

Профессиональная вёрстка

# Урок 4

## Практическое применение Bootstrap

Добавление аудио и видео.

## [Bootstrap](#)

[Что такое фреймворк \(framework\)](#)

[Что такое Bootstrap](#)

[Из чего состоит фреймворк](#)

[Сетка Bootstrap](#)

[Типографика](#)

[Выводы](#)

## [HTML5 аудио и видео](#)

[Аудио HTML5](#)

[Воспроизведение видео с помощью элемента <video>](#)

[Элемент <source>](#)

[Элемент <track>](#)

[Атрибут src](#)

[Структура файла .srt](#)

[Атрибут srclang](#)

[Атрибут kind](#)

[Субтитры \(subtitles\)](#)

[Подписи \(captions\)](#)

[Описания \(descriptions\)](#)

[Метаданные \(metadata\)](#)

[Главы \(chapters\)](#)

[Атрибут label](#)

[Атрибут default](#)

[Полезность медиадорожек для поисковой оптимизации](#)

[Практическое задание](#)

[Дополнительные материалы](#)

[Используемая литература](#)

# Bootstrap

## Что такое фреймворк (framework)

При создании нескольких сайтов можно заметить одну особенность: у них очень похожая структура, и некоторые элементы могут быть похожи на разных сайтах. Собрав несколько стилей в один общий, очень удобно и легко создавать новые сайты, если использовать стили, которые написали ранее и применяли для другого ресурса.

Когда над одним проектом работают несколько разработчиков, можно сделать вывод, что необходим один большой набор стилей, которым будут пользоваться все программисты. На данный момент фреймворков достаточно много, поэтому мы рассмотрим один из самых популярных – Bootstrap.

По сути своей фреймворк – это набор готовых инструментов поверх языка, которые решают стандартные рутинные задачи и позволяют сосредоточиться на конкретном проекте.

## Что такое Bootstrap

Bootstrap – это фреймворк для быстрого, удобного и интересного создания сайтов. Он был создан разработчиками компании Twitter, использовался для продуктов компании и первоначально назывался Twitter Bootstrap, но затем стал самостоятельным решением. У него очень много преимуществ перед конкурентами, что делает его популярным у разработчиков разного уровня.

Преимущества Bootstrap:

- Скорость – создание сайтов с данным фреймворком занимает намного меньше времени благодаря большому набору стилей, которые с легкостью вписываются в любую структуру. Ваш сайт всегда будет выглядеть актуально.
- Гибкость – при добавлении новых элементов не смещаются существующие. Можно использовать только те элементы, которые вам необходимы, не обязательно весь сайт сделать только с Bootstrap.
- Изменяемость – правка стилей достигается за счет добавления новых CSS-правил. Если необходимо поменять стиль, который уже добавлен в структуру, вы с легкостью сможете это сделать, не разрушая структур.
- Большое количество шаблонов – специальные шаблоны помогут в считанные минуты разобраться с заданием, которое вам дал заказчик.
- Большое количество разработчиков, которые его используют.
- Отличная документация.

Bootstrap особенно популярен среди тех, кто занимается созданием так называемых лендингов (посадочных / целевых страниц).

## Из чего состоит фреймворк

Данный инструмент, ускоряющий разработку сайта, помогает сэкономить огромное количество времени благодаря готовым решениям и набору компонентов. Следует заметить одну особенность: Bootstrap – это набор из трех фреймворков: CSS, HTML, JS плюс иконочный шрифт.

## Сетка Bootstrap

Один из самых популярных плюсов данного фреймворка – его сетка, так как она является неотъемлемой частью любого макета. Сетка помогает максимально быстро, удобно и правильно расположить блоки на сайте. Они в дальнейшем будут оптимально смотреться на любом устройстве. В данный момент адаптивная верстка – основной запрос у заказчиков. При добавлении префиксов возможно указать, каким образом будут отображаться блоки в зависимости от устройства, на котором будут просматривать сайт.

Рассмотрим префиксы немного подробнее.

