

GETTING STARTED WITH HTML5

Agenda

- History, Vision & Future of HTML5
- Getting Started With HTML5
- Structure of a Web Page
- Forms
- Audio and Video

1. History, Vision & Future of HTML5

1.1 What Is HTML5?

- Successor of HTML 4.01 and XHTML 1.1
- It comes with new tags, features and APIs
- Below is a non exhaustive list of features that tend to be labelled as "HTML5" in the medias:
 - New structural elements (<header>, <footer>, <nav> and more)
 - Forms 2.0 and client-side validation
 - Native browser support for audio and video (<video>, <audio>)
 - Canvas API and SVG
 - Web storage
 - Offline applications
 - Geolocation
 - Drag & Drop
 - Web Workers
 - New communications API (Server Sent Events, Web Sockets, ...)

1. History, Vision & Future of HTML5

1.2 History of HTML5?

- December 1997: HTML 4.0 is published by the W3C
- February - March 1998: XML 1.0 is published
- December 1999 - January 2000: ECMAScript 3rd Edition, XHTML 1.0 (Basically HTML tags reformulated in XML) and, HTML 4.01 recommendations are published
- May 2001: XHTML 1.1 recommendation is published
- August 2002: XHTML 2.0 first working draft is released.
- December 2002: XHTML 2.0 second working draft published.
- January 2008: First W3C working draft of HTML5 is published!!

1. History, Vision & Future of HTML5

1.3 Future of HTML5

- 84% of Developers Plan to Adopt Key HTML5 Features
- The key to understanding HTML5 is that it is not one, but a group of technologies. Within HTML5, developers have a tremendous amount of choice regarding what they use and what they don't use
- The power of HTML5 being ready for prime-time can be seen in Microsoft's choice to utilize it in [Windows 8](#)

Still on the fence with adopting HTML5 in your next project?

2. Getting started with HTML5

Wondering what it takes to get it started?

- Any Text editor such as [Notepad++](#), [Editplus](#), [Textmate](#), [Dream weaver](#)
- Modern browsers such as [Firefox 3.5 +](#), [IE9](#), [chrome](#), [safari](#)
- Prior knowledge of [HTML 4](#)

3. Structure of Web page

3.1. New and Updated HTML5 Elements

HTML5 introduces 28 new elements:

`<section>`, `<article>`, `<aside>`, `<hgroup>`, `<header>`, `<footer>`, `<nav>`, `<figure>`, `<figcaption>`, `<video>`, `<audio>`, `<source>`, `<embed>`, `<mark>`, `<progress>`, `<meter>`, `<time>`, `<ruby>`, `<rt>`, `<rp>`,

`<wbr>`, `<canvas>`, `<command>`, `<details>`, `<summary>`, `<datalist>`, `<keygen>` and `<output>`

An HTML page first starts with the DOCTYPE declaration

HTML5 also update some of the previous existing elements to better reflect how they are used on the Web or to make them more useful such as:

- The `<a>` element can now also contain flow content instead of just phrasing content
- The `<hr>` element is now representing a paragraph-level thematic break
- The `<cite>` element only represent the title of a work
- The `` element is now representing importance rather than strong emphasis

3. Structure of Web page

3.2. First HTML5 webpage

```
<!DOCTYPE html>
<html>
<head>
<title>Title of the document</title>
</head>

<body>
    That's all I need to create my first HTML5 page
</body>

</html>
```

Output: [Click here](#)

3. Structure of Web page

3.3. New Semantic Elements

<nav>: Represents a major navigation block. It groups links to other pages or to parts of the current page.

<nav> does not have to be used in every place you can find links.

For instance, footers often contains links to terms of service, copyright page and such,

the <footer> element would be sufficient in that case

```
<nav>
<ul>
<li><a href="/">Home</a></li>
<li><a href="/events">Current Events</a></li>
<li><a href="/contact">Contact us</a></li>
</ul>
</nav>
```

3. Structure of Web page

3.3. New Semantic Elements

<Header>: tag specifies a header for a document or section.

```
<body>
<header>
<h1>Little Green Guys With Guns</h1>
<nav>
<ul>
<li><a href="/games">Games</a>
<li><a href="/forum">Forum</a>
<li><a href="/download">Download</a>
</ul>
</nav>
</header>
<p>You have three active games:</p>
<!-- this is still part of the subsection entitled "Games" -->
...
```

