIGN=RIGHT` (выравнивание вправо).

VALIGN

Устанавливает вертикальное выравнивание текста в ячейке. Допустимые значения: `VALIGN=TOP` (выравнивание по верхнему краю), `VALIGN=MIDDLE` (выравнивание по центру), `VALIGN=BOTTOM` (выравнивание по нижнему краю).

WIDTH

Устанавливает ширину ячейки в пикселах (например, `WIDTH=200`).

HEIGHT

Устанавливает высоту ячейки в пикселах (например, `HEIGHT=40`).

Если ячейка таблицы пуста, вокруг нее не рисуется рамка. Если ячейка пуста, а рамка нужна, в ячейку можно ввести символьный объект ` ` (`non-breaking space` - неразрывающий пробел). Ячейка по-прежнему будет пустой, а рамка вокруг нее будет.

Пример HTML-документа, содержащего таблицу:

```

<HTML>
<HEAD>
<TITLE>Пример 10</TITLE>
</HEAD>
<H1>Простейшая таблица </H1>
<TABLE BORDER=1> <!--Это начало таблицы-->
<CAPTION> <!--Это заголовок таблицы-->
У таблицы может быть заголовок
</CAPTION>
<TR> <!--Это начало первой строки-->
<TD> <!--Это начало первой ячейки-->
Первая строка, первая колонка
</TD> <!--Это конец первой ячейки-->
<TD> <!--Это начало второй ячейки-->
Первая строка, вторая колонка
</TD> <!--Это конец второй ячейки-->
</TR> <!--Это конец первой строки-->
<TR> <!--Это начало второй строки-->
<TD> <!--Это начало первой ячейки-->
Вторая строка, первая колонка
</TD> <!--Это конец первой ячейки-->
<TD> <!--Это начало второй ячейки-->
Вторая строка, вторая колонка
</TD> <!--Это конец второй ячейки-->
</TR> <!--Это конец второй строки-->
</TABLE> <!--Это конец таблицы-->
</BODY>
</HTML>

```

9 СТИЛЬ В ДОКУМЕНТАХ HTML. ВНЕШНИЕ ТАБЛИЦЫ СТИЛЕЙ. КАСКАДЫ ТАБЛИЦ СТИЛЕЙ

Таблицы стилей упрощают определение интервалов между строками текста, отступов, цветов, используемых для текста и фона, размера и стиля шрифтов и другой информации. HTML 4.0 обеспечивает поддержку следующих функций таблиц стилей:

- ◆ *Гибкое размещение информации о стиле* Помещение таблиц стилей в отдельные файлы упрощает их повторное использование. Иногда полезно включать инструкции по представлению в документ, к которому

они применяются, в начало документа или в атрибуты элементов в теле документа.

- ◆ *Независимость от языков таблиц стилей.* Данная спецификация не привязывает HTML к конкретному языку таблиц стилей. Это позволяет использовать широкий диапазон таких языков, например, простые языки для большинства пользователей и более сложные для более специализированных случаев.

- ◆ *Каскады.* Эта возможность обеспечивается некоторыми языками таблиц стилей, такими как CSS, для объединения информации о стиле из нескольких источников. Это может быть, например, корпоративные положения о стиле, стили, общие для группы документов, а также стили, специфичные для одного документа. С использованием отдельного хранения эти таблицы стилей могут использоваться повторно, что упрощает работу авторов и повышает эффективность сетевого кэширования. Каскад определяет упорядоченную последовательность таблиц стилей, в которой правила более поздних таблиц имеют приоритет над более ранними. Не все языки таблиц стилей поддерживают каскады.

- ◆ *Зависимость от устройств.* Таблицы стилей, напротив, применяются к конкретным устройствам или группам устройств. Таблица стилей, предназначенная для экрана, может применяться при печати, но бесполезна для речевых браузеров. Данная спецификация позволяет Вам определить широкие категории устройств, к которым применяется конкретная таблица стилей. Это позволяет агентам пользователей избежать загрузки ненужных таблиц стилей. Языки таблиц стилей могут включать функции описания зависимости от устройств в одной таблице.

