

HTML

План

1. **Общая структура документа**
2. **Теги для работы со строками**
3. **Форматирование символов**
4. **Списки**
5. **Таблицы**

1. Общая структура документа

- HTML (HyperText Markup Language – "язык разметки гипертекста") – стандартный язык разметки документов во Всемирной паутине. Большинство веб-страниц создаются при помощи языка HTML. Язык HTML интерпретируется браузером и отображается в виде документа, в удобной для человека форме. HTML является приложением SGML (стандартного обобщенного языка разметки) и соответствует международному стандарту ISO 8879.
- Язык HTML был разработан британским ученым Тимом Бернерсом-Ли приблизительно в 1991-1992 годах в стенах Европейского совета по ядерным исследованиям в Женеве (Швейцария).

Популярные браузеры

1. Internet Explorer
2. Mozilla Firefox
3. Google Chrome
4. Safari
5. Opera

Спецификация HTML имеет несколько версий

- RFC 1866 – HTML 2.0, одобренный как стандарт 22 сентября 1995 года;
- HTML 3.2 – 14 января 1997 года;
- HTML 4.0 – 18 декабря 1997 года;
- HTML 4.01 – 24 декабря 1999 года;
- ISO/IEC 15445:2000 (так называемый ISO HTML, основан на HTML 4.01 Strict) – 15 мая 2000 года;
- HTML 5 – в разработке.

Структура HTML документа

- HTML– это теговый язык разметки документов. Чтобы браузер понимал, что имеет дело не с простым текстом, а с особым элементом, который задает его форматирование, применяются теги. Теги являются основой HTML и берутся в угловые скобки. Общий синтаксис написания тегов следующий.

```
<тег параметр1="значение" параметр2="значение">  
<тег параметр1="значение" параметр2="значение">  
  . . .  
</тег>
```

- Теги бывают одиночными и парными (контейнеры). Одиночный тег используется самостоятельно, а парный может включать внутри себя другие теги или текст. У тегов допустимы различные параметры, которые разделяются между собой пробелом. Впрочем, есть теги, без всяких дополнительных атрибутов. Параметры условно можно подразделить на обязательные, они непременно должны присутствовать, и необязательные, их добавление зависит от цели применения.

Рекомендации и свойства написания тегов

- теги не чувствительны к регистру;
- если для тега не добавлен какой-либо допустимый параметр, браузер будет подставлять значение, заданное по умолчанию;
- внутри тега между его параметрами допустимо ставить перенос строк;
- все параметры тегов рекомендуется брать в двойные или одинарные кавычки (хотя в современном браузере отсутствие кавычек не приведет к ошибкам);
- если какой-либо тег или его параметр был написан неверно, то браузер проигнорирует подобный тег ;
- существует определенная иерархия вложенности тегов(например, метатеги должны находиться внутри контейнера <HEAD>), причем если теги между собой равноценны в иерархии, то их последовательность не имеет значения;
- существует три состояния закрывающего тега:
- обязателен – закрывающий тег должен присутствовать всегда, иначе это приведет к ошибке при отображении документа;
- не требуется – для некоторых тегов вроде
закрывающего тега нет в принципе;
- не обязателен – говорит о том, что разработчик может его, как добавить, так и опустить, к ошибке это не приведет;
- порядок параметров в любом теге не имеет значения и на результат отображения элемента не влияет.

Общая структура документа

```
<html>
  <head>
    <title>Это заголовок
страницы</title>
  </head>
  <body>
    <h1>Здравствуйте!</h1>
    <p>Это моя первая страница HTML.
<b>Этот текст выводится жирным
шрифтом.</b>
    </p>
  </body>
</html>
```

2. Теги для работы со строками

Параграфы

```
<html>
```

```
  <body>
```

```
    <p>Это параграф 1.</p>
```

```
    <p>Это параграф 2.</p>
```

```
    <p>Это параграф 3.</p>
```

```
  </body>
```

```
</html>
```

Заголовки

Заголовки определяются с помощью тегов от <h1>до <h6>. <h1>определяет заголовок самого большого размера, а <h6>определяет заголовок самого маленького размера.

<h1>Это заголовок первого уровня</h1>

<h2>Это заголовок второго уровня</h2>

<h3>Это заголовок третьего уровня</h3>

<h4>Это заголовок четвертого уровня</h4>

<h5>Это заголовок пятого уровня</h5>

<h6>Это заголовок шестого уровня</h6>

Переносы строк

Для переноса внутри параграфа используется тег
, который выполняет принудительный перенос строки.

```
<html>
  <body>
    <p>Это <br>пара<br>граф с
переносами строк</p>
  </body>
</html>
```

Горизонтальная линейка <hr>

Разделять различные элементы можно при помощи горизонтальной линейки, для этого используйте тег <hr>:

```
<html>
  <body>
    <p>Этот параграф отобразится
сверху горизонтальной полосы.</p>
    <hr>
    <p>Этот параграф отобразится снизу
горизонтальной полосы.</p> </body>
</html>
```

Комментарии в HTML

Тег комментария используется для вставки комментариев в исходный код HTML. Комментарии будут проигнорированы браузером.

Комментарии можно использовать для пояснения кода, что может помочь при редактировании исходного кода в будущем.

```
<!-- Это комментарий -->
```

Этот пример демонстрирует некоторые особенности поведения по умолчанию элементов параграфа.

```
<html>
  <body>
    <p> Этот параграф
      содержит много строк
      в исходном коде,
      но браузер это
      игнорирует. </p>
    <p> Этот параграф содержит
      много пробелов в исходном
      коде, но
      браузер это игнорирует. </p>
  </body>
</html>
```


3. Форматирование СИМВОЛОВ

- `<i> ... </i>` или ` ... `- вывод курсивом, `<i>` - структурный элемент
- ` ... ` или ` ... ` - вывод жирным шрифтом, `` - структурный элемент
- `^{...}`- верхний индекс и `_{...}`- нижний индекс

- Тег `<pre>` сообщает браузеру, что находящийся внутри текст должен быть выведен как есть.

```
<html>
```

```
  <body>
```

```
    <pre> Это предварительно  
форматированный текст. Он  
сохраняет как пробелы, так и  
переносы строк.
```

```
    </pre>
```

```
  </body>
```

```
</html>
```

- Тег `<code>` маркирует код компьютерной программы. Браузеры обычно отображают его моноширинным шрифтом. Этот элемент является структурным, поэтому его использование предпочтительней использования тега `<tt>`, который маркирует моноширинный текст.