	Очень маленькие устройства Телефоны (<768px)	Малые устройства Планшеты (≥768px)	Средние устройства Настольные (≥992px)	Большие устройства Настольные (≥1200px)
Поведение разметки	Все время горизонтальный	Менять значения, как только встречается новая контрольная точка		
Ширина контейнера	Нет (автом.)	750px	970px	1170px
Класс префикса	.col-xs-	.col-sm-	.col-md-	.col-lg-
# колонок	12			
Ширина колонки	Авто	60 px	78 px	95 px
Промежуточная ширина	30 px (15 px) на каждой стороне колонки			
Вкладка	Да			
Отступ	Да			
Выравнивание колонки	Да			

Создание любой блочной структуры для любого мобильного устройства возможно только с заданием специального префикса. Разбиение на 12 колонок подобрано таким образом, что делится на 2, 3, 4, 6. Теперь не стоит продумывать, как разбить любой блок на 3 равные части, ведь есть такая сетка. Все, что будет находиться за этими 12 блоками, автоматически переносится на новую строку.

## Типографика

В фреймворке есть все для форматирования параметров шрифтов, абзацев, цитат и т. д. Нужно только выбрать из специально подобранных стилей, где уже тщательно выставлены все отступы, интервалы, заголовки, и добавить на сайт. Большое внимание уделили семантике.

## Выводы

Актуальную информацию можно найти на официальном сайте Twitter Bootstrap <http://getbootstrap.com/> или его русском аналоге <http://bootstrap-3.ru/index.php> (рекомендуется выбрать зарубежный вариант). Переносить большую часть материалов в методичку нецелесообразно, так как фреймворк постоянно улучшается и изменяется.

# HTML5: аудио и видео

## Аудио HTML5

В следующем коде приведен пример использования элемента <audio>:

```
<p>Ознакомительный фрагмент аудио. :</p>
<audio src="audio.mp3" controls></audio>
```

Атрибут src содержит имя аудиофайла для воспроизведения, а атрибут controls указывает браузеру, что нужно отобразить базовые элементы управления воспроизведением. Своим внешним видом эти элементы управления слегка отличаются от браузера к браузеру, но у них одинаковое назначение: разрешают пользователю начинать и останавливать воспроизведение, переходить в другое место записи и регулировать громкость (рис. 1).

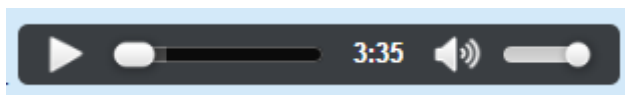


Рис. 1. Аудиопроигрыватель

Кроме атрибута controls, элемент <audio> поддерживает ещё три: preload, autoplay и loop. Атрибут preload указывает браузеру способ загрузки аудиофайла. Значение auto этого атрибута указывает браузеру загружать аудиофайл полностью, чтобы он был доступен, когда пользователь нажмет кнопку воспроизведения.

Загрузка осуществляется в фоновом режиме. Так посетитель веб-страницы может перемещаться по ней и просматривать её, не дожидаясь завершения загрузки аудиофайла.

Атрибут preload может принимать два значения. Значение metadata указывает браузеру загрузить первую небольшую часть файла, достаточную, чтобы определить некоторые его основные характеристики (например, общий размер файла). Атрибут none указывает браузеру не загружать аудиофайл автоматически. Эти опции можно использовать, чтобы сэкономить пропускную способность подключения.

```
<audio src="rubberduckies.mp3" controls preload="metadata"></audio>
```

Когда атрибуту `preload` задано значение `none` или `metadata`, загрузка аудиофайла начинается после того, как пользователь нажмет кнопку воспроизведения.

Если значение атрибута `preload` не установлено, браузеры действуют по своему усмотрению. Большинство браузеров предполагает `auto` в качестве значения по умолчанию. Важно отметить, что атрибут `preload` – не обязательное для выполнения правило, а рекомендация желаемого действия, которую браузер может игнорировать в зависимости от других обстоятельств.

Атрибут `autoplay` указывает браузеру начать воспроизведение сразу же после завершения загрузки страницы:

```
<audio src="audio.mp3" controls autoplay></audio>
```

Если этот атрибут не используется, пользователь должен нажать кнопку запуска, чтобы начать воспроизведение. Элемент `<audio>` можно использовать, чтобы проигрывать фоновую музыку.