- ◆ *Альтернативные стили* Авторы могут предлагать читателям несколько способов просмотра документа. Например, таблица стилей для представления компактных документов с мелким шрифтом, или таблица, задающая крупные шрифты для удобства чтения. Данная спецификация позволяет авторам указать предпочитаемую таблицу стилей, а также

альтернативные таблицы для определенных пользователей или устройств. Агенты пользователей должны предоставлять пользователям возможность выбора одной из альтернативных таблиц или отключать все таблицы стилей.

HTML позволяет авторам связывать с документом любое число внешних таблиц стилей. Язык таблиц стилей определяет взаимодействие нескольких внешних таблиц стилей (например, правила «каскадов» CSS).

Авторы могут указать ряд взаимоисключающих таблиц стилей, называемых альтернативными. Пользователи могут выбирать таблицы, которые им больше нравятся. Например, автор может указать один стиль для небольших экранов, другой - для слабовидящих пользователей (например, с использованием крупного шрифта). Агенты пользователей должны предоставлять пользователям возможности выбора одной из альтернативных таблиц.

Авторы также могут указать постоянные таблицы стилей, которые агенты пользователей должны применять в дополнение к альтернативным таблицам стилей.

Авторы указывают внешние таблицы стилей с помощью атрибутов элемента **LINK**.

1. Устанавливается в атрибуте href местоположение файла таблицы стилей. Значением атрибута href должен быть URI.

2. Устанавливается для атрибута type значение, указывающее язык связанного ресурса (таблицы стилей). Это позволяет агентам пользователей не загружать таблицы стилей, использующие неподдерживаемые языки.

3. Указывается, является ли таблицы стилей постоянной, предпочитаемой или альтернативной:

- Чтобы таблицы была постоянной, устанавливается для атрибута rel значение «stylesheet», и не устанавливайте атрибут title.

- Чтобы таблица была предпочитаемой, устанавливается для атрибута rel значение «stylesheet», и таблице дается имя с помощью атрибута title.

- Чтобы указать альтернативную таблицу, устанавливается для атрибута rel значение «alternate stylesheet», и таблице дается имя с помощью атрибута title.

Каскадные языки таблиц стилей, такие как CSS, позволяют использовать информацию о стиле из нескольких источников. Однако не все языки таблиц стилей поддерживают каскады. Чтобы определить каскад, авторы указывают последовательность элементов LINK и/или STYLE. Каскад информации таблиц стилей производится в порядке указания элементов в разделе HEAD.

10 ФРЕЙМЫ

Фреймы в HTML позволяют авторам представлять документы в нескольких разделах, которые могут быть независимыми или вложенными окнами. Это обеспечивает дизайнерам способ оставлять некоторую информацию видимой, в то время как другая информация прокручивается или заменяется. Например, в одном окне 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.

Документ HTML, в котором описывается компоновка фреймов (называемый документом с фреймами), выглядит не так, как документ HTML без фреймов. Стандартный документ имеет один раздел HEAD и один раздел BODY. Документ с фреймами имеет раздел HEAD и раздел FRAMESET, который заменяет раздел BODY.

В разделе FRAMESET задается расположение фреймов в основном окне агента пользователя. Кроме того, в разделе FRAMESET может присутствовать элемент NOFRAME с альтернативным содержанием для агентов пользователей, не поддерживающих фреймы или сконфигурированных так, чтобы их не

показывать. Элементы, обычно помещаемые в раздел BODY, не должны присутствовать до первого элемента FRAMESET, иначе элемент FRAMESET будет игнорироваться.

Таблица 7 Атрибуты <FRAMESET>:

атрибуты	значение	описание	
COLS	100*	Количество и размеры колонок	В одном теге <FRAMESET> можно использовать только один из этих атрибутов
ROWS	50.*	Количество и размеры строк	
BORDER	5	Ширина рамки	
BORDERCOLOR	Red,=ff0000	Цвет рамки таблицы	
FRAMEBORDER	No, yes	Определяет, будет ли выводиться граница фреймов	
FRAMESPACING	0	Расстояние между фреймами	

Авторы могут разделять данные между несколькими фреймами, включая эти данные с помощью элемента OBJECT. Элемент OBJECT следует включать в элемент HEAD документа с фреймами и давать ему имя с помощью атрибута id. Любой документ, являющийся содержимым фрейма, может ссылаться на этот идентификатор.