Для маркировки переменных используется тег `<var>`, который обычно отображается курсивом.

```
<html>
```

```
  <body>
```

```
    <p>Версии стандарта HTML  
обычно маркируются следующим  
образом
```

```
    <var>x</var>.<var>y</var>.</p>
```

```
  </body>
```

```
</html>
```

Этот пример показывает, как изменять направление вывода текста.

```
<html>
  <body>
    <p> Если используемый браузер
поддерживает двунаправленное
представление (bdo), то следующая
строка будет записана справа
налево (rtl): </p>
    <bdo dir="rtl"> Здесь какой-то
арабский текст </bdo>
  </body>
</html>
```

Этот пример показывает, как использовать длинные и короткие цитаты.

```
<html>
```

```
  <body>
```

Здесь представлена длинная цитата:

```
<blockquote> Это длинная цитата. Это  
длинная цитата. Это длинная цитата. Это  
длинная цитата. Это длинная цитата.  
</blockquote>
```

Здесь представлена короткая цитата:

```
<q> Это короткая цитата </q>
```

```
<p> Для элемента blockquote браузер  
вставляет дополнительные переносы строки,  
пустые строки и поля, но элемент q не  
изображается каким-то специальным образом.  
</p>
```

```
  </body>
```

```
</html>
```

Символьные элементы

- Некоторые символы, такие как символ <, имеют в HTML специальное значение. Поэтому их нельзя использовать в тексте в явном виде. Для их отображения используются символьные элементы CER (Character Entity Reference).

Таблица наиболее часто используемых символьных объектов

Результат	Описание	Имя объекта	Номер объекта
	неразрывный пробел	<code>&nbsp;</code>	<code>&#160;</code>
<	меньше	<code>&lt;</code>	<code>&#60;</code>
>	больше	<code>&gt;</code>	<code>&#62;</code>
&	амперсанд	<code>&amp;</code>	<code>&#38;</code>
"	двойная кавычка	<code>&quot;</code>	<code>&#34;</code>
'	апостроф	<code>&apos;</code>	<code>&#39;</code>
¢	цент	<code>&cent;</code>	<code>&#162;</code>
£	фунт стерлингов	<code>&pound;</code>	<code>&#163;</code>
¥	йена	<code>&yen;</code>	<code>&#165;</code>
§	параграф	<code>&sect;</code>	<code>&#167;</code>
©	авторское право	<code>&copy;</code>	<code>&#169;</code>
®	зарегистрированная торговая марка	<code>&reg;</code>	<code>&#174;</code>
x	умножение	<code>&times;</code>	<code>&#215;</code>
÷	деление	<code>&divide;</code>	<code>&#247;</code>

4. Создание гиперссылок

Этот пример показывает, как создавать ссылки в документе HTML.

```
<html>
  <body>
    <p> <a href="page1.htm"> Этот
текст</a> является ссылкой на
страницу на этом Web-сайте. </p>
    <p><a href="http://www.csru.ru/">
Этот текст </a> является ссылкой на
страницу во Всемирной Паутине. </p>
  </body>
</html>
```

Изображение в виде ссылки

```
<html>
  <body>
    <p> Можно также использовать в
    качестве ссылки изображение: <a
    href="http://www.cspu.ru/">  </a> </p>
  </body>
</html>
```

Синтаксис гипертекстовой ссылки

- `текст ссылки`

С помощью атрибута href можно задавать ссылки не только по протоколу HTTP, но и по другим:

- **http://...** - создает ссылку на www-документ;
- **ftp://...** - создает ссылку на ftp-сайт или расположенный на нем файл;
- **mailto:...** - запускает почтовую программу-клиент с заполненным полем имени получателя. Если после адреса поставить знак вопроса, то можно указать дополнительные атрибуты, разделенные знаком "&";
- **news:..** - создает ссылку на конференцию сервера новостей;
- **telnet://...** - создает ссылку на telnet-сессию с удаленной машиной;
- **wais://...** - создает ссылку на WAIS - сервер;
- **gopher://...** - создает ссылку на Gopher - сервер;

С помощью атрибута **target** можно определить, где будет открыт документ, на который указывает ссылка. В качестве значения необходимо задать либо имя одного из существующих фреймов, либо одно из следующих зарезервированных имен:

- **_self** - указывает, что определенный в параметре href документ должен отображаться в текущем фрейме;
- **_parent** - указывает, что документ должен отображаться во фрейме-родителе текущего фрейма. Иначе говоря, **_parent** ссылается на окно, содержащее frameset, включающий текущий фрейм;
- **_top**- указывает, что документ должен отображаться в окне-родителе всей текущей фреймовой структуры;
- **_blank** - указывает, что документ должен отображаться в новом окне.

```
<html>
  <body>
    <a href="http://www.cspu.ru/" target="_blank">Добро
пожаловать!</a>
    <a href="http://www.cspu.ru/" target="_parent">Добро
пожаловать!</a>
    <a href="http://www.cspu.ru/" target="_top">Добро
пожаловать!</a>
    <a href="http://www.cspu.ru/" target="_self">Добро
пожаловать!</a>
  </body>
</html>
```

Для перехода внутри страницы к конкретному разделу используют атрибут `name` гипертекстовой ссылки `a`, при помощи которого создается именованный якорь.

```
<a name="label">ссылка на  
именованный якорь</a>
```

В данном примере внутри документа создается своеобразная метка, при этом именованный якорь никак не отображается в окне браузера.

Для перехода на место, которое отмечено именованным якорем используется знак # и имя якоря в конце URL следующим образом:

```
<a href="page1.html#label">переход на  
именованный якорь </a>
```

Переход внутри файла "page1.html" на именованный якорь выглядит следующим образом:

```
<a href="#label">переход на именованный  
якорь </a>
```

Если браузер не сможет найти указанный именованный якорь на вызываемой странице, то переход произойдет на начало документа, при этом сообщения об ошибке доступа не возникает.

5. Списки

- Упорядоченные
- Неупорядоченные

Неупорядоченный список начинается с тега ``. Каждый элемент списка начинается с тега ``.

```
<html>
  <body>
    <h4>Неупорядоченный
список:</h4>
    <ul>
      <li>элемент 1</li>
      <li>элемент 2</li>
      <li>элемент 3</li>
    </ul>
  </body>
</html>
```

Упорядоченные списки

Упорядоченный список начинается с тега ``.