However, we mustn't think that "header" is only for masthead of a website. "header" can be use as a heading of an blog entry or news article as every article has its title and published date and time

```
<article>
<header>
<h1>Military Offers Assurances to Egypt and Neighbors</h1>
Published : <time datetime="2011-02-13" pubdate>February 13 2011</time>
</header>
<p>
CAIRO - As a new era dawned in Egypt on Saturday,
the army leadership sought to reassure Egyptians and
the world that it would shepherd a transition to civilian
rule and honor international commitments like the peace treaty with Israel.
</p>
</article>
```

3. Structure of Web page

3.3. New Semantic Elements

<article>: The web today contains a ocean of news articles and blog entries. That gives W3C a good reason to define an element for article instead of `<div class="article">`.

We should use article for content that we think it can be distributable. Just like news or blog entry can we can share in RSS feed

"article" element can be nested in another "article" element.

An article element doesn't just mean article content. You can have [header](#) and [footer](#) element in an article. In fact, it is very common to have [header](#) as each article should have a title.

```
<body>
<h1>My blog</h1>
<article>

  <header>
    <h1>The Very First Rule of Life</h1>
    <p><time pubdate datetime="2009-10-09T14:28-08:00"></time></p>
  </header>

  <p>If there's a microphone anywhere near you, assume it's hot and
  sending whatever you're saying to the world. Seriously.</p>
  <p>...</p>

  <footer>
    <a href="?comments=1">Show comments...</a>
  </footer>
</article>
</body>
```

3. Structure of Web page

3.3. New Semantic Elements

<aside>: The "aside" element is a section that somehow related to main content, but it can be separate from that content

```
<article>
<p>
As of writing, the only web browser completely support date time input is Opera.
In HTML5, it is the job of web browser to ensure user can only enter a valid date time
into the input textbox.
</p>

<aside>
Picking a date from Calendar is not the only way to input a date value even though it's
HTML5 specifications does not mention anything about displaying a calendar for date inp
</aside>
</article>
```

3. Structure of Web page

3.3. New Semantic Elements

<footer>: Similarly to "header" element, "footer" element is often referred to the footer of a web page. Well, most of the time, footer can be used as what we thought.

Please don't think you can only have one footer per web document, you can have a footer in every section, or every article.

```
<ARTICLE>
<H1>My Favorite Trains</H1>
<P>I love my trains. My favorite train of all time is a Kof.</P>
<P>It is fun to see them pull some coal cars because they look so
dwarfed in comparison.</P>
<FOOTER> <!-- footer for article -->
<P>Published <TIME PUBDATE DATETIME="2009-09-15T14:54-07:00"></TIME></P>
</FOOTER>
</ARTICLE>
```

```
<FOOTER> <!-- site wide footer -->
<NAV>
<P><A HREF="/credits.html">Credits</A>-
<A HREF="/tos.html">Terms of Service</A> -
<A HREF="/index.html">Blog Index</A></P>
</NAV>
<P>Copyright © 2009 Gordon Freeman</P>
</FOOTER>
```

```
</BODY>
```

3. Structure of Web page

3.3. New Semantic Elements

<Progress>: The new "progress" element appears to be very similar to the "[meter](#)" element. It is created to indicate progress of a specific task.

The progress can be either determinate OR indeterminate. Which means, you can use "progress" element to indicate a progress that you do not even know how much more work is to be done yet.

Progress of Task A : `<progress value="60" max="100">60%</progress>`

Progress of Task A : 

Browsers	Progress element support
IE 9 Beta	
Firefox 13	✓
Safari 5	
Chrome 8	✓
Opera 11	✓

3. Structure of Web page

3.3. New Semantic Elements

<meter>: "Meter" is a new element in HTML5 which represents value of a known range as a gauge. The keyword here is "known range". That means, you are only allowed to use it when you are clearly aware of its minimum value and maximum value.

