

JS Level 0



HTML

HTML

На прошлой лекции мы с вами остановились на вот такой структуре, которая генерируется с помощью Emmet:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  |
</body>
</html>
```

Веб-технологии

Сегодня мы познакомимся с более интересными тегами: изображениями, аудио, видео.

img

Изображения (картинки) показываются с помощью тега `img` (сокращение от `image` - изображение):

```
<img></img>
```

Закрывающий тег

На прошлой лекции, рассматривая элементы, мы говорили, что есть открывающий и закрывающий тег.

Если между тегами элемента нет содержимого, вы можете встретить сокращённую запись: `` -> `` (но это работает не для всех браузеров, поэтому лучше не использовать).

Но и это не всё: для некоторых элементов можно вообще оставлять только открывающий тег: ``.

Закрывающий тег

Как определить, для каких элементов закрывающий тег нужен, а для каких - нет?

Достаточно просто:

1. Нужно открыть спецификацию html: <https://html.spec.whatwg.org>
2. Пролистать в оглавлении до нужного раздела (элементы будут подсвечиваться оранжевым цветом):

4.8 Embedded content

4.8.1 The **picture** element

4.8.2 The **source** element

4.8.3 The **img** element ←

3. В карточке описания посмотреть на блок tag omission (отсутствие тега):

Tag omission in text/html:

No **end tag**.

Если написано "No end tag" (нет закрывающего тега), то закрывающий тег можно не писать.

Атрибуты

Это всё хорошо, элемент мы написали, но картинки-то нет? Чтобы задать саму картинку, нам понадобятся атрибуты.

Атрибуты - это "свойства" элемента (примерно так же, как у объектов в реальной жизни есть свойства, например цвет у автомобиля, у HTML элементов есть свои свойства).

Атрибуты

Атрибуты задаются в открывающем теге в виде: **названиеАтрибута="значениеАтрибута"**.

Так же, как и для тегов - есть сокращения. Если значение атрибута - пустая строка (т.е. **названиеАтрибута=""**), то допускается не писать **=""**, т.е. достаточно написать лишь **названиеАтрибута**).

Обратите внимание: в HTML принято использовать двойные (для этого надо нажать на **Shift** и на тот же символ кавычки).

Атрибуты

Например, для элемента **img** есть атрибут **src** (source - источник), который позволяет указать адрес картинки в сети Интернет:

```

```

Получим вот такой результат:



Изображения

Но откуда мы узнаем адрес изображения? Почему он именно такой? Чаще всего мы будем знать адрес изображения исходя из того, что сами его "разместим" по этому адресу в сети интернет (как это сделать узнаем через несколько лекций). Пока же мы будем давать вам готовые адреса.

Атрибуты

Где взять информацию о том, какие атрибуты есть и что в них можно писать?

Информация есть в спецификации:

Content attributes:

Global attributes

alt — Replacement text for use when images are not available

src — Address of the resource

srcset — Images to use in different situations, e.g., high-resolution displays, small monitors, etc.

sizes — Image sizes for different page layouts

crossorigin — How the element handles crossorigin requests

usemap — Name of [image map](#) to use

ismap — Whether the image is a server-side image map

width — Horizontal dimension

height — Vertical dimension

referrerpolicy — [Referrer policy](#) for [fetches](#) initiated by the element

decoding — Decoding hint to use when processing this image for presentation

loading — Used when determining loading deferral

Переводчик

Но спецификация на английском языке, и если вы его не знаете, то вам приходится достаточно тяжело. Конечно же, английский язык нужно учить, но мы можем воспользоваться переводчиками, которые работают в виде сайтов.

Что нужно сделать: вы выделяете левой кнопкой мыши нужный текст (кликаете левой кнопкой мыши чуть левее текста, чтобы не кликнуть на сам текст и двигаете мышью слева-направо, выделяя текст - отпускать при этом левую кнопку не надо):

Content attributes:

[Global attributes](#)

alt — Replacement text for use when images are not available

src — Address of the resource

srcset — Images to use in different situations, e.g., high-resolution displays, small monitors, etc.