Каждый элемент списка начинается с тега ``. У тега `` может быть два атрибута: `start` (определяет первое число, с которого начинается нумерация пунктов) и `type` (определяет стиль нумерации пунктов). Может иметь значения:

- "A" - заглавные буквы A, B, C ...
- "a" - строчные буквы a, b, c ...
- "I" - большие римские числа I, II, III ...
- "i" - маленькие римские числа i, ii, iii ...
- "1" - арабские числа 1, 2, 3 ...

```
<html>
  <body>
    <h4>Упорядоченный список:</h4>
    <ol>
      <li>элемент 1</li>
      <li>элемент 2</li>
      <li>элемент 3</li>
    </ol>
  </body>
</html>
```

Вложенные списки

```
<html>
  <body>
    <h4>Вложенный список:</h4>
    <ul>
      <li>элемент 1</li>
      <li>элемент 2</li>
        <ul>
          <li>элемент 2.1</li>
          <li>элемент 2.2</li>
        </ul>
      <li>элемент 3</li>
    </ul>
  </body>
</html>
```

Списки определений

- Список определений не является списком элементов. Это список терминов и определений терминов.
- Список определений начинается с тега `<dl>`. Каждый термин списка определений начинается с тега `<dt>`. Каждое определение списка начинается с тега `<dd>`.

```
<html>
  <body>
    <dl>
      <dt>элемент 1</dt>
      <dd>описание элемента 1</dd>
      <dt>элемент 2</dt>
      <dd>описание элемента 2</dd>
    </dl>
  </body>
</html>
```

5. Таблицы

- В web-дизайне таблицы являются одним из основных методов структурирования информации. Они представляют данные в виде удобных для восприятия колонок и строк, что значительно упрощает анализ информации. С их помощью можно легко отделить одну часть страницы от другой.
- Основным тегом для обозначения таблицы является `<table>`. Элемент TABLE представляет собой тег-контейнер, в котором размещается содержимое таблицы. Построение таблицы осуществляется по строкам, для обозначения которых применяется контейнер TR. Внутри контейнера строк помещаются контейнеры для обозначения ячеек. Стандарт HTML определяет два типа контейнеров для обозначения ячеек `<th>` и `<td>`. Первый предназначен для обозначения заголовков, а второй для данных в ячейках. Отличие этих двух элементов заключается лишь в том как их содержимое отображается браузером, заголовки большинство браузеров выделяют полужирным шрифтом и центрируют в своих ячейках. Ячейка данных может содержать текст, изображения,

Пример таблицы: одна ячейка

```
<table border="1">  
  <tr>  
    <td>Одна строка и одна  
    ячейка</td>  
  </tr>  
</table>
```

Пример таблицы: одна строка и три столбца

```
<table border="1">  
  <tr>  
    <td> столбец 1</td>  
    <td> столбец 2</td>  
    <td> столбец 3</td>  
  </tr>  
</table>
```

Пример таблицы: две строки и три столбца

```
<table border="1">  
  <tr>  
    <td>1.1</td>  
    <td>1.2</td>  
    <td>1.3</td>  
  </tr>  
  <tr>  
    <td>2.1</td>  
    <td>2.2</td>  
    <td>2.3</td>  
  </tr>  
</table>
```

Для тега `table` определены следующие атрибуты.

- **align**- определяет способ горизонтального выравнивания таблицы на странице. Возможные значения: `left`, `center`, `right`. Значение по умолчанию - `left`.
- **valign**- должен определять способ вертикального выравнивания для содержимого таблицы. Возможные значения: `top`, `bottom`, `middle`.
- **border**- определяет ширину внешней рамки таблицы (в пикселах). При `BORDER="0"` или при отсутствии этого параметра рамка отображаться не будет.
- **cellpadding**- определяет расстояние (в пикселах) между рамкой каждой ячейки таблицы и содержащимся в ней материалом.
- **cellspacing**- определяет расстояние (в пикселах) между границами соседних ячеек.
- **width**- определяет ширину таблицы. Ширина задается либо в пикселах, либо в процентном отношении к ширине окна браузера. По умолчанию этот параметр определяется автоматически в зависимости от объема содержащегося в таблице материала.
- **height**- определяет высоту таблицы. Высота задается либо в пикселах, либо в процентном отношении к высоте окна браузера. По умолчанию этот параметр определяется автоматически в зависимости от объема содержащегося в таблице материала.
- **bgcolor**- определяет цвет фона ячеек таблицы. Задается либо RGB-значением в шестнадцатиричной системе, либо одним из 16 базовых цветов.
- **background**- позволяет заполнить фон таблицы рисунком. В качестве значения необходимо указать URL рисунка.

- Ячейки таблицы без содержимого в большинстве браузеров выводятся не очень хорошо. Если необходимо сделать ячейку таблицы пустой, то разместите в ней неразрывный пробел ** **;
- Такие элементы как **<thead>**, **<tbody>** и **<tfoot>** используются редко в связи с тем, что не все браузеры их поддерживают.

- Для тегов `<td>` и `<th>` очень полезными являются атрибуты `colspan` и `rowspan`. Первый показывает сколько ячеек надо объединить по горизонтали, а второй по вертикали. Следует заметить, что если мы применяем объединение ячеек, то общее число ячеек с учетом объединенных должно быть равным.

```

<html> <body>
  <h4>Правильное применение
  colspan:</h4>
  <table border="1">
    <tr>
      <td colspan="2">100</td>
      <td>300</td>
    </tr>
    <tr>
      <td>400</td>
      <td>500</td>
      <td>600</td>
    </tr>
  </table>
  <h4>Неправильное применение
  colspan:</h4>
  <table border="1">
    <tr>
      <td colspan="2">100</td>
      <td>200</td>
      <td>300</td>
    </tr>
    <tr>
      <td>400</td>
      <td>500</td>
      <td>600</td>
    </tr>
  </table>