One example is score of rating. I would rate this movie `<meter min="0" max="10" value="8">8 of 10</meter>`.

```
Science : <meter min="0" max="100" value="95">95 of 100</meter> <br />
Math : <meter min="0" max="100" value="60">60 of 100</meter><br />
Geography : <meter min="0" max="100" value="20">20 of 100</meter> <br />
History : <meter min="0" max="100" value="50">50 of 100</meter>
```

Science : 

Math : 

Geography : 

History : 

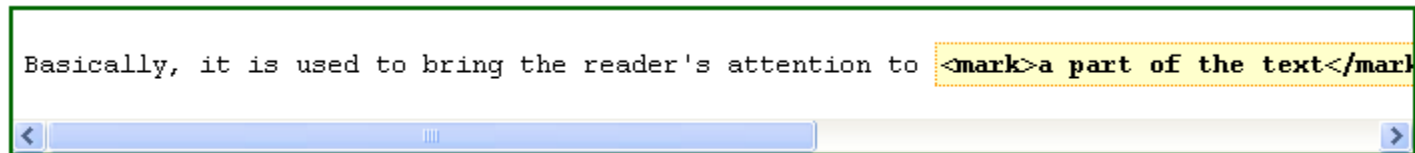
Browsers	Render meter element as gauge
IE 9 Beta	
Firefox 13	
Safari 5	
Chrome 8	✓
Opera 11	✓

3. Structure of Web page

3.3. New Semantic Elements

<mark>: The mark <mark> element represents a run of text in one document marked or highlighted for reference purposes, due to its relevance in another context.

Basically, it is used to bring the reader's attention to a part of the text that might not have been



3. Structure of Web page

3.3. New Semantic Elements

<figure>: The <figure> tag specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.

While the content of the <figure> element is related to the main flow, its position is independent of the main flow, and if removed it should not affect the flow of the document

```
<figure>
  
</figure>
```

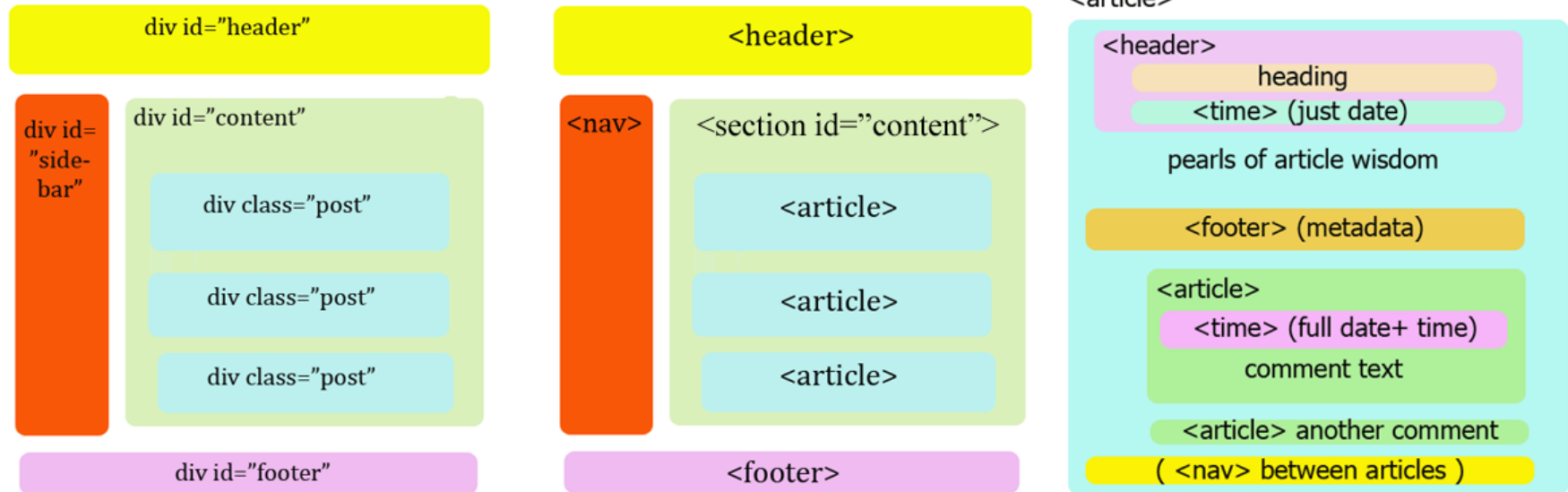
3. Structure of Web page

3.3. Lets put altogether



3. Structure of Web page

3.3. Designing blog with html5



IE layout

There's one gotcha about styling HTML 5 pages in IE: it doesn't work.