Атрибут `loop` указывает браузеру повторять воспроизведение:

```
<audio src="audio.mp3" controls loop></audio>
```

## Воспроизведение видео с помощью элемента `<video>`

С элементом `<audio>` хорошо идет в паре элемент `<video>`. Он применяет такие же атрибуты: `src`, `controls`, `autoplay` и `loop`. Пример использования этого элемента показан в следующем коде:

```
<p>Демонстрационное видео!</p>
<video src="video.mp4" controls></video>
```

Как и в случае с элементом `<audio>`, атрибут `controls` указывает браузеру создать набор элементов управления воспроизведением (рис. 2). В большинстве браузеров эти элементы скрываются при щелчке где-нибудь в области страницы и отображаются опять при наведении курсора мыши на область видеоплеера.



Рис. 2. Видеопроеигрыватель

Кроме общих с элементом `<audio>` атрибутов, элемент `<video>` имеет три собственных: `height`, `width` и `poster`.

Атрибуты `height` и `width` устанавливают высоту и ширину окна воспроизведения в пикселах, соответственно. Следующий код показывает пример создания области воспроизведения размером 400 x 300 пикселей:

```
<video src="video.mp4" controls width= "400" height="300"></video>
```

Размеры окна воспроизведения должны совпадать с размером видео, но лучше явно указать их, чтобы оформление страницы не искажалось до того, как видеофайл загрузится (или если видеофайл вообще не загружается).

Атрибут `poster` позволяет указать изображение, которое можно использовать вместо видео:

```
<video src="video.mp4" controls poster="pict.jpg"></video>
```

Браузеры показывают это изображение в трех ситуациях: когда первый кадр видео еще не загрузился, атрибуту `preload` присвоено значение `none` или указанный видеофайл отсутствует.

## Элемент `<source>`

Для совместимости с различными браузерами у нас есть не один видеофайл, а три: это OGV-файл, MP4-файл и WebM-файл. HTML5 позволяет сделать ссылки на все три файла с помощью элемента `<source>`. Каждый элемент `<video>` может содержать более одного тега `<source>`. Браузер пройдет по списку источников видео по порядку и выберет первым то, что он в состоянии воспроизвести.

Браузер проводит проверку, какое видео он сможет воспроизвести. В худшем случае он загружает каждое видео и пытается его проиграть, однако это большая трата пропускной способности. Вы

сэкономите много трафика, если сообщите браузеру информацию о каждом видео. Это можно сделать атрибутом type тега <source>.

```
<video width="320" height="240" controls>
<source src="video.mp4" type='video/mp4; codecs="avc1.42E01E, mp4a.40.2"'>
<source src="video.webm" type='video/webm; codecs="vp8, vorbis"'>
<source src="video.ogv" type='video/ogg; codecs="theora, vorbis"'>
</video>
```

Элемент <video> определяет ширину и высоту видео, но не ссылку на видеофайл. Внутри <video> три элемента <source>. Каждый элемент <source> ссылается на отдельный видеофайл (с атрибутом src), а также дает информацию о видеоформате (в атрибуте type).

Тип атрибута – комбинация из трех блоков информации: формат файла, видеокодек и аудиокодек. Для видеофайла .ogv формат контейнера – Ogg, представленный здесь как video/ogg, строго говоря, это MIME-тип для видеофайлов Ogg. Видеокодек – Theora, аудиокодек – Vorbis. Само значение должно быть заключено в кавычки, поэтому нужно использовать различные виды кавычек, чтобы окружить значение целиком:

```
<source src="pr6.ogv" type='video/ogg; codecs="theora, vorbis"'>
```

С WebM почти так же, но с другим MIME-типом (video/webm вместо video/ogg) и другим видео-кодеком (vp8 вместо theora), написанным в параметре codecs:

```
<source src="pr6.webm" type='video/webm; codecs="vp8, vorbis"'>
```

Видео H.264 является более сложным:

```
<source src="pr6.mp4" type='video/mp4; codecs="avc1.42E01E, mp4a.40.2"'>
```

Преимуществом является то, что браузер проверяет атрибут type первым и смотрит, может ли он воспроизвести видеофайл. Если браузер решает, что не может этого сделать, он не будет скачивать файл даже частично.

