

# Разработка и поддержка кроссплатформенных приложений

02.03.03 - Математическое обеспечение и администрирование информационных систем, направленность (профиль) -  
разработка и администрирование информационных систем

<http://vikchas.ru>

## Лекция 2 и задание работы «Кроссплатформенные приложения – основные понятия и интерпретация HTML 5»

**Часовских Виктор Петрович**

доктор технических наук,  
Профессор кафедры ШИиКМ  
ФГБОУ ВО «Уральский государственный  
экономический университет

Екатеринбург 2023

# Термины и определения

HTML (HyperText Markup Language) представляет язык разметки гипертекста, используемый преимущественно для создания документов в сети интернет. HTML начал свой путь в начале 90-х годов как примитивный язык для создания веб-страниц, и в настоящий момент уже трудно представить себе интернет без HTML.

Подавляющее большинство сайтов так или иначе используют HTML.

В 2014 году официально была завершена работа над новым стандартом - HTML5, который фактически произвел революцию, привнеся в HTML много нового.

Фактически с добавлением новых функций HTML5 стал не просто новой версией языка разметки для создания веб-страниц, но и фактически платформой для создания приложений, а область его использования вышла далеко за пределы веб-среды интернет: HTML5 применяется также для создания мобильных приложений под Android, iOS, Windows Mobile и даже для создания десктопных приложений для обычных компьютеров (в частности, в ОС Windows 8/8.1/10).

В итоге, как правило, HTML 5 применяется преимущественно в двух значениях:

HTML 5 как обновленный язык разметки гипертекста, некоторое развитие предыдущей версии HTML 4

HTML 5 как мощная платформа для создания веб-приложений, которая включает не только непосредственно язык разметки гипертекста, обновленный HTML, но и язык программирования JavaScript и каскадные таблицы стилей CSS 3.

## Создание документа HTML5

Для создания документа HTML5 нам нужны в первую очередь два элемента: DOCTYPE и html. Элемент **doctype** или Document Type Declaration сообщает веб-браузеру тип документа. `<!DOCTYPE html>` указывает, что данный документ является документом html и что используется html5, а не html4 или какая-то другая версия языка разметки.

А элемент html между своим открывающим и закрывающим тегами содержит все содержимое документа.

Внутри элемента `html` мы можем разместить два других элемента: **head** и **body**. Элемент `head` содержит метаданные веб-страницы - заголовок веб-страницы, тип кодировки и т.д., а также ссылки на внешние ресурсы - стили, скрипты, если они будут использоваться. Элемент `body` собственно определяет содержимое `html`-страницы.

Теперь изменим содержимое файла `index.html` следующим образом:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    <title>Документ HTML5</title>
```

```
  </head>
```

```
  <body>
```

```
    <div>Содержание документа HTML5</div>
```

```
  </body>
```

```
</html>
```

В элементе `head` определено два элемента:

элемент `title` представляет заголовок страницы

элемент `meta` определяет метаинформацию страницы.

Для корректного отображения символов предпочтительно указывать кодировку.

В данном случае с помощью атрибута `charset="utf-8"` указываем кодировку `utf-8`. Сохраним и откроем файл `index.html` в браузере:

В пределах элемента элемента `body` используется только один элемент - `div`, который оформляет блок. Содержимым этого блока является простая строка.

Поскольку мы выбрали в качестве кодировки `utf-8`, то браузер будет отображать веб-страницу именно в этой кодировке. Однако необходимо чтобы сам текст документа также соответствовал выбранной кодировке `utf-8`.

Как правило, в различных текстовых редакторах есть соответствующие настройки для установки кодировки.



# HTML Documents

Все HTML-документы должны начинаться с объявления типа документа: `<!DOCTYPE html>` .

Сам HTML-документ начинается с `<html>` и заканчивается `</html>` .

Видимая часть HTML-документа находится между `<body>` и `</body>` .

## Пример

```
<!DOCTYPE html>
<html>
<body>

<h1>Мой первый заголовок</h1>
<p>Мой первый абзац.</p>

</body>
</html>
```

## СКРИНШОТ

# Заголовки HTML

Заголовки HTML определяются с помощью `<h1>` для `<h6>` тегов.

`<h1>` определяет наиболее важный заголовок. `<h6>` определяет наименее важный заголовок:

## Example

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
```

## СКРИНШОТ

# Абзацы HTML

Абзацы HTML определяются тегом `<p>`:

## Example

```
<p>This is a paragraph.</p>  
<p>This is another paragraph.</p>
```

СКРИНШОТ

## Ссылки HTML

HTML-ссылки определяются тегом `<a>`:

### Example

```
<a href="https://html5css.ru">This is a link</a>
```

Просмотр демо в редакторе

Назначение ссылки указано в атрибуте `href`.