sizes — Image sizes for different page layouts

Переводчик

Content attributes:

[Global attributes](#)

alt — Replacement text for use when images are not available

src — Address of the resource

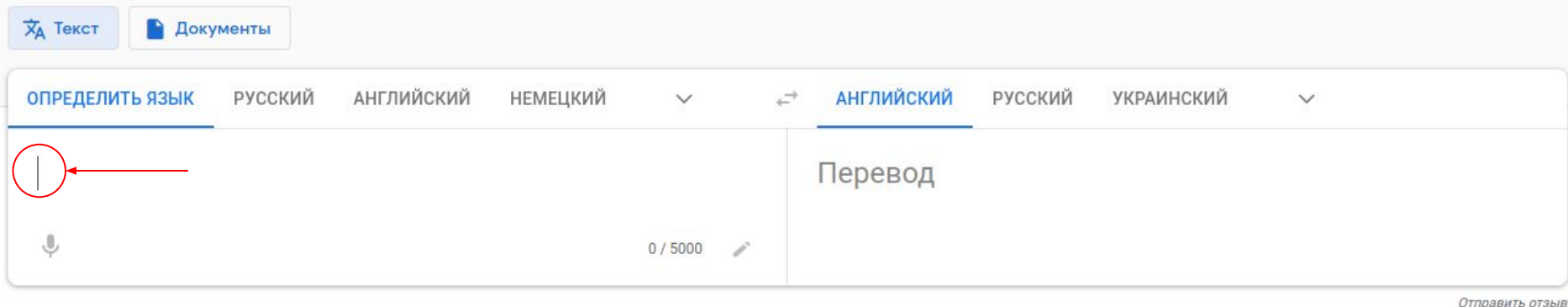
srcset — Images to use in different situations, e.g., high-resolution displays, small monitors, etc.

sizes — Image sizes for different page layouts

После этого (как выделили), нажимаете на клавиатуре **Ctrl + C** (скопировать текст).

Переводчик

Теперь перейдите на сайт translate.google.com:



Текст Документы

ОПРЕДЕЛИТЬ ЯЗЫК РУССКИЙ АНГЛИЙСКИЙ НЕМЕЦКИЙ ▼ ↔ АНГЛИЙСКИЙ РУССКИЙ УКРАИНСКИЙ ▼

Перевод

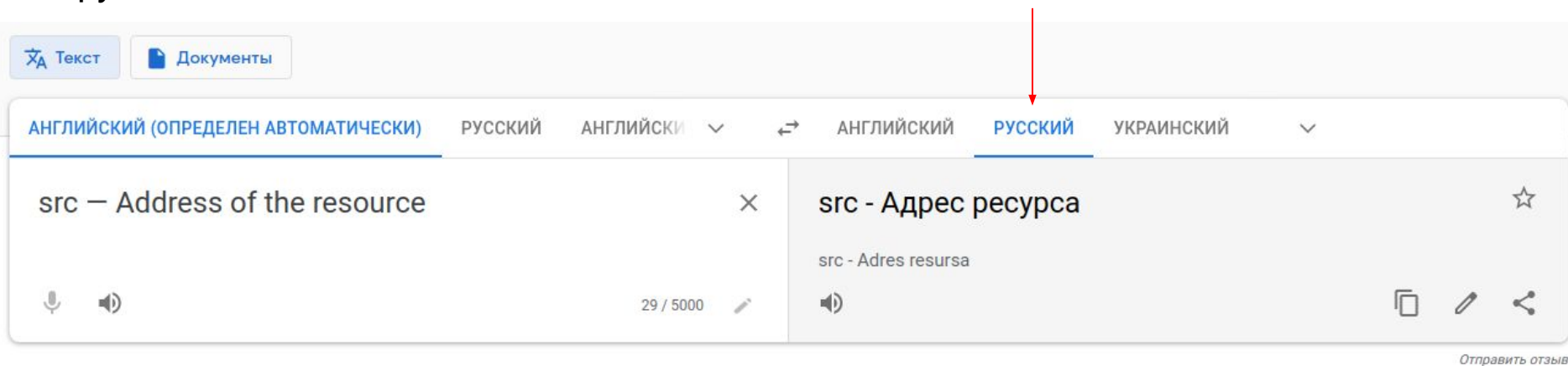
0 / 5000

Отправить отзыв

И нажмите **Ctrl + V** ("вставить" скопированный текст).

Переводчик

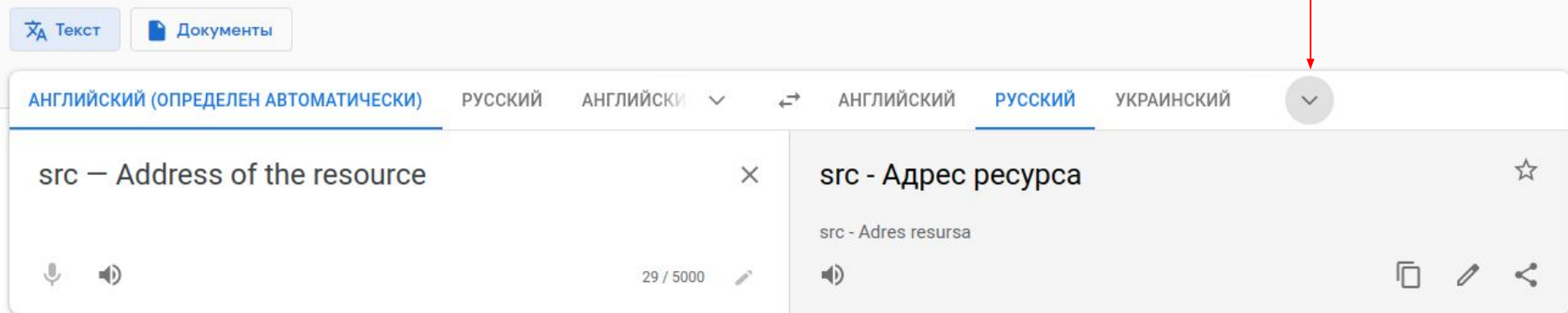
Затем кликните справа на нужном языке (на который нужно перевести), например, русский:



Система за вас переведёт фразу.

Переводчик

Если нужный вам язык не отображается, то вы можете выбрать из всех доступных языков, нажав на кнопку отображения списка:



← Найти языки

азербайджанский	грузинский	корейский	немецкий	словацкий	фризский
албанский	гуджарати	корсиканский	непальский	словенский	хауса
амхарский	датский	креольский (Гаити)	нидерландский	сомалийский	хинди
🕒 английский	зулу	курманджи	норвежский	суахили	хмонг
арабский	иврит	кхмерский	ория	суданский	хорватский
армянский	игбо	кхоса	панджаби	таджикский	чева
африкаанс	идиш	лаосский	персидский	тайский	чешский
баскский	индонезийский	латинский	польский	тамильский	шведский
белорусский	ирландский	латышский	португальский	татарский	шона
бенгальский	исландский	литовский	пушту	телугу	шотландский (гэльский)
бирманский	испанский	люксембургский	руанда	турецкий	эсперанто
болгарский	итальянский	македонский	румынский	туркменский	эстонский
боснийский	йоруба	малагасийский	✓ русский	узбекский	яванский