## Элемент <track>

Элемент <track> задает любой текст, который бы вы хотели показывать во время проигрывания медиафайлов: субтитры, подписи, описания, главы или метаданные.

Элемент <track> позволяет указать дополнительные ресурсы с текстом, который будет отображаться синхронно с временной шкалой вашего видео- или аудиофайла.

У <track> нет парного закрывающего тега (то есть это пустой элемент). Он должен располагаться внутри тега <video> или <audio>. К тому же, если внутри аудио- или видеотега присутствует элемент <source>, то элемент <track> должен располагаться после него.

```
<video width="640" height="320" controls>
<source src="some_video.mp4" type="video/mp4">
<source src="some_video.ogv" type="video/ogg">
<track src="some_video_subtitles.srt"
      kind="subtitles"
      srclang="en"
      label="English_subs">
</video>
```



## Атрибут src

Этот атрибут определяет адрес текстового файла, содержащего данные дорожки, и является обязательным. Его значением должен быть абсолютный или относительный URL. Это означает, что файлы должны быть расположены на сервере.

Например:

```
<track src="video_captions.srt">
```

## Структура файла .srt

Атрибут элемента отслеживания src указывает на текстовый файл, содержащий данные о хронометрированных метках отслеживания в любом формате, который распознается браузером. Chrome поддерживает формат WebVTT, который имеет описанный ниже вид.

```
WEBVTT FILE
railroad
00:00:10.000 --> 00:00:12.500
Left uninspired by the crust of railroad earth
manuscript
00:00:13.200 --> 00:00:16.900
that touched the lead to the pages of your manuscript.
```

## Атрибут srclang

Атрибут srclang указывает язык прикрепленного текста. Этот атрибут необходимо указывать, если значение атрибута kind выставлено в subtitles. Значение атрибута srclang должно быть валидным языковым тегом в формате BCP47. Например, значение en используется для английского.

```
<track src="video_subtitles.srt" kind="subtitles" srclang="en">
```

Этот пример указывает, что файл субтитров представлен для английского языка.

## Атрибут kind

Этот атрибут указывает тип дорожки, которую мы добавляем к медиафайлу. Он может принимать одно значение из некоторого набора:

- Subtitles.
- Captions.
- Description.
- Metadata.
- Chapters.

## Субтитры (subtitles)

Обычно представляют из себя перевод диалогов, которые звучат в видео- или аудиофайле. Они бывают полезны, когда пользователь не понимает языка оригинала: у него есть возможность читать перевод диалогов. Указание языка источника обязательно. Это делается путем указания соответствующего значения атрибуту srclang:

```
<track src="video_subtitles.srt" kind="subtitles" srclang="en">
```

## Подписи (captions)

Подпись – краткое описание, сопровождающее видео во время проигрывания. Бывает полезно в ситуациях, когда пользователю нужно сообщить важную информацию, если звук плохо различим или не слышен. Простой пример:

```
<track src="video_captions.srt" kind="captions">
```

## Описания (descriptions)

Как понятно из наименования, данный тип трека используется для описания медиаконтента. Синхронные дорожки, помеченные как описания, обычно представляют собой отдельную аудиодорожку. Например:

```
<track src="video_descriptions.srt" kind="descriptions">
```

## Метаданные (metadata)

Предназначен для треков, которые предоставляют метаданные. Они не отображаются браузерами. Значения метаданных предназначены для использования скриптами вроде JavaScript. Например:

```
<track src="video_metadata.srt" kind="metadata">
```

## Главы (chapters)

Представляет собой заголовок, предназначен для использования в навигации по медиаресурсу. Дорожки, обозначенные как главы, обычно отображаются пользователю в виде интерактивного списка.

```
<track src="video_chapters.srt" kind="chapters">
```

## Атрибут label

Атрибут предназначен для указания названия текстовой дорожки, подключенной к аудио / видеофайлу. Используется браузерами при отображении пользователю списка доступных дорожек и представляет собой название дорожки в удобном пользователю виде.

Если вы добавите атрибут label элементу <track>, его значение нельзя оставлять пустым. Значение должно быть строковым параметром. Если атрибут отсутствует, браузер подставит значение по умолчанию, вроде «без названия».

```
<track src="video_subtitles.srt"
      kind="subtitles"
      srclang="en"
      label="Английские субтитры">
```

В этом примере можно видеть, что атрибут label содержит значение «Английские субтитры».