Атрибуты используются для предоставления дополнительных сведений о элементах HTML.

## Изображения в формате HTML

HTML-образы определяются тегом `<img>`.

Исходный файл (`src`), альтернативный текст (`alt`), `width` и `height` предоставляются в качестве атрибутов:

### Пример

```

```

СКРИНШОТ

## Кнопки HTML

HTML-кнопки определяются тегом `<button>`:

### Example

```
<button>Click me</button>
```

## СКРИНШОТ

# Списки HTML

Списки HTML определяются `<ul>` (неупорядоченный/маркированный список) или тегом `<ol>` (упорядоченный/нумерованный список), за которым следуют теги `<li>` (элементы списка):

## Пример

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

## СКРИНШОТ

# HTML Элементы (элементы)

## Элементы HTML

HTML-элемент обычно состоит из тега **Start** и тега **End** с содержимым, вставленным между:

```
<tagname>Содержание идет здесь...</tagname>
```

HTML- элемент — это все, от начального тега до конечного тега:

```
<p>Мой первый абзац.</p>
```

Начальный тег	Содержимое элемента
<h1>	Мой первый заголовок

```
<p>
```

Мой первый абзац.

```
<br>
```

Элементы HTML без содержимого называются пустыми элементами. Пустые элементы не имеют конечного тега, такого как `<br>` элемент (который указывает на разрыв строки).

## Вложенные элементы HTML

Элементы HTML могут быть вложенными (элементы могут содержать элементы).

Все HTML-документы состоят из вложенных элементов HTML.

Этот пример содержит четыре элемента HTML:

### Пример

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

### СКРИНШОТ

### Пример объяснения

`<html>` элемент определяет **весь документ**.

Он имеет тег **Start** `<html>` и тег **End** `</html>`.

Элемент **Content** — это другой элемент HTML ( `<body>` элемент).

```
<html>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>
```

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

`<body>` элемент определяет **тело документа**.

Он имеет тег **Start** `<body>` и тег **End** `</body>`.

Элемент **Content** — это два других HTML-элемента ( `<h1>` и `<p>` ).

```
<body>
```

```
<h1>My First Heading</h1>
<p>My first paragraph.</p>
```

```
</body>
```

`<h1>` элемент определяет **заголовок**.

Он имеет тег **Start** `<H1>` и тег **End** `</H1>`.

Элемент **Content**: мой первый заголовок.

```
<h1>My First Heading</h1>
```

The `<p>` element defines a **paragraph**.

It has a **start** tag `<p>` and an **end** tag `</p>`.

The element **content** is: My first paragraph.

```
<p>My first paragraph.</p>
```

## Не забывайте, что конечный тег

Некоторые элементы HTML будут отображаться корректно, даже если вы забыли конечный тег:

### Пример

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

```
<p>This is a paragraph
<p>This is a paragraph
```

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

## СКРИНШОТ

Приведенный выше пример работает во всех браузерах, так как закрывающий тег считается необязательным.

**Никогда не полагайтесь на это. Если вы забыли конечный тег, это может привести к неожиданным результатам и/или ошибкам.**

---

## Пустые элементы HTML

Элементы HTML без содержимого называются пустыми элементами.

`<br>` пустой элемент без закрывающего тега (тег `<br>` определяет разрыв строки).

Пустые элементы могут быть "закреты" в открывающем теге так: `<br />`.

HTML5 не требует закрытия пустых элементов. Но если требуется более строгая проверка, или если необходимо сделать документ удобочитаемым синтаксическими анализаторами XML, необходимо правильно закрыть все элементы HTML.

---

## Использовать строчные теги

Теги HTML не чувствительны к регистру: `<p>` означает то же, что и `<P>`.

Стандарт HTML5 не требует строчных тегов, но W3C **рекомендует** строчные буквы в HTML и **требует** нижнего регистра для более строгих типов документов, таких как XHTML.