Элемент FRAME описывает свойства отдельного фрейма.

Таблица 8 Атрибуты <FRAME>:

Атрибуты	Значение	Описание
SRC	One.html	Адрес загружаемого во фрейм файла
NAME	first	Имя фрейма
NORESIZE	-	Обозначает, что границы фрейма нельзя переместить
SRCOLLING	Yes, no, auto	Определяет, отображать ли во фрейме полосы прокрутки
FRAMEBORDER	Yes, no	Определяет, будет ли выводить границу фрейма
MARGINHEIGHT	0	Отступ от правой и левой границ фрейма до его содержимого
MARGINWIDTH	0	Отступ от верхней и нижней границ фрейма до его содержимого

Элемент NOFRAME задает содержимое, которое должно отображаться, только если не отображаются фреймы. Агенты пользователей,

поддерживающие фреймы, должны отображать содержимое объявления NOFRAME, только если они сконфигурированы так, чтобы не отображать фреймы. Агенты пользователей, не поддерживающие фреймы, должны отображать содержимое элемента NOFRAME в любом случае.

11 ФОРМЫ

Форма - это инструмент, с помощью которого HTML-документ может послать некоторую информацию в некоторую заранее определенную точку внешнего мира, где информация будет некоторым образом обработана.

Форма HTML - это раздел документа, в котором содержатся обычная информация, разметка и специальные элементы, называемые управляющие элементы (флажки, кнопки с зависимой фиксацией, меню и т.д.), а также метки этих управляющих элементов. Обычно пользователи "заполняют" форму, модифицируя управляющие элементы (вводя текст, выбирая пункты меню и т.д.) перед тем, как предоставить форму агенту пользователя для обработки (например, на Web-сервер, на почтовый сервер и т.д.) В HTML определены следующие типы управляющих элементов:

- ◆ *кнопки*

Авторы могут создавать три типа кнопок: кнопки отправки, кнопки сброса, прочие кнопки. Авторы создают кнопки с помощью элемента BUTTON или INPUT.

- ◆ *флажки*

Флажки (и кнопки с зависимой фиксацией) - это переключатели вкл./выкл., которые могут переключаться пользователем. Переключатель «включен», если для управляющего элемента установлен атрибут selected.

- ◆ *кнопки с зависимой фиксацией*

Кнопки с зависимой фиксацией похожи на флажки за исключением того, что, если несколько кнопок используют одно и то же имя управляющего элемента, они являются взаимоисключающими: если одна кнопка

включена, другие обязательно выключены. Для создания кнопок с зависимой фиксацией используется элемент INPUT.

- ◆ *меню*

Предоставляют пользователям варианты на выбор. Меню создается с помощью элемента SELECT, а также элементов OPTGROUP и OPTION.

- ◆ *текстовый ввод*

Для ввода текста пользователем авторы могут создавать управляющие элементы двух типов. Элемент INPUT создает управляющий элемент для ввода текста из одной строки, а элемент TEXTAREA - элемент для ввода текста из нескольких строк.

- ◆ *выбор файлов*

Управляющие элементы этого типа позволяют пользователям выбирать файлы, содержимое которых может передаваться вместе с формой. Для создания этого управляющего элемента используется элемент INPUT.

- ◆ *скрытые управляющие элементы*

Авторы могут создавать управляющие элементы, не представляемые пользователям, но имеющие значения, которые передаются с формой. Обычно они используются для хранения информации между обменом клиент/сервер, которая в противном случае могла бы пропасть вследствие stateless природы протокола HTTP. Для создания скрытого управляющего элемента используется элемент INPUT.

- ◆ *объекты*

Авторы могут помещать в формы общие объекты, так что связанные с ними значения будут передаваться с другими управляющими элементами. Для создания таких управляющих элементов используется элемент OBJECT.

Элемент FORM используется для создания самой формы. Элемент SELECT создает меню.