валлийский	казахский	малайский	самоанский	уйгурский	японский
венгерский	каннада	малаялам	себуанский	украинский	
вьетнамский	каталанский	мальтийский	сербский	урду	
гавайский	киргизский	маори	сесото	филиппинский	
галисийский	китайский (традиционн...	маратхи	сингальский	финский	
греческий	китайский (упрощенны...	монгольский	синдхи	французский	

webref

Переводчик и спецификация - это, конечно, хорошо, но хотелось бы более простого описания на время нашего обучения.

Для этого есть специальные веб-сайты, один из лучших на русском языке - webref.ru.

Вы заходите на этот сайт и
в правом меню будет панелька
с элементами:



Введите HTML или CSS

Элементы HTML

<!-->

<!DOCTYPE>

<a>

<abbr>

<acronym>

<address>

<applet>

Свойства CSS

!important

::-ms-browse

::-ms-check

::-ms-clear

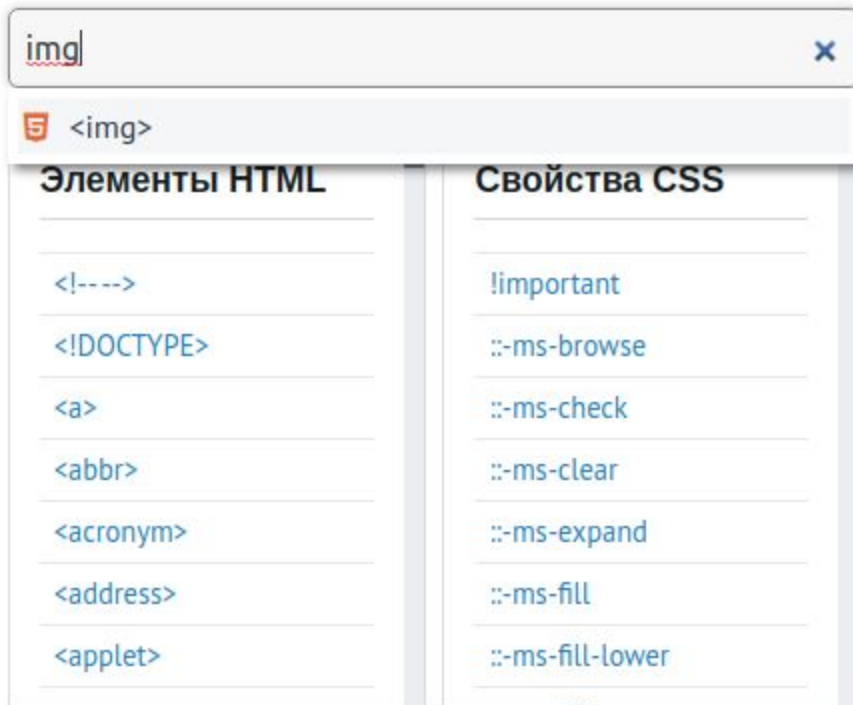
::-ms-expand

::-ms-fill

::-ms-fill-lower

webref

Вводите название нужного элемента и нажимаете Enter:



The screenshot shows a search interface for 'webref'. At the top, a search bar contains the text 'img' and a close button (X). Below the search bar, a dropdown menu displays the search results for ''. The results are organized into two columns: 'Элементы HTML' (HTML Elements) and 'Свойства CSS' (CSS Properties). The 'Элементы HTML' column lists various HTML tags, including '<!-->', '<!DOCTYPE>', '<a>', '<abbr>', '<acronym>', '<address>', and '<applet>'. The 'Свойства CSS' column lists various CSS properties, including '!important', '::ms-browse', '::ms-check', '::ms-clear', '::ms-expand', '::ms-fill', and '::ms-fill-lower'.

Элементы HTML	Свойства CSS
<!-->	!important
<!DOCTYPE>	::ms-browse
<a>	::ms-check
<abbr>	::ms-clear
<acronym>	::ms-expand
<address>	::ms-fill
<applet>	::ms-fill-lower

webref

И уже на русском языке читаете описание вместе с примером использования:

Синтаксис