You can force it to quite easily with a JavaScript hack

`document.createElement('element name');`

[HTML 5 enabling script](#) which I use in the header to conjure all the missing elements into existence all at once.

3. Structure of Web page

3.3. why so much fuss for semantics in html5

Well, they add semantics to the page. The browser now knows which area of your site is the header or the footer because there are header and footer elements, whereas div might be called “branding

The first is for [search engine use](#); it’s easy to imagine Messrs Google or Yahoo! giving lower weighting to content in footer elements, or extra weight to content in the header.

[site navigable](#) for people with disabilities. People with learning difficulties might instruct their browser always to put the articles before the navigation

HTML 5 Forms

4. Forms

4.1. Introduction

HTML5 web forms have introduced new form elements, input types, attributes, and other features. Many of these features we've been using in our interfaces for years: form validation, combo boxes, placeholder text, and the like. The difference is that where before we had to resort to JavaScript to create these behaviors, they're now available directly in the browser; all you need to do is set an attribute in your markup to make them available.

4. Forms

4.1. Introduction

HTML5 web forms have introduced new form elements, input types, attributes, and other features. Many of these features we've been using in our interfaces for years: form validation, combo boxes, placeholder text, and the like. The difference is that where before we had to resort to JavaScript to create these behaviors, they're now available directly in the browser; all you need to do is set an attribute in your markup to make them available.

HTML5 not only makes **marking up forms easier** on the developer, it's also better for the user. With client-side validation being handled natively by the browser, there will be **greater consistency** across different sites, and many **pages will load faster** without all that redundant JavaScript.

Now Lets look at

HTML5 [Form Attributes](#)

HTML5 [Input Types](#)

HTML5 [Form Elements](#)

4. Forms

4.1. form attributes

<form> / <input> autocomplete Attribute

The autocomplete attribute specifies whether a form or input field should have autocomplete on or off. When autocomplete is on, the browser automatically complete values based on values that the user has entered before.

Example: [../forms/autocomplete.html](#)

4. Forms

4.1. form attributes

<form> / <input> novalidate Attribute

The novalidate attribute is a boolean attribute.

When present, it specifies that the form-data (input) should not be validated when submitted.

Example: [../forms/noValidate.html](#)

4. Forms

4.2. input attributes

<input> autofocus Attribute

The autofocus attribute is a boolean attribute.

When present, it specifies that an <input> element should automatically get focus when the page loads.

Only one form element can have autofocus in a given page.

Example: ../forms/autofocus.html

4. Forms

4.2. input attributes

<input> formaction Attribute

- The **formaction** attribute specifies the URL of a file that will process the input control when the form is submitted.
- The **formaction** attribute overrides the action attribute of the <form> element.
- **Note:** The **formaction** attribute is used with type="submit" and type="image"

Example: ../forms/formaction.html

4. Forms

4.2. input attributes

<input> formmethod Attribute

- The formmethod attribute defines the HTTP method for sending form-data to the action URL.
- The formmethod attribute overrides the method attribute of the <form> element.
- **Note:** The formmethod attribute can be used with type="submit" and type="image".

Example: [../forms/formmethod.html](#)

4. Forms

4.2. input attributes

<input> formnovalidate Attribute

- The novalidate attribute is a boolean attribute.
- When present, it specifies that the <input> element should not be validated when submitted.
- The formnovalidate attribute overrides the novalidate attribute of the <form> element.
- **Note:** The formnovalidate attribute can be used with type="submit".

Example: [../forms/formnovalidate.html](http://localhost:8080/./forms/formnovalidate.html)

4. Forms

4.2. input attributes

<input> formtarget Attribute

- The novalidate attribute is a boolean attribute.
- When present, it specifies that the <input> element should not be validated when submitted.
- The formnovalidate attribute overrides the novalidate attribute of the <form> element.
- **Note:** The formnovalidate attribute can be used with type="submit".

Example: [../forms/formtarget.html](#)

4. Forms

4.2. input attributes

<input> list Attribute

- Datalists are currently only supported in Firefox and Opera, but they are very cool.
- They fulfill a common requirement: a text field with a set of predefined autocomplete options.