Элементы форм

TEXTAREA

<TEXTAREA> - текстовое поле любого размера. По умолчанию формируется чистое поле, состоявшее из 4-х строк по 40 символов в каждой. Можно указать некоторый текст по умолчанию, поместив его внутрь контейнера <TEXTAREA>... </TEXTAREA>. Атрибуты <TEXTAREA>:

Таблица 8 Элементы форм:

атрибут	значение	описание
ROWS	5	Число строк в поле.
COLS	50	Кол-во символов в строке.
NAME	Коммент	Имя поля. Обязательный атрибут.

Пример.

Вид в окне браузера:

Укажите доп. сведения о себе:



html-код:

укажите доп. сведения о себе:

<TEXTAREA ROWS="5" cols="30"> я хороший! </TEXTAREA>

SELECT

<SELECT> - список значений выпадающего меню или списка. Эти значения перечислены внутри контейнера <SELECT>...</SELECT> с помощью тегов <OPTION>. Можно указать некоторый текст по умолчанию, поместив его внутрь контейнера <TEXTAREA>... </TEXTAREA>.

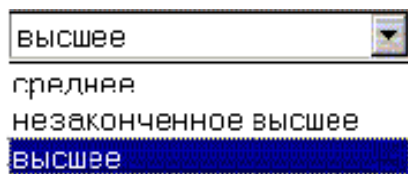
Таблица 9 Атрибуты <SELECT>:

атрибут	значение	описание
SIZE	5	Число выводимых на экран пунктов.
MULTIPLE	-	Режим выбора нескольких значений.
NAME	Коммент	Имя поля. Обязательный атрибут.

Пример. Раскрывающийся список.

Вид в окне браузера:

Образование:



Образование: BR

SELECT NAME = Menu

OPTION VALUE = «1» среднее

OPTION VALUE = «2» незаконченное высшее

OPTION VALUE = «3» SELEKTED высшее

SELECT

Таблица 10 Атрибуты OPTION:

атрибут	значение	описание
VALUE	любое	Значение, присваиваемое выбранному элементу списка и отсылаемое серверу.
SELEKTED	-	Делает данный элемент списка выбранным.

ТЕХТ

INPUT TYPE = "text" NAME = "Klient" - одностороннее текстовое поле.

Таблица 11 Атрибуты ТЕХТ:

Атрибуты	Значение	Описание
SIZE	40	Длина поля в символах.
MAXLENGTH	70	Максимальное количество символов, которое можно записать в поле.
VALUE	Введите ваше имя	
TYPE	Text password checkbox radio	Текстовое поле Поле для ввода пароля Флажок Переключатель
NAME	Klient	Имя элемента формы

Пример:

Вид в окне браузера:

Введите ваше имя

HTML – код

Введите ваше имя: BR

```
<INPUT TUPE="text" NAME="Klient" SIZE="30">
```

PASSWORD

<INPUT TYPE="password" NAME="Klient_ID"> - поле для ввода пароля. Атрибуты те же, что и у простого текстового поля. Отсутствует лишь VALUE.

Пример.

Вид в окне браузера:

Введите пароль.



Html-код

Введите пароль:


```
<INPUT TYPE="password" NAME="Klient_ID" SIZE="30">
```

CHECKBOX

<INPUT TYPE="checkbox" NAME="flag1"> - флажок, который можно установить или оставить пустым.

Таблица 12 Атрибуты CHECKBOX:

атрибуты	значение	описание
CHECKED	-	Установка флажка по умолчанию
NAME	Flag1	Имя элемента формы
VALUE	любое	Отсылается серверу

Пример:

Вид в окне браузера:

Какими иностранными языками владеете?