```
">
```

Закрывающий тег

Не требуется.

Атрибуты

align	Определяет, как рисунок будет выравниваться по краю и способ обтекания текстом. Устарел
alt	Альтернативный текст для изображения. Обязательный
border	Толщина рамки вокруг изображения. Устарел
height	Высота изображения.
hspace	Горизонтальный отступ от изображения до окружающего контента. Устарел
ismap	Говорит браузеру, что картинка является серверной картой-изображением.
longdesc	Указывает адрес документа, где содержится аннотация к картинке. Устарел в HTML5
sizes	Задаёт размеры изображения для разных макетов страницы. HTML5
src	Путь к графическому файлу. Обязательный
srcset	Путь к графическим файлам с учётом размера изображения и устройств. HTML5
vspace	Вертикальный отступ от изображения до окружающего контента. Устарел
width	Ширина изображения.
usemap	Ссылка на элемент <map>, содержащий координаты для клиентской карты-изображения.

Также для этого элемента доступны [универсальные атрибуты](#) и [события](#).

audio

Для отображения аудио-элементов используется тег audio. Так же, как и для изображений, нам необходим файл, в котором и будут содержаться сами данные, например, музыка*:

```
<audio src="https://lms.openjs.io/ukelele.mp3"></audio>
```

Примечание*: не бывает просто файлов изображений, аудио и видео. У большинства файлов есть владельцы, которые могут не разрешать просто так использовать их изображения, аудио и видео. Поэтому нужно быть достаточно аккуратным, копируя чужие произведения. Музыку мы взяли с сайта [bensound.com](https://www.bensound.com), который позволяет использовать их медиа-файлы, если указывается на то, что именно они - авторы. Данный файл мы взяли с ресурса <https://www.bensound.com>

audio

Обратите внимание: у **audio**, как и у **video**, закрывающий тег обязателен.

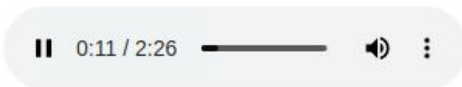
audio

Если мы разместим подобным образом элемент, то не увидим ничего на страничке.

Дело в том, что элемент аудио по умолчанию никак не отображается. С помощью специального атрибута **controls** можно настроить отображение элементов управления:

```
<audio src="https://lms.openjs.io/ukelele.mp3" controls></audio>
```

Получим вот такой результат:



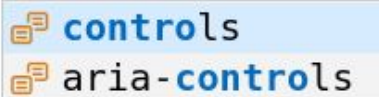
Атрибуты

`controls` как раз является примером того, что мы обсуждали: вместо `controls=""` мы пишем сокращённую версию.

Атрибуты

Когда вы будете набирать код, VSCode будет помогать вам, указывая те атрибуты, что он знает:

```
<body>  
<audio src="https://lms.openjs.io/ukelele.mp3" contro></audio>  
</body>  
</html>
```

The image shows a screenshot of a code editor with a dropdown menu. The code in the background is: `<body>
<audio src="https://lms.openjs.io/ukelele.mp3" contro></audio>
</body>
</html>`. The word 'contro' is highlighted in red. A dropdown menu is open, showing two options: 'controls' and 'aria-controls'. Both options are preceded by a small icon of a document with a speech bubble. The 'controls' option is highlighted in blue.

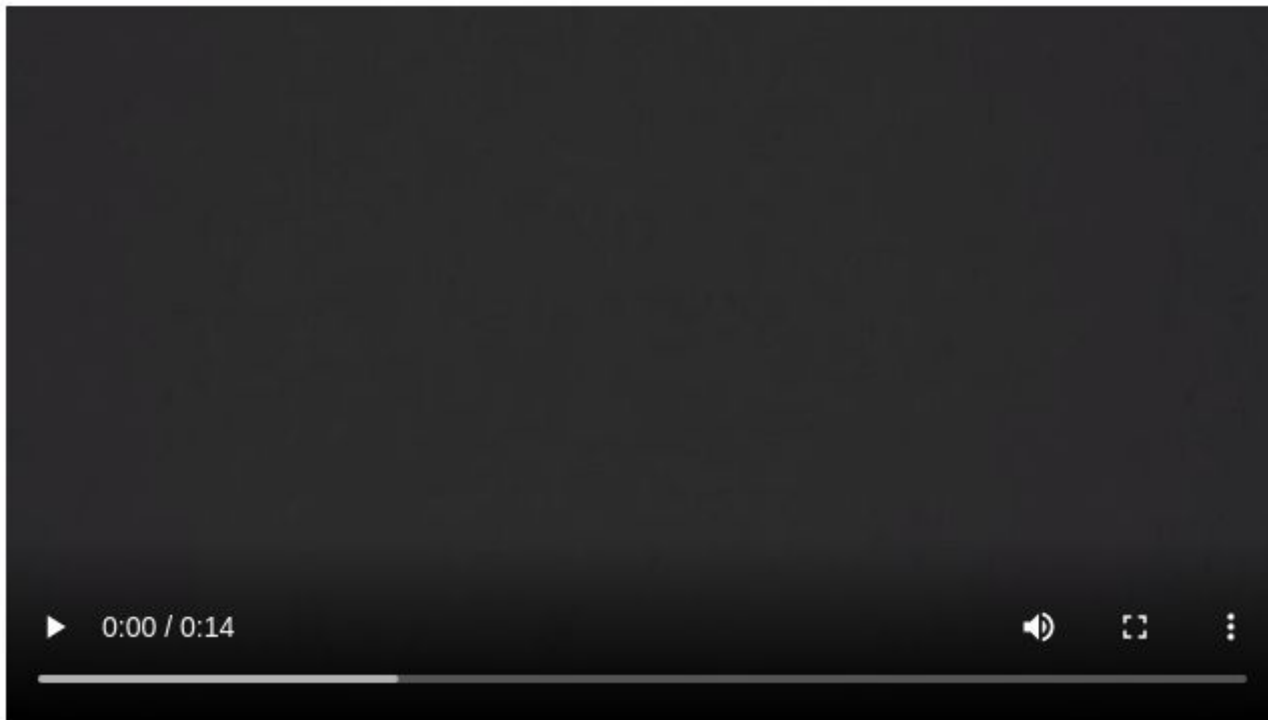
Обратите внимание: если атрибута, который вы набираете там нет, то скорее всего, вы пишете неправильно имя атрибута.

Кроме того, если атрибут подсвечен (вы можете перемещать подсветку с помощью клавиш вверх и вниз), то нажатие на Tab приведёт к тому, что VSCode за вас до конца допишет название атрибута.

video

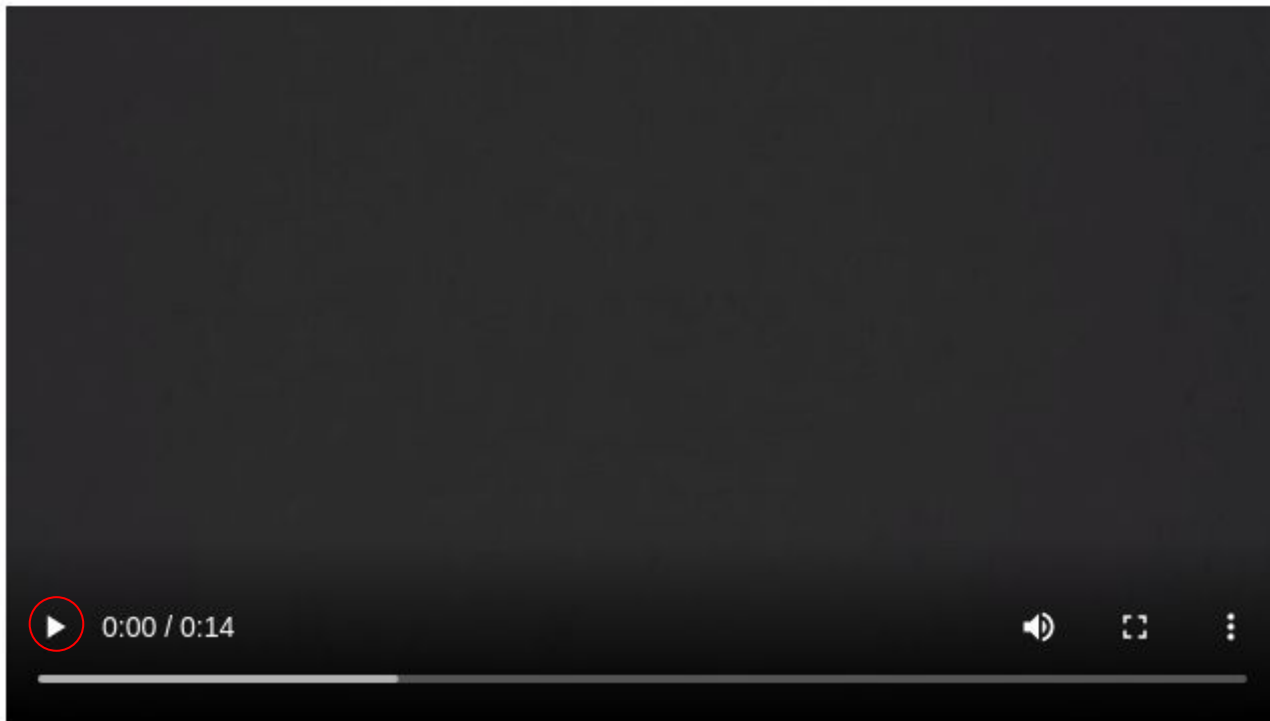
С элементом видео всё то же самое, что и с **audio**:

```
<video src="https://lms.openjs.io/video.mp4" controls></video>
```



audio, video

Для начала проигрывания **audio** и **video** обычно используется специальная кнопка, которая называется Play (выглядит в виде треугольника):



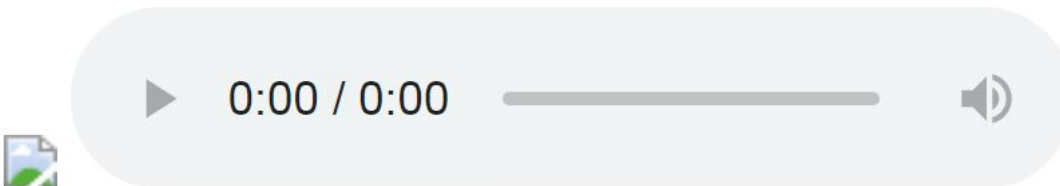
Ошибки

Что будет, если мы случайно ошибёмся и укажем неправильные адреса?

Давайте посмотрим:

```
  
<audio src="https://lms.openjs.io/404.mp3" controls></audio>  
<video src="https://lms.openjs.io/404.mp4" controls></video>
```

Для изображения будет отображаться "битая" картинка, для аудио - заблокированный плеер, а видео не будет отображаться совсем (в браузере Chrome):



Ошибки

Q: когда такое может произойти?

A: например, вы сделали документ, выложили в Интернет. А кто-то (например, владелец другого сайта), взял и удалил изображения со своего сайта, адреса которых вы использовали.

alt

У изображений (элемента **img**) предусмотрен специальный атрибут alt (альтернативное текстовое описание), который используется в случаях:

1. Не удалось использовать картинку
2. Пользователь использует не графический браузер (например, голосовой - он может прочитать описание картинки, но показать её не сможет):

```

```



Для аудио и видео атрибута **alt** нет.

Итоги

Итоги

В этой лекции мы начали знакомиться с новыми элементами, поговорили про спецификацию и сайт [webref](#).

Обязательно используйте полученные в практике (в том числе работу со спецификацией и сайтом [webref](#)).

Домашнее задание

Орг.моменты

Курс состоит из 8 обязательных занятий. Мы выкладываем уроки каждую субботу в:

- 19:00 (Москва)
- 21:00 (Душанбе и Ташкент)

Каждую субботу в:

- 16:00 (Москва)
- 18:00 (Душанбе и Ташкент)

дедлайн сдачи домашнего задания.

Если не успеете сдать в срок домашнее задания, тогда этот курс будет для вас закончен и вы сможете зарегистрироваться на запуск следующего через несколько месяцев.

Все вопросы вы сможете задавать в [телеграм-чате](#).

ДЗ: Картинки

Создайте проект аналогично тому, как мы это делали на лекции. Внутри элемента **body** разместите изображение, которое находится по адресу:

https://lms.openjs.io/logo_js.svg

В качестве альтернативного текста используйте "Логотип JS".

Прочитайте на сайте [webref](#) (или спецификации про размеры изображения - ширину и высоту). Установите их в значения 300 (и ширину, и высоту).

Важно: архивировать также нужно содержимое каталога docs.

ДЗ: Видео

Создайте проект аналогично тому, как мы это делали на лекции. Внутри элемента `body` разместите видео, которое находится по адресу:

<https://lms.openjs.io/spring.webm> без элементов управления и с атрибутами `autoplay` и `muted`.

Важно: архивировать также нужно содержимое каталога docs.

Спасибо за внимание

alif academy совместно с aims
2021г.