



Конспект «Возможности HTML и CSS»

HTML5

Чтобы использовать HTML5, нужно задать правильный тип документа:

```
<!DOCTYPE html>
```

header и footer

`<header>` — это не только привычная шапка сайта с логотипом и меню, но и «шапка» какой-нибудь статьи или раздела сайта, в которой могут содержаться заголовки, оглавление и так далее.

`<footer>` — это подвал сайта: с копирайтами, контактной информацией и так далее. Но может использоваться и в других разделах сайта, например в статье, чтобы разместить дополнительную информацию: данные об авторе, дополнительные ссылки и так далее.

main

Тег `<main>` обозначает основное содержание сайта и по спецификации может использоваться на странице только один раз.

article и section

Где и когда использовать разные контейнеры:

- `<div>` — контейнер общего назначения, не обязательно смысловой. Дивы используются для разметки мелких блоков, создания сетки и декоративных эффектов.
- `<section>` — более крупный логический контейнер, объединяющий содержание по смыслу. Например, блок «О компании», список товаров, раздел личной информации в профиле и так далее.
- `<article>` — самостоятельный, цельный и независимый раздел документа. Этот раздел можно в неизменном виде использовать в различных местах, в том числе и на других сайтах. Примеры: статья, пост в блоге, сообщение на форуме и так далее.

nav

Тег `<nav>` — специальный элемент для навигационных блоков. Меню и блоки ссылок лучше оборачивать не в дивы, а в навы.

aside

`<aside>` — это дополнительное содержание, не связанное напрямую с основным. Ещё такие блоки часто называют «сайдбарами» или боковыми панелями.

time

`<time>` — специальный тег для обозначения даты и времени. У этого тега есть атрибут

`datetime` — для указания даты в машиночитаемом формате ISO 8601. Содержимое `datetime` не отображается.

`<time datetime="2014-04-20">Вчера</time>` мы готовили тренажёр к публикации.

figure и figcaption

`<figure>` — цельный и независимый блок содержания. Внутри этого тега размещают демонстрационный материал: изображения, схемы, куски кода и так далее.

`<figcaption>` — разъясняющий комментарий, или «легенда». Размещается первым или последним элементом внутри `<figure>`. Пример:

```
<figure>
  схема,
  график,
  диаграмма
  и так далее
  <figcaption>Легенда</figcaption>
</figure>
```

video

Парный тег `<video>` предназначен для вставки видео. Его основные атрибуты:

- `width` и `height` — задают ширину и высоту видео;
- `controls` — пустой атрибут, при наличии которого отображается панель управления видео;
- `preload` — задаёт режим предзагрузки видео, имеет 3 возможных значения:
 - `none` — не загружать ничего;

- `metadata` — загрузить служебную мета-информацию (длительность, первый кадр и так далее);
- `auto` — можно загрузить всё видео.
- `src` — задаёт адрес видеофайла;
- `autoplay` — пустой атрибут, при наличии которого воспроизведение видео начинается автоматически;
- `poster` — задаёт адрес картинки-обложки, которая отображается, когда видео ещё не загрузилось или не воспроизводится.

Разные браузеры поддерживают разные форматы, поэтому лучше указывать адреса файлов в нескольких форматах. Делается это с помощью тегов `<source>`:

```
<video controls>  
  <source src="video.mp4" type="video/mp4">  
  <source src="video.ogv" type="video/ogg">  
  <source src="video.webm" type="video/webm">  
</video>
```

В атрибуте `src` указывается адрес видеофайла, а в атрибуте `type` его тип (также там могут указываться и кодеки). Браузер выбирает из списка видеофайлов первый, который может проиграть, и загружает его.

audio

Тег `<audio>` предназначен для вставки звука. Его основные атрибуты:

- `controls` — пустой атрибут, при наличии которого отображается панель управления проигрывателем;
- `preload` — задаёт режим предзагрузки аудио, имеет 3 возможных значения:
 - `none` — не загружать ничего;
 - `metadata` — загрузить служебную мета-информацию;
 - `auto` — можно загрузить весь файл.
- `src` — задаёт адрес аудиофайла;
- `autoplay` — пустой атрибут, при наличии которого воспроизведение звука начинается автоматически.

Лучше указывать адреса файлов в нескольких форматах. Для этого используют теги `<source>`:

```
<audio controls>  
  <source src="sound.mp3" type="audio/mpeg">
```

```
<source src="sound.oga" type="audio/ogg">
</audio>
```

Шрифты

Есть возможность подключать и использовать на странице любые нестандартные шрифты. Подключение веб-шрифтов производится с помощью CSS-правила `@font-face`. Например:

```
@font-face {
  font-family: "Roboto";
  src:
    local("Roboto Regular"),
    url("roboto.woff") format("woff");
}
```

Нужно указать название шрифта, которое вы будете использовать в `font-family`, и источники, из которых браузер сможет загрузить шрифт. Сначала указывают название шрифта в системе, чтобы браузер попытался найти его локально, а затем указывают адрес файла шрифта в интернете.

Если шрифт очень редкий и его нет ни в одном из шрифтовых сервисов, поступают так:

1. Берут файл шрифта (например, `.ttf`) и конвертируют в веб-формат в сервисе наподобие [Font Squirrel](#) (для кириллических шрифтов надо указать дополнительные параметры конвертации).
2. Затем сконвертированные файлы шрифта размещают у себя на сервере.
3. И подключают шрифт с помощью `@font-face`.

SVG

SVG — векторный формат, который хорошо поддерживается браузерами. Векторные изображения можно уменьшать и увеличивать без потери качества.

Продолжить



Практикум

Тренажёры

Профессии

Фронтенд-разработчик

Подписка

Для команд и компаний

Учебник по PHP

React-разработчик

Фулстек-разработчик

Бэкенд-разработчик

Курсы

HTML и CSS. Профессиональная вёрстка сайтов

HTML и CSS. Адаптивная вёрстка и автоматизация

JavaScript. Профессиональная разработка веб-интерфейсов

JavaScript. Архитектура клиентских приложений

React. Разработка сложных клиентских приложений

PHP. Профессиональная веб-разработка

PHP и Yii. Архитектура сложных веб-сервисов

Node.js. Разработка серверов приложений и API

Анимация для фронтендеров

Вёрстка email-рассылок

Vue.js для опытных разработчиков

Регулярные выражения для фронтендеров

Шаблонизаторы HTML

Алгоритмы и структуры данных

Анатомия CSS-каскада

Блог

С чего начать

Шпаргалки для разработчиков

Отчеты о курсах

Информация

Об Академии

О центре карьеры

Соглашение

Конфиденциальность

Сведения об образовательной организации

Лицензия № 3026

Услуги

Работа наставником

Для учителей

Стать автором

Остальное

Написать нам

Мероприятия

Форум

