

HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Страница</title>
    <meta http-equiv="Content-Type"
      content="text/html; charset=utf-8">
    <meta name="description" content="Сайт">
    <link rel="stylesheet" href="./style.css">
  </head>
  <body id="the_body">
    <p class="article">...</p>
    <script src="./script.js"></script>
  </body>
</html>
```

Особенности HTML разметки

- Произвольный регистр: `
 ==
`
- Атрибуты без двойных кавычек: `color=red`
- Сокращенные атрибуты: `disabled`
- Непарные тэги: `<p>` вместо `<p></p>`
- Перестановки тэгов: `<i></i>`
- «Свои» тэги: `<magic></magic>`

Особенности XHTML разметки

- Только нижний регистр тэгов
- Атрибуты в двойных кавычках: `color="red"`
- Атрибуты – без сокращений: `disabled="disabled"`
- Тэги всегда парные: `<p></p>`
- Строгая вложенность: `<i></i>`
- `id` вместо `name` в якорях `#`
- Необходим `DOCTYPE`

DOCTYPE

DOCTYPE уточняет тип содержимого, указывает HTML парсеру как правильно разбирать данный документ.

XHTML 1.1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"  
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

DOCTYPE

HTML 4 Transitional

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01  
Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

HTML5

```
<!DOCTYPE html>
```

HTML тэги

Тэги верхнего уровня

- `html` - обертка
- `head` - заголовок страницы, не отображается
- `body` - тело страницы, то, что видит пользователь

Тэги внутри head

- `title` - отображается в заголовке окна браузера
- `meta` - содержит информацию для user-agent'ов

```
<meta name="description" content="Для друзей">
```

```
<meta http-equiv="Content-Type" content="text/html;  
charset=utf-8">
```

Тэги внутри head (2)

- `link` - указание связанных ресурсов, например CSS

```
<link rel="stylesheet" href="/style.css">
```

```
<link rel="alternate" href="/news.rss"
```

```
type="application/rss+xml">
```

- `script` - загрузка JavaScript

```
<script src="./jquery.js" charset="windows-1251" async
```

```
defer></script>
```

Рекомендации по link и script

Загрузку CSS (тэг `link`) рекомендуется ставить в тэге `head`, а загрузку JavaScript (тэг `script`) - наоборот ближе к концу страницы. Это повышает скорость отрисовки страницы.

Блочные и
строчные тэги

Блочные тэги

- `h1` - `h6` — различные уровни заголовков
- `p` — разбиение текста на параграфы
- `hr` — горизонтальная линия
- `pre` — блок преформатированного кода, например исходный код
- `blockquote` — цитирование длинного блока текста
- `div` — абстрактный блочный контейнер

Пример

Верстка этого слайда сделана в HTML и является отличным примером использования блочных тэгов.

Подзаголовков

Хотя из-за стилей оформления презентации это не так очевидно.

Строчные тэги

- `a` — гиперссылки
- `em`, `i` — акцентирование
- `strong`, `b` — выделение
- `img` — вставка изображений
- `sub` — нижний индекс
- `sup` — верхний индекс
- `span` — абстрактный строчный контейнер

Списки и таблицы

Списки в HTML

`ul`, `ol`, `li` - маркированные списки

```
<ul>  
  <li><code>a</code> – гиперссылки</li>  
  <li><code>em</code> – акцентирование</li>  
</ul>
```

Таблицы в HTML

```
<table>
  <caption>квартальный отчет</caption>
  <thead>
    <tr>
      <td>дата</td>
      <td colspan="2">доход</td>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th rowspan="2">2011-01-01</th>
      <td>100500</td> <td>33</td>
    </tr>
    <tr>
      <td>100</td> <td>500</td>
    </tr>
  </tbody>
</table>
```

Таблицы в HTML

квартальный отчет

дата	доход	
2011-01-01	100500	33
	100	500

Гиперссылки

Гиперссылки

```
<a href="http://duckduckgo.com" target="_blank">  
      
</a>
```

- `href` - URL гиперссылки
- `target` - в каком окне открывать ссылку
- `name` - имя якоря, вместо `href`

Действия браузера при переходе

Поведение браузера зависит от протокола в URL

- `http`, `https`, `ftp` - переход по ссылке
- `mailto` - запуск почтового клиента
- `javascript` - выполнение JavaScript кода
- `#anchor` - прокрутка текущей страницы

```
<a name="anchor"></a>
```

```
<div id="anchor"></div>
```

ФОРМЫ

Формы

```
<form method="POST" action="/add/"  
    enctype="multipart/form-data" target="frame3">  
  <input name="image" type="file">  
  <input name="id" type="hidden" value="3">  
  <input name="nick" type="text">  
  <input type="submit" value="Отправить">  
  <button name="action" value="more">  
    Все равно отправить  
  </button>  
</form>
```


Аттрибуты формы

- `action` - URL, на который будет отправлена форма
- `method` - HTTP метод, GET или POST
- `target` - имя окна браузера, в котором открыть результат
- `enctype` - способ кодирования данных формы. По умолчанию - `application/x-www-form-urlencoded`. Если форма содержит поля для ввода файлов, то должно быть `multipart/form-data`

application/x-www-form-urlencoded

Это способ кодирования данных формы для передачи через URL.

Допустим есть форма со следующими данными:

id: 3

name: Вася

friend: [4, 5]

В закодированном виде это будет выглядеть так:

```
id=3&name=%D0%92%D0%B0%D1%81%D1%8F&friend=4&friend=5
```

Поля ввода
данных

Элементы ввода формы

- `input` - универсальное поле, может быть:
 - `type="hidden"` - невидимое
 - `type="text"` - текстовое поле
 - `type="checkbox"` - checkbox, переключатель да/нет
- `button` - кнопка
- `textarea` - многострочное поле ввода
- `select`, `option` - выпадающий список

Атрибуты элементов ввода

```
<input type="text" name="username"  
       value="" placeholder="Вася" autocomplete="off">
```

- `type` - определяет внешний вид и функционал
- `name` - имя, с которым данный элемент попадет в запрос
- `value` - начальное значение, пользователь может изменить
- `placeholder` - подсказка для пользователя

Подробное руководство

htmlbook.ru