```

```

<h4>Правильное применение
rowspan:</h4>
<table border="1">
  <tr>
    <td rowspan="2">100</td>
    <td>200</td>
    <td>300</td>
  </tr>
  <tr>
    <td>500</td>
    <td>600</td>
  </tr>
</table>
<h4>Неправильное применение
rowspan:</h4>
<table border="1">
  <tr>
    <td rowspan="2">100</td>
    <td>200</td>
    <td>300</td>
  </tr>
  <tr>
    <td>400</td>
    <td>500</td>
    <td>600</td>
  </tr>
</table>
</body>
</html>

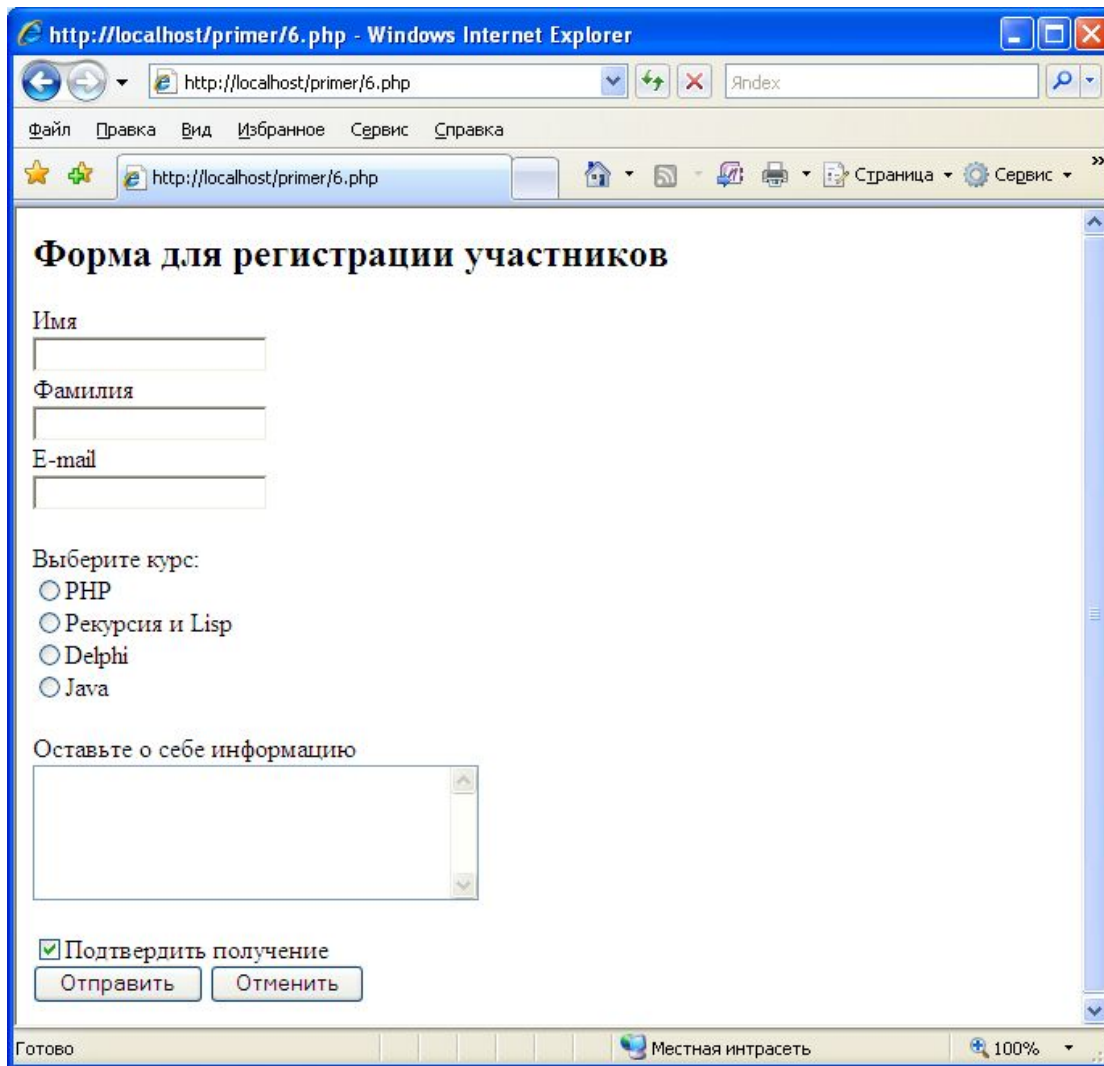
```

Блок

- Элемент `<div>` является блочным элементом и предназначен для выделения фрагмента документа с целью изменения вида содержимого. Как правило, вид блока управляется с помощью стилей. Чтобы не описывать каждый раз стиль внутри тега, можно выделить стиль во внешнюю таблицу стилей, а для тега добавить атрибут `class` или `id` с именем селектора **(более подробно при изучении CSS)**

Формы HTML

- Формы HTML предназначены для организации взаимодействия с пользователем. Они позволяют вводить текст, осуществлять выбор из предложенных значений при помощи списков или кнопок. С помощью форм можно организовать интерактивный обмен информацией между Web-страницей и сервером. Можно определить формы как электронные бланки для заполнения различных данных таких как, например, имя, возраст, выбор страны проживания и других. Как правило, форма работает совместно с установленным на сервере сценарным приложением, обрабатывающим введенную информацию.



Структура формы

```
<form  
  name="имя формы"  
  action="путь к программе-обработчику формы"  
  method="метод передачи данных">
```