Example: ../forms/datalist.html

4. Forms

4.2. input attributes

<input> required Attribute

The Boolean required attribute tells the browser to only submit the form if the field in question is filled out correctly. Obviously, this means that the field can't be left empty, but it also means that, depending on other attributes or the field's type, only certain types of values will be accepted.

If a required field is empty or invalid, the form will fail to submit, and focus will move to the first invalid form element.

The required attribute can be set on any input type except button, range, color, and hidden, all of which generally have a default value.

- **Example: [../forms/required.html](https://forms/required.html)**

4. Forms

4.2. input attributes

<input> placeholder Attribute

The placeholder attribute allows a short hint to be displayed inside the form element, space permitting, telling the user what data should be entered in that field.

The placeholder text disappears when the field gains focus, and reappears on blur if no data was entered.

Developers have provided this functionality with JavaScript for years, but in HTML5 the placeholder attribute allows it to happen natively, with no JavaScript required.

- **Example: [../forms/placeholder.html](#)**

```
$("#input[placeholder], textarea[placeholder]").each(function() {  
    if($(this).val()==""){  
        $(this).val($(this).attr("placeholder"));  
        $(this).focus(function(){  
            if($(this).val()==$(this).attr("placeholder")) {  
                $(this).val("");  
                $(this).removeClass('placeholder');  
            }  
        });  
        $(this).blur(function(){  
            if($(this).val()==""){  
                $(this).val($(this).attr("placeholder"));  
                $(this).addClass('placeholder');  
            }  
        });  
    }  
});
```

4. Forms

4.2. input attributes

<input> readonly Attribute

The readonly attribute is similar to the disabled attribute: it makes it impossible for the user to edit the form field. Unlike disabled, however, the field *can* receive focus, and its value is submitted with the form.

```
<label for="about">Article Title</label>  
<input type="text" name="about" id="about" readonly>
```

4. Forms

4.2. input attributes

<input> multiple Attribute

The multiple attribute, if present, indicates that multiple values can be entered in a form control. While it has been available in previous versions of HTML, it only applied to the select element. In HTML5, it can be added to email and file input types as well.

If present, the user can select more than one file, or include several comma-separated email addresses.

At the time of writing, multiple file input is only supported in Chrome, Opera, and Firefox.

Select images: <input type="file" name="img" multiple="multiple" />

4. Forms

4.3. input types

Introduction

you're familiar with: checkbox, text, password, and submit. Here's the full list of types that were available before HTML5:

■ button ■ checkbox ■ file ■ hidden ■ image ■ password ■ radio ■ reset ■ submit ■ text

HTML5 gives us input types that provide for more data-specific UI elements and native data validation. HTML5 has a total of **13** new input types:

- search
- email
- url
- tel
- datetime
- date
- month
- week
- time
- datetime-local
- number
- range
- color

4. Forms

4.3. input types

<input type> email

The email type (type="email") is, unsurprisingly, used for specifying one or more email addresses. It supports the Boolean multiple attribute, allowing for multiple, comma-separated email addresses.

Search type is only supported in Chrome, Opera, firefox and safari.

E-mail: <input type="email" name="usermail" />

4. Forms

4.3. input types

<input type> url

The url type is used for input fields that should contain a URL address.

The value of the url field is automatically validated when the form is submitted.

Search type is only supported in Chrome, Opera, firefox

Add your homepage: `<input type="url" name="homepage" />`

4. Forms

4.3. input types

<input type= tel

For telephone numbers, use the tel input type (type="tel"). Unlike the url and email types, the tel type doesn't enforce a particular syntax or pattern. Letters and numbers—indeed, any characters other than new lines or carriage returns—are valid.

Telephone: <input type="tel" name="usrtel" />

4. Forms

4.3. input types

<input type= range

The range input type (type="range") displays a slider control in browsers that support it (currently Opera and WebKit). As with the number type, it allows the min, max, and step attributes.

Define a control for entering a number whose exact value is not important

```
<label for="rating">On a scale of 1 to 10, my knowledge of HTML5 is:</label>  
<input type="range" min="1" max="10" name="rating" type="range">
```


4. Forms

4.3. input types

<input type="color">

The color input type (type="color") provides the user with a color picker
Supported only in **Opera**

Select your favorite color: <input type="color" name="favcolor" />

HTML 5 audio & video

5. Audio & video

4.3. new html5 element

<audio>

Browser Support



Internet Explorer 9, Firefox, Opera, Chrome, and Safari support the <audio> element.