## Атрибут default

Используется для указания дорожки, которая будет выбрана по умолчанию. Атрибут default может быть задан только одному элементу <track> для одного медиафайла. Такой элемент будет выбран по умолчанию, если пользователь не выберет другую дорожку.

Следующий пример показывает, что из предложенных субтитров на разных языках (английский, японский, русский) по умолчанию будут выбраны и отображены вместе с видео субтитры на русском:

```
<track kind="subtitles" src="video_subtitles_ru.srt" srclang="ru" default>
<track kind="subtitles" src="video_subtitles_en.srt" srclang="en">
<track kind="subtitles" src="video_subtitles_ja.srt" srclang="ja">
```

Теперь, когда мы рассмотрели основные атрибуты, которые могут использоваться с тегом <track>, распишем небольшой пример использования нескольких типов дорожек в рамках элементов <video> и <source>:

```
<video src="sample.ogv">
  <source src="video.mp4" type="video/mp4">
  <source src="video.ogv" type="video/ogg">
  <track kind="captions" src="video_captions.srt" srclang="en">
  <track kind="descriptions" src="video_descriptions.srt" srclang="en">
  <track kind="chapters" src="video_chapters.srt" srclang="en">
  <track kind="subtitles" src="video_subtitles_en.srt" srclang="en"
default>
  <track kind="subtitles" src="video_subtitles_oz.srt" srclang="oz">
  <track kind="metadata" src="video_metadata1.srt" srclang="en"
label="Метаданные 1">
  <track kind="metadata" src="video_metadata2.srt" srclang="en"
label="Метаданные 2">
</video>
```

## Полезность медиадорожек для поисковой оптимизации

Элемент <track> открыл дорогу к оптимизации под поисковые системы для видео, а также дал возможность сделать ваши видео более понятными поисковым системам.

Поисковые системы все лучше понимают различное мультимедиа-содержимое, и чем большим количеством информации вы можете сопроводить свои файлы, тем более таргетированный и большего объема трафик вы сможете получить.

Некоторые ключевые преимущества поисковой оптимизации:

- Улучшает ваше присутствие в поисковой выдаче: поисковые системы при поиске проходятся по всем текстовым данным, ассоциированным с вашим видео.
- Лучшая связность: поисковые системы возвращают результаты поиска, которые указывают на конкретные части видео, ассоциированные с временными кодами.
- Связанное содержимое: текстовые файлы могут быть легко включены в связанное текстовое содержимое на той же странице.

- Доступность и пользовательское взаимодействие: субтитры и подписи улучшают юзабилити и доступность для людей с инвалидностью.
- Миниатюры в результатах поиска: страница с видео может быть отображена в результатах поиска в виде фрагмента с миниатюрой, что может повысить количество переходов по ссылкам.

## Практическое задание

1. Научиться пользоваться официальной документацией: <http://getbootstrap.com/>.
2. Приступить к верстке макета 3, товар каталога single page.psd.
3. \* Создать шаблон с использованием Twitter Bootstrap.
4. \* Добавить блок видео вместо блока с любой картинкой на сайте.
5. \* Ознакомиться с новой, 4-й версией Bootstrap.

Задания со \* предназначены для учеников, которые справились с основной частью.

## Дополнительные материалы

Самое важное и основное – <http://getbootstrap.com/>.

<audio> – <https://developer.mozilla.org/ru/docs/Web/HTML/Element/audio>.

<video> – <https://developer.mozilla.org/ru/docs/Web/HTML/Element/video>.

<track> – <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/track>.

## Используемая литература

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. <http://www.wisdomweb.ru/>.
2. <http://html5book.ru/css3-flexbox/>.
3. <http://sass-scss.ru/guide/>.
4. <http://dedushka.org/uroki/6901.html>.
5. Гоше Х. HTML5. Для профессионалов. – СПб.: Питер, 2013. – 496 с.: ил.
6. Хоган Б. HTML5 и CSS3. Веб-разработка по стандартам нового поколения. – СПб.: Питер, 2012. – 272 с.
7. Макфарланд Д. Большая книга CSS3. – СПб.: Питер, 2016. – 608 с.