Элементы формы или тело формы

```
</form>
```

Упрощенная запись

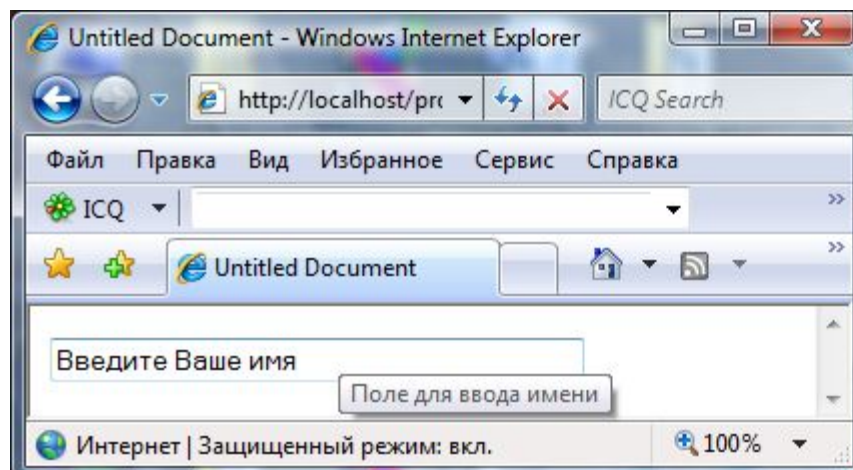
```
<form>
```

Элементы формы или тело формы

```
</form>
```

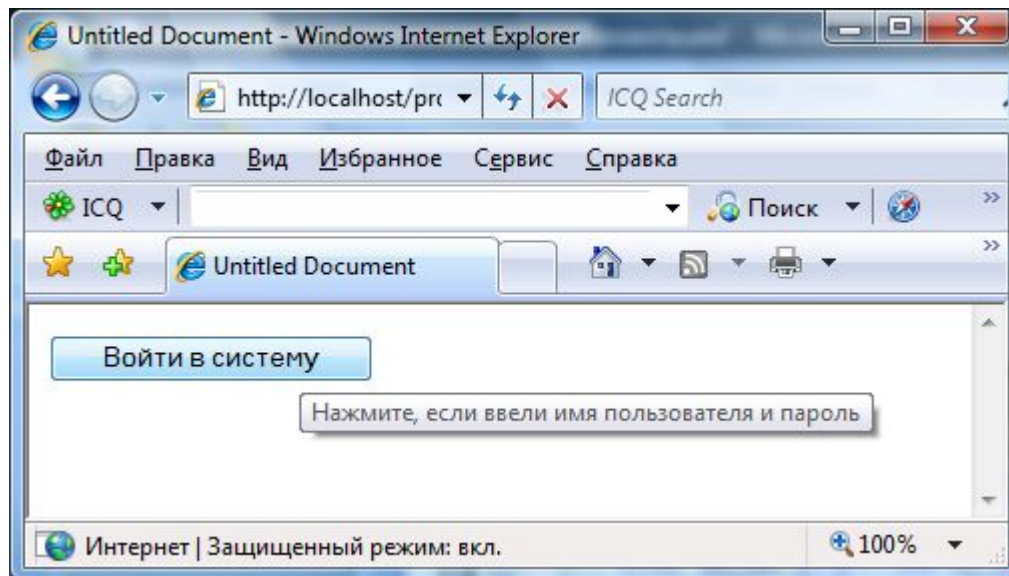
Поле ввода

```
<input  
  type=text  
  name="имя элемента"  
  value="значение элемента"  
  title="всплывающая подсказка"  
  size=длина  
  maxlength=максимальное количество элементов>
```



Кнопка

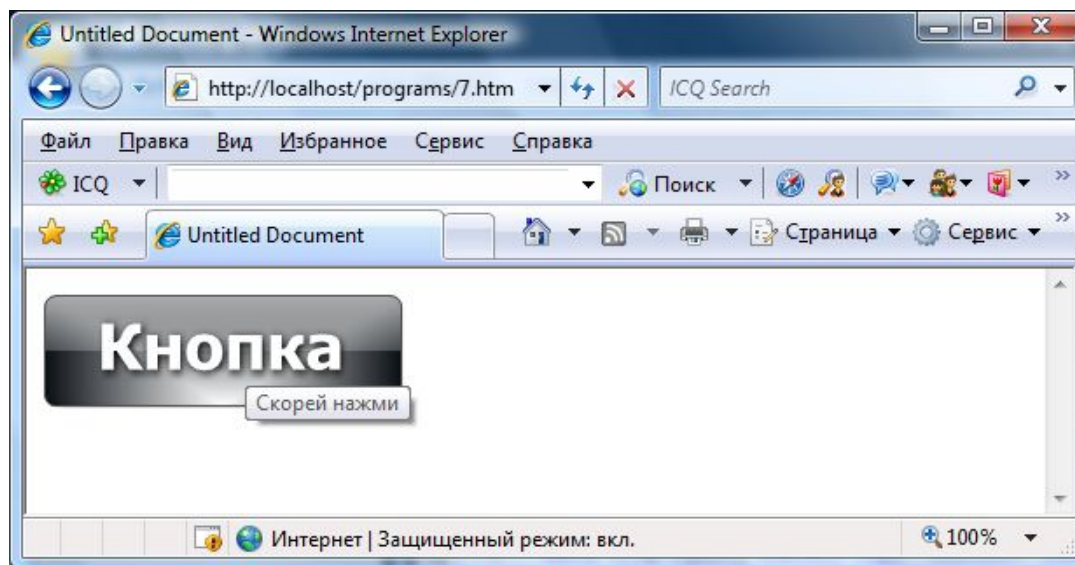
```
<input  
  type=submit  
  name="имя элемента"  
  value="текст кнопки"  
  title="всплывающая подсказка"  
>
```



Кнопка с рисунком

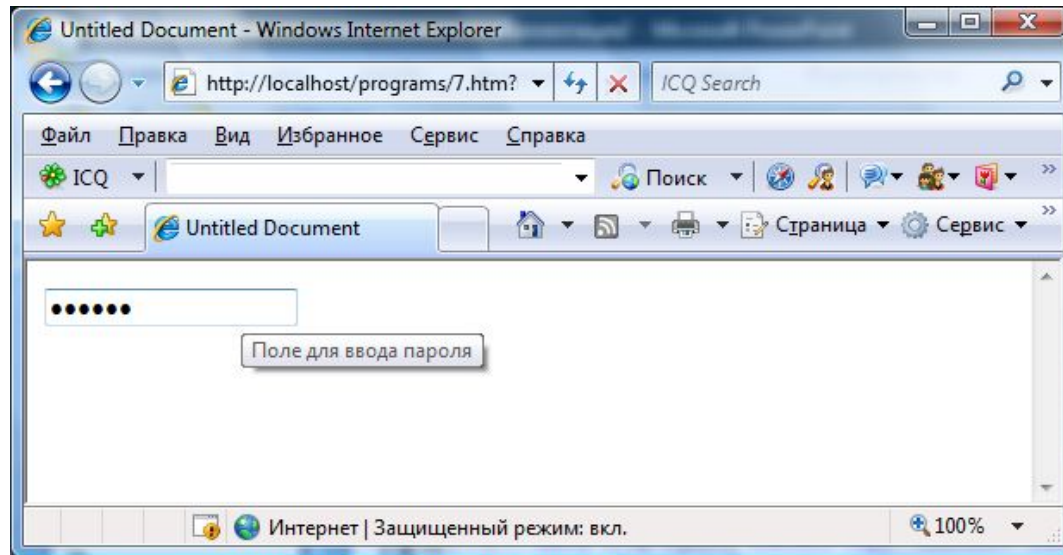
```
<input  
  type=image  
  name="имя элемента"  
  src="путь к файлу-рисунку"  
  title="всплывающая подсказка"
```