- Английский
- Немецкий
- Французский
- Итальянский
- Испанский

html-код:

Какими иностранными языками владеете?


```
<INPUT TYPE="checkbox" NAME="flag1"> английский <BR>
```

```
<INPUT TYPE="checkbox" NAME="flag2"> немецкий <BR>
```

```
<INPUT TYPE="checkbox" NAME="flag3"> французский <BR>
```

```
<INPUT TYPE="checkbox" NAME="flag4"> итальянский <BR>
```

```
<INPUT TYPE="checkbox" NAME="flag5"> испанский
```

RADIO

```
<INPUT TYPE="radio" NAME="color" VALUE="green">зеленый
```

<INPUT TYPE="radio" NAME="color" VALUE="yellow">желтый
 <INPUT TYPE="radio" NAME="color" VALUE="red">красный
 - группа переключателей. Предлагается выбор только одного из
 родственных элементов. Объединяются в одну группу с помощью одинакового
 значения атрибута NAME.

Таблица 13 Атрибуты RADIO:

атрибуты	значение	описание
CHECKED		Установка флажка по умолчанию
NAME	Flag1	Имя элемента формы
VALUE	любое	Отсылается серверу

Пример:

Вид в окне браузера:

Уровень знаний ПК:

- нет
- пользователь
- продвинутый пользователь
- программист

html-код:

Уровень знаний ПК:

<INPUT TYPE="radio" NAME="PC" VALUE="no">нет

<INPUT TYPE="radio" NAME="PC" VALUE="user">пользователь

<INPUT TYPE="radio" NAME="PC" VALUE="superuser" продвинутый
 пользователь

<INPUT TYPE="radio" NAME="PC" VALUE="programmer">

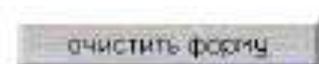
программист

RESET

<INPUT TYPE="reset" VALUE="очистить форму"> -кнопка,
 с помощью которой можно отметить всю введенную в форму информацию,
 восстановив первоначальные значения. По умолчанию на кнопке
 устанавливается надпись "reset" (в русских версиях браузеров –« сброс»), её
 можно изменить с помощью атрибута VALUE.

Пример.

Вид в окне браузера



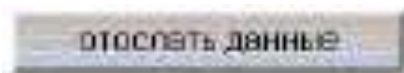
Html-код:

<INPUT TYPE="reset" VALUE="ОЧИСТИТЬ ФОРМУ">

SUBMIT

`<INPUT TYPE='submit' VALUE='отослать данные'>` - кнопка, с помощью которой можно отправить на сервер данные, введенные в форму. По умолчанию на кнопке устанавливается надпись "submit" (в русских версиях браузеров – "подача запроса"), её можно изменять с помощью атрибута VALUE.

Пример



Html -код:

```
< INPUT TYPE='reset' VALUE='отослать данные' >
```

Пример использования форм:

```
<HTML>
<HEAD>
<TITLE>Пример 12</TITLE>
</HEAD>
<H1>Несколько более сложная форма </H1>
<FORM ACTION="http://206.31.82.215/hp/nc/fd-win.pht" METHOD=post>
<H2>Расскажите немного о себе...</H2>
<P>Указывать подлинные данные совсем не обязательно.
Для целей демонстрации вполне подойдут и вымышленные. </P>
<P>Имя: <INPUT TYPE=text SIZE=40 NAME=fn><BR>
Фамилия: <INPUT TYPE=text SIZE=40 NAME=ln><BR>
Пол: <INPUT TYPE=radio NAME=gender VALUE="male" checked>мужской
<INPUT TYPE=radio NAME=gender VALUE="female">женский<BR>
Возраст: <INPUT TYPE=text SIZE=5 NAME=age> лет<BR>
<INPUT TYPE=submit VALUE="Запустить обработчик"></P>
</FORM>
</BODY>
</HTML>
```

СПИСОК ЛИТЕРАТУРЫ

1. Фуфаев Д. Э., Фуфаев Э. В. Разработка и эксплуатация автоматизированных информационных систем: учебник для студентов среднего профессионального образования – М.: Издательский центр «Академия», 2019.
2. Мезенцев К. Н. Автоматизированные информационные системы: учебник для студентов среднего профессионального образования – М.: Издательский центр «Академия», 2022.

Интернет-ресурсы:

1. <http://www.isuct.ru>
2. <http://www.lcard.ru/>
3. <http://www.interface.ru>
4. <http://www.caseclub.ru>
5. <http://www.citcite.ru>
6. <http://citforum.ru>