Until now, there has not been a standard for playing audio files on a web page.

Today, most audio files are played through a plug-in (like flash). However, different browsers may have different plug-ins.

HTML5 defines a new element which specifies a standard way to embed an audio file on a web page: the <audio> element.

Currently, there are 3 supported file formats for the <audio> element: MP3, Wav, and Ogg:

Browser	MP3	Wav	Ogg
Internet Explorer 9	YES	NO	NO
Firefox 4.0	NO	YES	YES
Google Chrome 6	YES	YES	YES
Apple Safari 5	YES	YES	NO
Opera 10.6	NO	YES	YES

Example: ../audioVideo/audio.html

5. Audio & video

4.3. new html5 element

Browser Support



Internet Explorer 9, Firefox, Opera, Chrome, and Safari support the <audio> element.

<audio> : Add Fallback to flash

To be safe, we need to add the fallback to a Flash audio player, in case the browser doesn't support any of the formats we specified. For instance, Firefox 3.5 only supports the audio tag with *Ogg* format, but we might only have the *mp3* file available

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

  <object class="playerpreview" type="application/x-shockwave-flash"
    data="player_mp3_mini.swf" width="200" height="20">
    <param name="movie" value="player_mp3_mini.swf" />
    <param name="bgcolor" value="#085c68" />
    <param name="FlashVars" value="mp3=test.mp3" />
    <embed href="player_mp3_mini.swf" bgcolor="#085c68" width="200"
      height="20" name="movie" align=""
      type="application/x-shockwave-flash" flashvars="mp3=test.mp3">
    </embed>
  </object>

</audio>

<div id="player_fallback"></div>
<script>
  if (document.createElement('audio').canPlayType) {
    if (!document.createElement('audio').canPlayType('audio/mpeg')) {
      swfobject.embedSWF(
        "player_mp3_mini.swf",
        "player_fallback",
        "200",
        "20",
        "9.0.0",
        "",
        { "mp3": "test.mp3",
          "bgcolor": "#085c68" });
    }
  }
</script>
```

Attribute		Value	Description
autoplay	New	autoplay	Specifies that the audio will start playing as soon as it is ready
controls	New	controls	Specifies that audio controls should be displayed (such as a play/pause button etc).
loop	New	loop	Specifies that the audio will start over again, every time it is finished
preload	New	auto metadata none	Specifies if and how the author thinks the audio should be loaded when the page loads
src	New	URL	Specifies the URL of the audio file

5. Audio & video

4.3. new html5 element

<video>

Browser Support



Internet Explorer 9, Firefox, Opera, Chrome, and Safari support the <audio> element.

Until now, there has not been a standard for showing a video/movie on a web page.

Today, most videos are shown through a plug-in (like flash). However, different browsers may have different plug-ins.

HTML5 defines a new element which specifies a standard way to embed a video/movie on a web page: the <video> element.

Currently, there are 3 supported video formats for the <video> element: MP4, WebM, and Ogg:

Browser	MP4	WebM	Ogg
Internet Explorer 9	YES	NO	NO
Firefox 4.0	NO	YES	YES
Google Chrome 6	YES	YES	YES
Apple Safari 5	YES	NO	NO
Opera 10.6	NO	YES	YES

5. Audio & video

4.3. new html5 element

<video attributes>

Browser Support



Internet Explorer 9, Firefox, Opera, Chrome, and Safari support the <audio> element.

Attribute		Value	Description
autoplay	New	autoplay	Specifies that the video will start playing as soon as it is ready
controls	New	controls	Specifies that video controls should be displayed (such as a play/pause button etc).
height	New	pixels	Sets the height of the video player
loop	New	loop	Specifies that the video will start over again, every time it is finished
muted	New	muted	Specifies that the audio output of the video should be muted
poster	New	URL	Specifies an image to be shown while the video is downloading, or until the user hits the play button
preload	New	auto metadata none	Specifies if and how the author thinks the video should be loaded when the page loads
src	New	URL	Specifies the URL of the video file
width	New	pixels	Sets the width of the video player