В [html5css.ru](http://html5css.ru) Мы всегда используем строчные теги.

## HTML Attributes

Атрибуты предоставляют дополнительную информацию о HTML-элементах.

# Атрибуты HTML

- Все элементы HTML могут иметь **атрибуты**
- Атрибуты предоставляют **Дополнительные сведения** об элементе
- Атрибуты всегда указываются в **начальном теге**
- Атрибуты обычно поставляются в парах "имя/значение", например: **name="value"**

## Атрибут href

HTML-ссылки определяются тегом `<a>`. Адрес ссылки указан в атрибуте `href` :

### Пример

```
<a href="https://html5css.ru">Это ссылка</a>
```

**СКРИНШОТ**

## Атрибут src

HTML-образы определяются тегом `<img>`.

Имя файла источника изображения указывается в атрибуте `src` :

### Example

```

```

**СКРИНШОТ**

---

## Атрибуты ширины и высоты

Изображения в HTML имеют набор атрибутов **size**, который определяет ширину и высоту изображения:

### Пример

```

```

## СКРИНШОТ

Размер изображения указан в пикселях: width="500", означает 500 пикселей в ширину.

## The alt Attribute

Атрибут `alt` указывает альтернативный текст, который будет использоваться, когда изображение не может быть отображено.

Значение атрибута может быть прочитано программами чтения с экрана. Таким образом, кто-то "прослушивание" на веб-странице, например, слепой человек, может "слышать" элемент.

### Пример

```

```

## СКРИНШОТ

Атрибут `alt` также полезен, если изображение не существует:

### Пример

Посмотрите, что произойдет, если мы попытаемся отобразить изображение, которое не существует:

```

```

## Атрибут style

Атрибут `style` используется для указания стиля элемента, например цвета, шрифта, размера и т. д.

### Пример

```
<p style="color:red">I am a paragraph</p>
```



# Атрибут lang

Язык документа может быть объявлен в теге `<html>` .

Язык объявляется с атрибутом `lang`.

Объявление языка важно для приложений специальных возможностей (программы чтения с экрана) и поисковых систем:

```
<!DOCTYPE html>  
<html lang="en-US">  
<body>
```

...

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

Первые две буквы указывают язык (EN). Если есть диалект, используйте еще две буквы (US).

# Атрибут Title

Здесь атрибут `title` добавляется к элементу `<p>` . Значение атрибута Title будет отображаться в виде подсказки при наведении указателя мыши на абзац:

## Пример

```
<p title="I'm a tooltip">  
This is a paragraph.  
</p>
```

СКРИНШОТ

# Мы предлагаем: использование атрибутов нижнего регистра

Для стандарта HTML5 не требуются имена атрибутов в нижнем регистре.

Атрибут Title может быть написан с прописными или строчными буквами, например **Title** или **title**.

W3C **рекомендует** строчные буквы в HTML и **требует** строчные для более строгих типов документов, таких как XHTML.

В html5css.ru всегда используем строчные имена атрибутов.

---

## Мы рекомендуем: цитировать значения атрибутов

Стандарт HTML5 не требует кавычек вокруг значений атрибутов.

**href** атрибут, показанный выше, *может* быть написан без кавычек:

### Вад

```
<a href=https://html5css.ru>
```

СКРИНШОТ

### Хорошо

```
<a href="https://html5css.ru">
```

СКРИНШОТ

W3C **рекомендует** котиловки в HTML и **требует** цитаты для более строгих типов документов, таких как XHTML.

Иногда **необходимо** использовать кавычки. Этот пример не будет отображать атрибут Title правильно, поскольку он содержит пробел:

### Пример

```
<p title>About Html5css.ru>
```

СКРИНШОТ

Наиболее часто используются кавычки. Пропуск кавычек может привести к ошибкам.

В html5css.ru мы **всегда** используем кавычки вокруг значений атрибутов.

---

## Одинарные или двойные кавычки?

Двойные кавычки вокруг значений атрибутов являются наиболее распространенными в HTML, но также можно использовать одинарные кавычки.

В некоторых ситуациях, когда само значение атрибута содержит двойные кавычки, необходимо использовать одинарные кавычки:

```
<p title='John "ShotGun" Nelson'>
```

Или наоборот:

```
<p title="John 'ShotGun' Nelson">
```

**СКРИНШОТ**

# Благодарю за внимание!