>



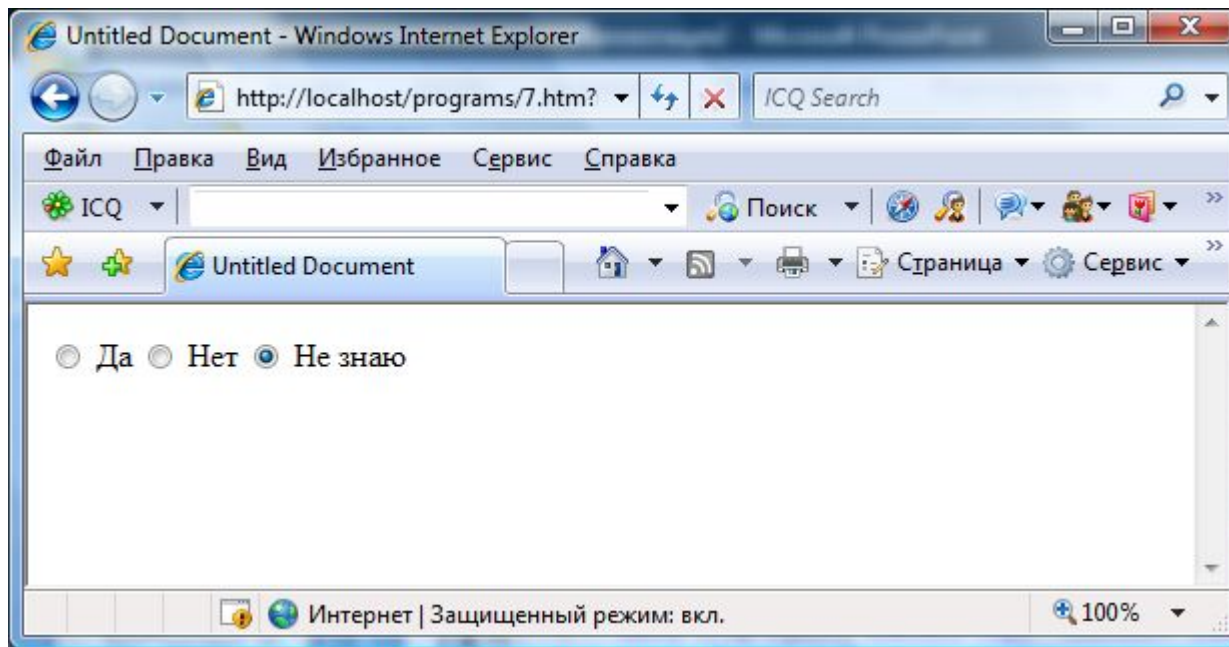
Поле для ввода пароля

```
<input  
  type="password"  
  name="имя элемента"  
  value="пароль"  
  title="всплывающая подсказка"  
  size=длина  
  maxlength="максимальное количество символов">
```



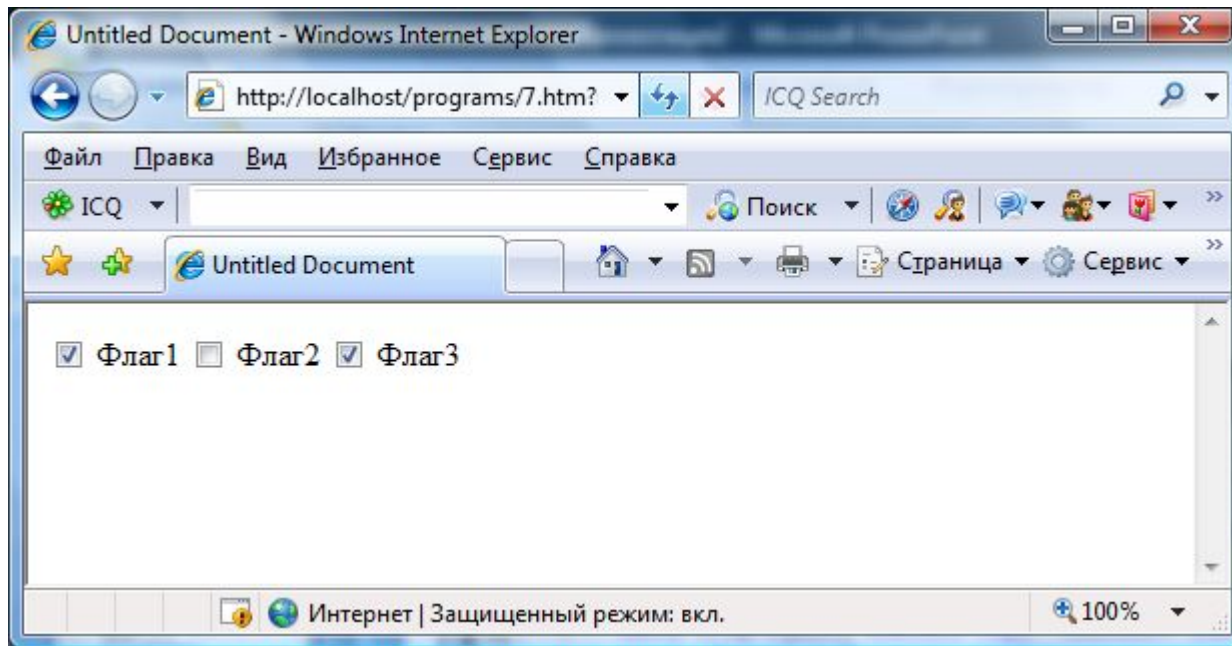
Переключатель (Radiobutton)

```
<input  
  type=radio  
  name="имя элемента"  
  value="значение"  
  [checked]>  
Текст для переключателя
```



Флаг (checkbox)

```
<input  
  type= checkbox  
  name="имя элемента"  
  value="значение"  
  [checked]>  
Текст флага
```

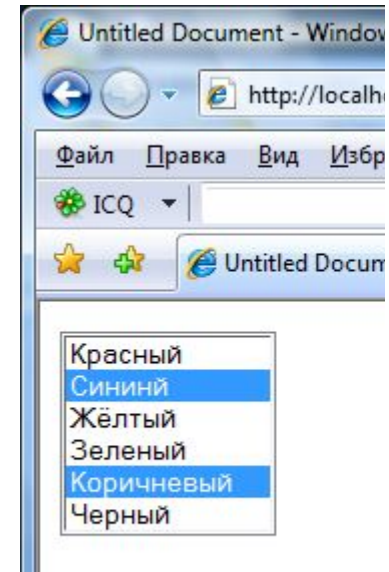
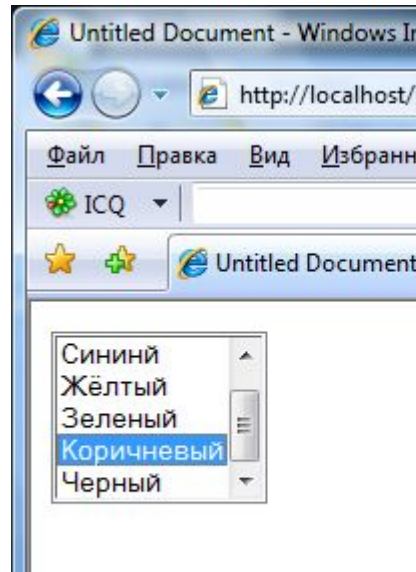
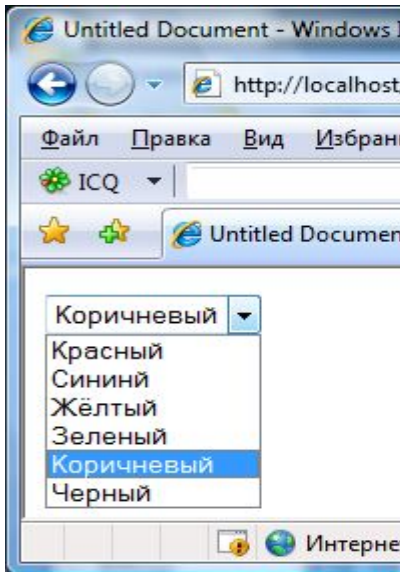


Скрытое поле

```
<input  
  type= hidden  
  name="имя элемента"  
  value="значение"  
>
```

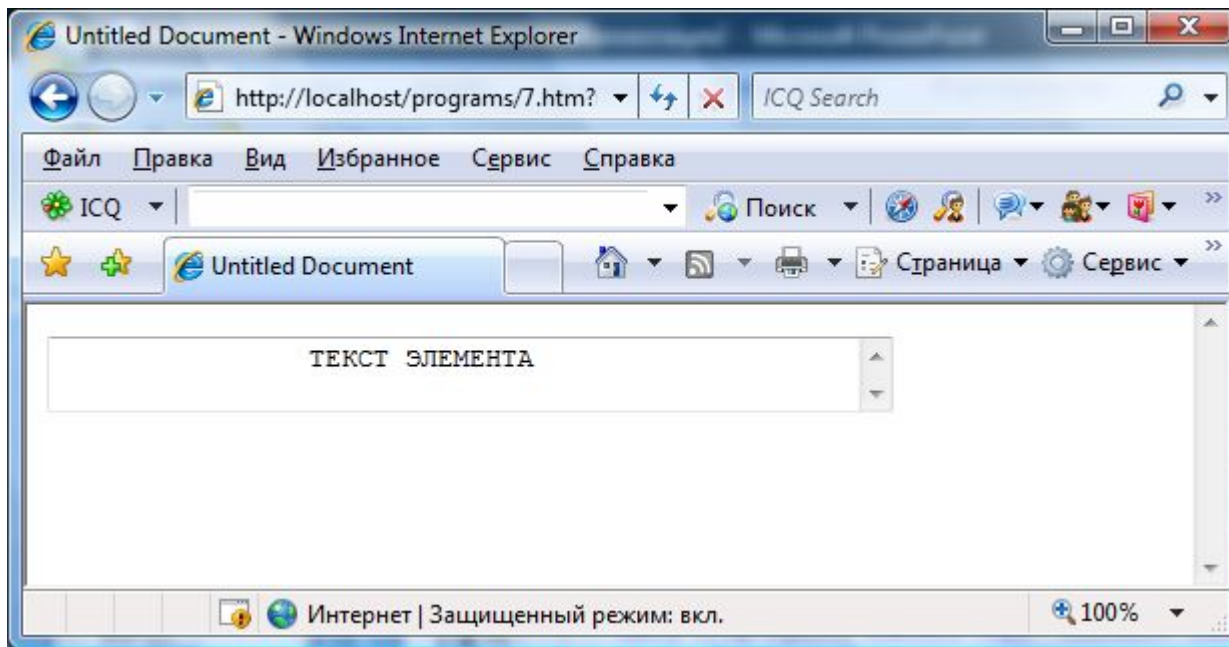
Список

```
<select  
  size="количество видимых элементов"  
  name="имя списка"  
  [multiple]>  
  <option value="значение1" [selected]>Отображаемая строка 1</option>  
  ...  
  <option value="значениеN" [selected]>Отображаемая строка N</option>  
</select>
```



Поле для ввода многострочного текста (TextArea)

```
<textarea  
    name="имя элемента"  
    rows=высота  
    cols=длина>  
    ТЕКСТ ЭЛЕМЕНТА  
</textarea>
```



Вставка изображений

```
<html>
  <body>
    <p> Изображение:
      
    </p>
    <p> Изображение из другой папки:  </p>
    <p> Изображение с сайта csru.ru:  </p>
  </body>
</html>
```

Графические форматы

- **JPEG – Joint Photographic Experts Group**
- Формат JPEG был разработан для передачи фотографий между различными платформами. Благодаря поддержке 24-битовой цветовой палитры (миллионы цветов) формат JPEG получил широкое распространение для отображения иллюстраций высокого качества. При этом применяемый в JPEG алгоритм сжатия позволяет варьировать размер загружаемого файла в зависимости от требуемого качества отображения иллюстрации.
- Кроме того, формат JPEG поддерживает опцию "прогрессивный" JPEG, которая во многом похожа со свойством чередования строк формата GIF, но реализована на совершенно других принципах. В файл JPEG записывается несколько иллюстраций полного размера, но различного уровня качества. Сначала отображается рисунок низкого качества (и, соответственно, малого объема), затем догружаются все более качественные изображения. При этом общий объем загружаемого файла возрастает несущественно по сравнению с оригинальным JPEG.

Графические форматы

- **PNG – Portable Network Graphics**
- В последнее время все большее распространение получает формат PNG, который был разработан для сетевых иллюстраций с целью заменить форматы JPEG и GIF и объединивший в себе все их преимущества.
- Формат PNG поддерживает 24-, 32- и 48-битовую цветовую палитру, 8- и 16- битовую палитру серого, 8-битовую пользовательскую палитру. Реализованный в PNG алгоритм сжатия позволяет сжимать изображения лучше чем GIF (от 5 до 25%). Для него реализован механизм чересстрочного вывода первого изображения с использованием двумерной чересстрочности, что позволяет обеспечить минимальные затраты времени на вывод первого изображения.
- Кроме того, в PNG реализован эффективный механизм контроля целостности файла, позволяющего обнаружить ошибки при передаче по сети. Хотя PNG и поддерживает прозрачность, такие изображения некорректно отображаются в некоторых версиях MS Internet Explorer, поэтому если вы хотите, чтобы во всех браузерах рисунок с прозрачным фоном отображался корректно, лучше использовать формат GIF.

PNG в IE6



Выравнивание изображений

Для выравнивания изображений используется атрибут `align`.

- **`align=top`** – изображение выравнивается по верхнему краю текущей текстовой строки, не меняя позиции по горизонтали. При этом речь идет как о тексте, так и о графике;
- **`align=middle`** – изображение центрируется по вертикали на базовой линии текущей текстовой строки, не меняя позицию по горизонтали;
- **`align=bottom`** – нижний край изображения выравнивается по горизонтали на базовой линии текущей текстовой строки;
- **`align=left`**– изображение смещается к левому краю рабочей зоны, последующий текст сразу же начинает "обтекать" графику;
- **`align=right`**– то же что и для `left`, только изображение смещается к правой части рабочей зоны.

```
<html>
  <body>
    <p>Изображение выравнено по верху
    
    при помощи align="top"</p>

    <p>Значение "middle" атрибута align
    
    центрирует изображение по вертикали</p>

    <p>Выравнивание изображения по нижнему краю
    
    (align="bottom") используется по умолчанию (можно сравнить с предыдущим
    примером).</p>

    <p>В этом примере изображение
    
    смещено влево и начиная со следующей строки текст как бы обтекает вставленный
    рисунок.
    При таком способе выравнивания желательно помещать изображение в начале
    параграфа.</p>

    <p>
    Изображение прижато к правому краю рабочей области (align="right").
    Текст занимает все свободное пространство слева от изображения обтекая
    его.</p>
  </body>
</html>
```

- Кроме того, для форматирования изображений внутри текста используются атрибуты **hspace** и **vspace**, которые определяют интервал между графическим изображением и обтекающим текстом по горизонтали и вертикали соответственно. Размер отступов задается в **пикселях**.

- Изображение в HTML может быть обрамлено прямоугольной рамкой. Для этого используется атрибут `border`, который задает ширину рамки вокруг изображения в пикселях.

```
<p> В
данном варианте толщина рамки
установлена в 5 пикселей. </p>
```

- В последнее время атрибут `border = 0` при разработке Web-сайтов в подавляющем большинстве случаев используется именно для отмены рамки вокруг изображения-гиперссылки.

Размеры изображений

- По умолчанию изображение отображается в браузере с сохранением своих исходных размеров и пропорций. При этом, если изображение превышает размер окна браузера, то оно может либо автоматически уменьшится до размера окна, либо предоставить возможность просмотра по частям с использованием полос прокрутки.
- При помощи атрибутов **height** и **width** (высота и ширина, соответственно) можно задать размеры отображения графического файла на Web-странице.

Атрибут Alt

- В ряде случаев графическое изображение на Web-странице не может быть отражено. Это может происходить при просмотре страниц с использованием текстовых браузеров, при запрете на отображение графики в целях увеличения скорости загрузки и снижения расходов на Интернет, а также в ряде других случаев. При этом смысл страницы может быть существенно искажен, а при использовании графических меню и кнопок навигация по сайту будет значительно затруднена. Во избежание этого в HTML используется атрибут alt, который задает цепочку символов (максимальная длина 1024 символа), отображаемую в браузере вместо изображения и в произвольной форме описывающую его (заменяющий текст).

```
<html>
  <body>
    
    <br> Для демонстрации работы
атрибута alt необходимо в свойствах
обозревателя запретить использование
графики на странице. <br>
    Обратите внимание, что даже при
отображенной графике замещающий текст
появляется на экране при наведении
указателя мыши на изображение.
Благодаря этому атрибут alt можно
использовать и в качестве
своеобразной подписи к рисунку.
  </body>
</html>
```


Карты изображений

- Наряду с использованием изображений в качестве иллюстраций к тексту или элементов оформления Web-страницы, в HTML предусмотрена возможность создания карт изображений (imagemap), отдельные области которого могут являться гиперссылками на другие разделы или страницы Web-сайта. В общем виде это соответствует использованию изображения в качестве гиперссылки с тем отличием, что на одной карте изображений можно создать несколько несовпадающих областей, и соответственно, гиперссылок.

- Применение таких карт изображений удобно для создания географических справочных систем, путеводителей, карт погоды. Также карты изображений находят широкое применение при создании сложных графических меню.
- В HTML предусмотрены два варианта обработки карт изображений: обработка карты изображения непосредственно браузером и передача на сервер координат указателя мыши для дальнейшей обработки.

Автономные (обрабатываемые браузером) карты изображений описываются с помощью атрибута usemap тега :

```

```

Где "figure_1.png" - имя файла, содержащего изображение, "#coordinates" - ссылка на часть HTML-документа, описывающего координатные области.

Координатные области карты изображений определяются при помощи тегов <map> и <area>.

- Тег <area> описывает координаты отдельной области изображения, параметры которой и адрес гиперссылки задаются при помощи атрибутов shape, coords и href.
- Атрибут shape определяет форму области-ссылки. По умолчанию ему присваивается значение shape="rect"-прямоугольник. Также область может быть описана в форме окружности (shape="circle") или многоугольника (shape="poly").
- Атрибут coords определяет размеры области. В зависимости от типа размечаемой области может меняться значение этого атрибута. Так для shape="rect" указываются две пары координат (левого верхнего и правого нижнего угла прямоугольника) в пикселях. Для shape="circle" указываются координаты центра окружности и ее радиус, а для shape="poly" последовательно указываются координаты вершин многоугольника.
- Атрибут href задает URL-адрес ссылки для перехода в случае щелчка мышью на выбранной области.

Карта изображений, на которой в соответствии с изображением выделены прямоугольная, круглая и треугольная координатная области.

```
<html>
  <body>
    
    <map name="coordinates">
      <area coords="130,194,270,130"
href="rect.html" alt="Прямоугольник">
<area shape="circle" coords="73,90,40"
href="circle.html" alt="Круг">
      <area shape="poly"
coords="168,109,280,109,223,13"
href="polygon.html" alt="Полигон
(треугольник)">
    </map>
  </body>
</html>
```

Создание графических меню

```
<html>
  <body>
    <table cellpadding='0' cellspacing='0'>
<tr><td><a href=""></a></td></tr>
    <tr><td><a href=""></a></td></tr>
    <tr><td><a href=""></a></td></tr>
    <tr><td><a href=""></a></td></tr>
    <tr><td><a href=""></a></td></tr>
  </table>
</body>
</html>
```